Title :ppg signal,feaure extract,peak detection ?

Theory: A PPG (Photoplethysmogram) signal is a measurement of the blood volume changes in the microvascular bed of tissue, typically used to monitor heart rate and other vital signs. However, PPG signals can often be contaminated by noise, which may come from various sources like motion artifacts, ambient light fluctuations, or electrical interference.

When a PPG signal is mixed with noise, the overall quality of the signal deteriorates, which makes it harder to extract meaningful information such as heart rate, respiratory rate, or blood oxygen saturation. Noise can manifest as high-frequency fluctuations, low-frequency drifts, or random spikes.

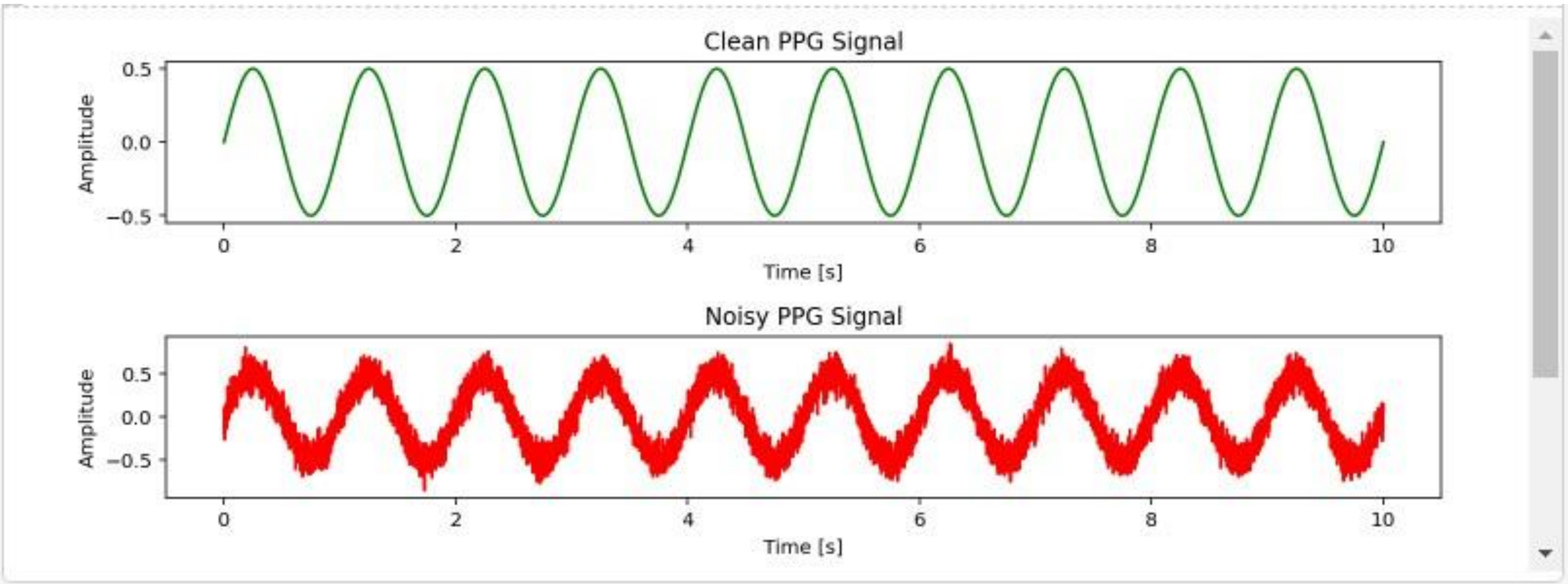
Source code:

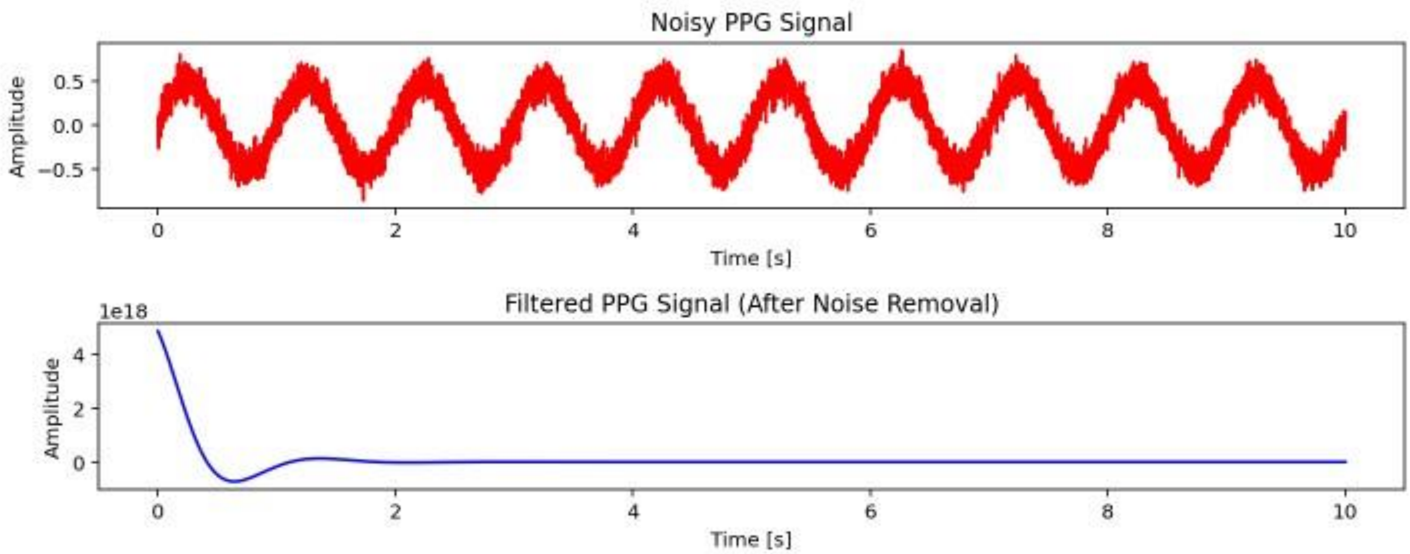
```
import numpy as np

import matplotlib.pyplot as plt

from scipy.signal import butter, filtfilt


# Simulating a clean PPG signal (sine wave for simplicity)

fs = 1000  # Sampling frequency (Hz)

t = np.linspace(0, 10, fs * 10)  # 10 seconds of data

freq = 1  # Heart rate in Hz (1 beat per second, 60 BPM)

clean_ppg = 0.5 * np.sin(2 * np.pi * freq * t)


# Adding Gaussian noise to the PPG signal

noise = np.random.normal(0, 0.1, len(t))

noisy_ppg = clean_ppg + noise


# Bandpass filter to remove high and low-frequency noise
```

```
def butter_bandpass(lowcut, highcut, fs, order=4):
    nyquist = 0.5 * fs
    low = lowcut / nyquist
    high = highcut / nyquist
    b, a = butter(order, [low, high], btype='band')
    return b, a

def bandpass_filter(data, lowcut, highcut, fs, order=4):
    b, a = butter_bandpass(lowcut, highcut, fs, order)
    return filtfilt(b, a, data)

# Bandpass filter settings (0.5 Hz - 5 Hz for PPG signals)
lowcut = 0.5  # Lower frequency (heart rate component)
highcut = 5.0  # Upper frequency (removes high-frequency noise)

# Apply bandpass filter to the noisy signal
filtered_ppg = bandpass_filter(noisy_ppg, lowcut, highcut, fs)

# Plotting the signals
plt.figure(figsize=(10, 6))

# Original Clean PPG Signal
plt.subplot(3, 1, 1)
plt.plot(t, clean_ppg, label="Clean PPG Signal", color='g')
plt.title("Clean PPG Signal")
```

```python
plt.xlabel("Time [s]")
plt.ylabel("Amplitude")

# Noisy PPG Signal
plt.subplot(3, 1, 2)
plt.plot(t, noisy_ppg, label="Noisy PPG Signal", color='r')
plt.title("Noisy PPG Signal")
plt.xlabel("Time [s]")
plt.ylabel("Amplitude")

# Filtered PPG Signal
plt.subplot(3, 1, 3)
plt.plot(t, filtered_ppg, label="Filtered PPG Signal", color='b')
plt.title("Filtered PPG Signal (After Noise Removal)")
plt.xlabel("Time [s]")
plt.ylabel("Amplitude")

plt.tight_layout()
```

```
plt.show()
```

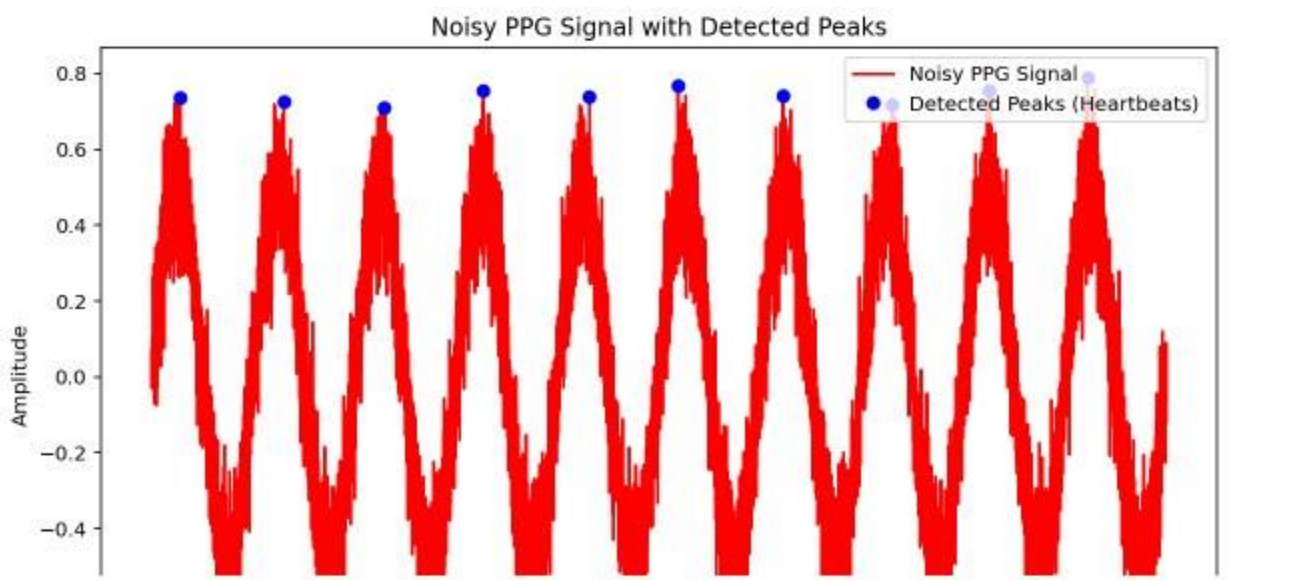


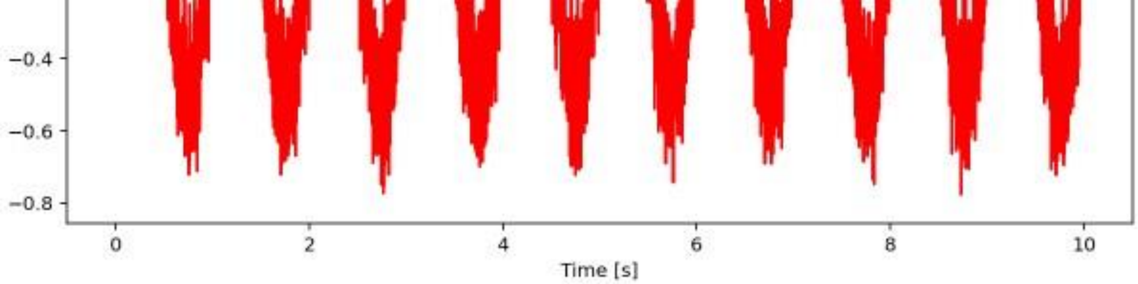



```
import numpy as np

import matplotlib.pyplot as plt

from scipy.signal import find_peaks

from scipy.stats import iqr


# Simulating a clean PPG signal (sine wave for simplicity)

fs = 1000  # Sampling frequency (Hz)

t = np.linspace(0, 10, fs * 10)  # 10 seconds of data

freq = 1  # Heart rate in Hz (1 beat per second, 60 BPM)

clean_ppg = 0.5 * np.sin(2 * np.pi * freq * t)
```

```
# Adding Gaussian noise to the PPG signal
noise = np.random.normal(0, 0.1, len(t))
noisy_ppg = clean_ppg + noise

# Detecting peaks (representing heartbeats) in the noisy PPG signal
peaks, _ = find_peaks(noisy_ppg, distance=fs/freq*0.8)  # distance set to avoid too close peaks

# Extracting Inter-Beat Intervals (IBIs)
ibi = np.diff(peaks) / fs  # Convert sample indices to time in seconds

# Heart Rate Calculation (in beats per minute)
heart_rate = 60 / np.mean(ibi)  # Beats per minute

# Extracting some statistical features:
peak_amplitudes = noisy_ppg[peaks]  # Peak amplitudes
mean_amplitude = np.mean(peak_amplitudes)
std_amplitude = np.std(peak_amplitudes)

# IQR of the peak-to-peak interval (shows variability)
peak_to_peak_intervals = np.diff(peaks)  # Interval between consecutive peaks
iqr_value = iqr(peak_to_peak_intervals)  # Interquartile range of intervals

# Plotting the noisy PPG signal with detected peaks
```

```
plt.figure(figsize=(10, 6))
plt.plot(t, noisy_ppg, label="Noisy PPG Signal", color='r')
plt.plot(t[peaks], noisy_ppg[peaks], 'bo', label="Detected Peaks (Heartbeats)")
plt.title("Noisy PPG Signal with Detected Peaks")
plt.xlabel("Time [s]")
plt.ylabel("Amplitude")
plt.legend()
plt.show()


# Display extracted features:
print(f"Heart Rate (BPM): {heart_rate:.2f}")
print(f"Mean Peak Amplitude: {mean_amplitude:.3f}")
print(f"Standard Deviation of Peak Amplitude: {std_amplitude:.3f}")
print(f"Interquartile Range (IQR) of Peak-to-Peak Intervals: {iqr_value:.3f}")
```






```
Heart Rate (BPM): 60.38
Mean Peak Amplitude: 0.743
Standard Deviation of Peak Amplitude: 0.023
Interquartile Range (IQR) of Peak-to-Peak Intervals: 60.000
```