A Self-Supervised Algorithm for Denoising Photoplethysmography Signals for Heart Rate Estimation from Wearables

Pranay Jain * Cheng Ding † Cynthia Rudin ‡ Xiao Hu§

1 Implementation Details

In this section, we describe the implementations of SPEAR's model architecture and the baseline architectures used in our experiments.

1.1 SPEAR Autoencoder Model Architecture.

The denoising autoencoder in the SPEAR algorithm uses a convolutional neural network architecture. Figure 1 illustrates the model architecture for the autoencoder. The network is summarized as follows:

- The encoder consists of 4 convolutional blocks. Each block consists of a 1D convolutional layer with ReLU activation, followed by batch normalization. Each of the encoder conv layers has a stride of 2 and zero padding.
- The encoder layers have 16, 32, 64 and 128 filters respectively. The kernel sizes are 32, 64, 128 and 320 respectively.
- The decoder network consists of 5 convolutional blocks. Each of the first 4 blocks consists of 1D convolutional transpose layer with ReLU activation followed by batch normalization. Each of the first four conv layers has a stride of 2 and zero padding.
- The decoder layers have 128, 64, 32 and 16 filters respectively. The kernel sizes are 320, 128, 64 and 32 respectively.
- The final block consists of a convolutional layer with a single filter, kernel size 3 and stride of 1. This layer uses sigmoid activation. The output of this layer is the denoised signal of the same dimension as the input.

1.2 Baseline Model Architectures

Convolutional HR Prediction Models. Baseline 3 `CNN_HR_DaLiA` and Baseline 5 `CNN_HR_Stanford` use a convolutional neural network for HR prediction. They use an identical model architecture. The model consists of two Convolutional-ReLU-MaxPool-Dropout blocks. The convolution layers are one-dimensional and have a kernel size of 9 and 64 and 32 filters respectively. The Max Pooling layer had a size of 4 and Dropout was used with probability 0.1. The two blocks are followed by a Fully Connected Layer that outputs a single HR prediction label. The training data consisted of overlapping time windows of PPG signals: 8-second windows were generated in a sliding window fashion with a 2s interval (6 s of overlap). The model was trained for 100 epochs.

Convolutional + Recurrent HR Prediction Models. Baselines 4 and Baseline 6 (`CNN+LSTM_HR_DaLiA`, `CNN+LSTM_HR_Stanford` respectively) use a combination of convolutional and Long-Short-Term-Memory (LSTM) layers. The model architecture contains 2 Convolutional-ReLU-MaxPool-Dropout blocks, with the same hyperparameters as defined for the convolutional-only models. These blocks are followed by 2 LSTM layers with a hyperbolic tangent activation function. The LSTM layers have 64 and 128 units respectively. Finally, a fully-connected layer is added, which outputs a prediction label for the HR. The model was trained for 100 epochs.

2 Ablation and Sensitivity Analysis

In this section, we perform ablation studies to verify the effectiveness of each component of SPEAR's algorithm design. We define four variants of SPEAR. `SPEAR-LSTM` has the same architecture as `SPEAR`, but

*Department of Computer Science, Duke University, USA (pranay.jain455@duke.edu).

†Department of Biomedical Engineering, Georgia Institute of Technology & Emory University, USA.

‡Department of Computer Science, Duke University, USA.

§Professor and Asa Griggs Candler Chair of Nursing Data Science, Associate Director — Center for Data Science, Nell Hodgson Woodruff School of Nursing, Associated Faculty of Biomedical Informatics, School of Medicine, Associated Faculty of Computer Science, College of Arts and Sciences, Emory University, USA.

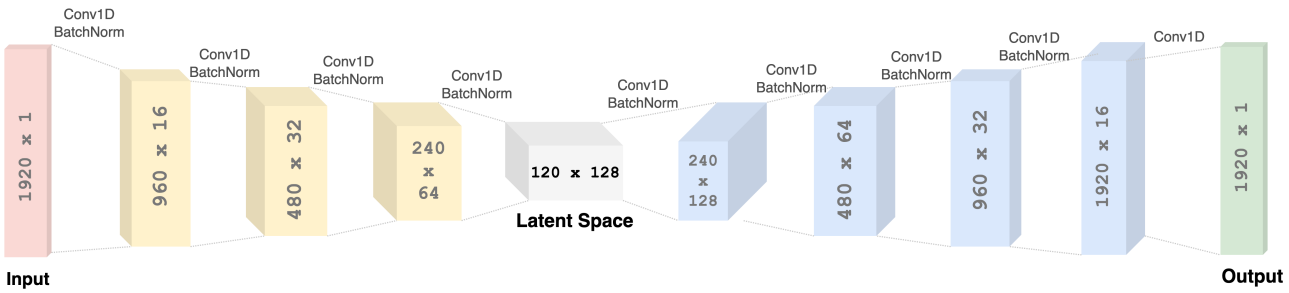

Figure 1. Model Architecture for the denoising autoencoder used in SPEAR.

Model	DaLiA	Stanford
SPEAR-LSTM	5.45	4.46
SPEAR-N	5.75	3.65
SPEAR-N	5.85	4.13
SPEAR-L	10.84	10.52
SPEAR	5.36	3.18

Table 1. Ablation & Sensitivity Analysis Results.

adds an LSTM layer in the encoder network. Since LSTM-based architectures had superior performance in the HR estimation baselines (`CNN+LSTM_HR_DaLiA` and `CNN+LSTM_HR_Stanford`), this comparison baseline was used to see if any similar improvements would be found in our denoising model as well. `SPEAR-N` is trained such that the artifact locations are replaced with gaussian noise instead of setting it to 0. In this case, the autoencoder does not receive a 0 signal at the location of the artifact, so it reconstructs the full signal, not just the corrupted regions. `SPEAR-Sm` follows the same training procedure as SPEAR but it uses smaller kernel sizes of convolutional layers in the model architecture. `SPEAR-L` removes the first two convolutional layers from the encoder and last two layers from the decoder in SPEAR's model architecture. The results are shown in Table 1, indicating that SPEAR is not sensitive to changes in kernel size or gaussian noise in the input, but it is sensitive to major ablations such as removal of the convolutional layers. We also see that adding an LSTM layer to the SPEAR architecture does not offer any improvements, and as such does not offer the same benefits of added complexity that were seen in the HR estimation baselines.

3 Denoising Results

In this section, we provide denoising results from the SPEAR algorithm. Figures 2 - 9 display these results. Results are provided for out-of-sample signals obtained from the PPG DaLiA and Stanford test sets.

- Each PPG signal is 30 seconds long and sampled at 64 Hz.
- The first signal in each result is the original signal from the dataset. These signals contain some noise artifacts, which are highlighted in red (noise regions are predicted by the Segade model).
- The second signal in each result is the denoised signal produced by SPEAR. It can be observed that the signal is only reconstructed in the noise-corrupted regions, the rest of the signal is unaltered.
- The third signal in each result visualizes the denoised signal after bandpass filtering and peak detection. Bandpass filtering makes the signal smoother which improves the performance of peak detection algorithms. The green dots in the signal show the peaks that were detected by the peak detection algorithm.
- For the examples from the PPG DaLiA dataset, the fourth signal shows the synchronously recorded ECG signal, which is used as ground truth. These signals are included to demonstrate the efficacy of the peak detection on the denoised signal.
- Note that in some cases the peaks of the uncorrupted regions may appear larger in the denoised version than the original – this is because the artifacts in the original signal may have greater amplitude, which causes the rest of the signal to have ostensibly smaller peaks. When these artifacts are removed, the peaks in the clean signal appear larger.
- The denoising examples from the stanford dataset consist of signals that are corrupted with real noise artifacts. This is in contrast with the Stanford experiment in Section 4.3, where simulated noise

was added to generate clean-noisy pairs. For visualization purposes, we demonstrate real noisy signals from the stanford dataset.

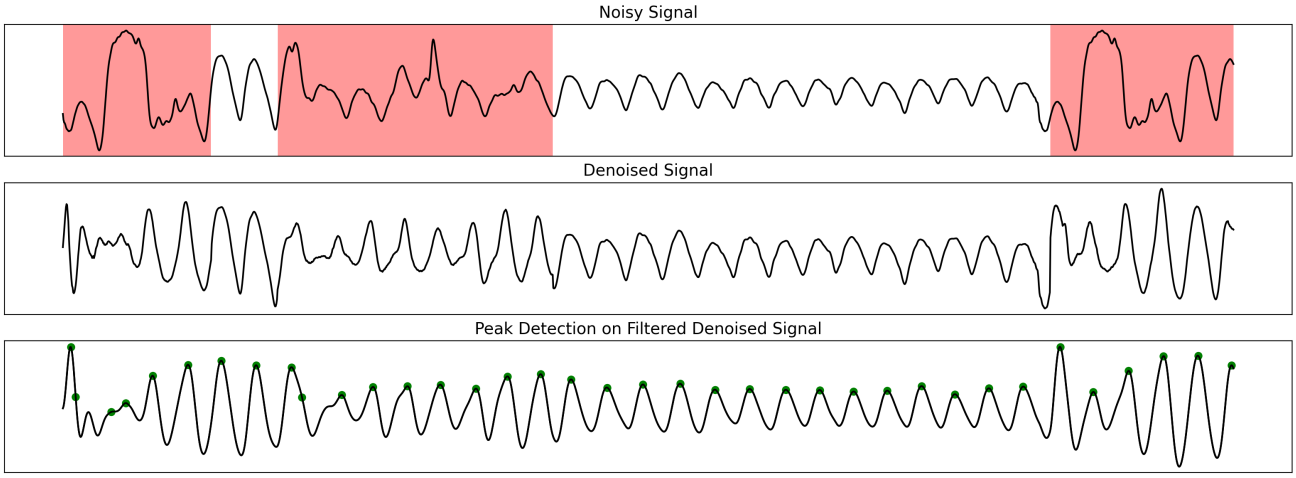

Figure 2. Example 1 of denoising from Stanford test set.

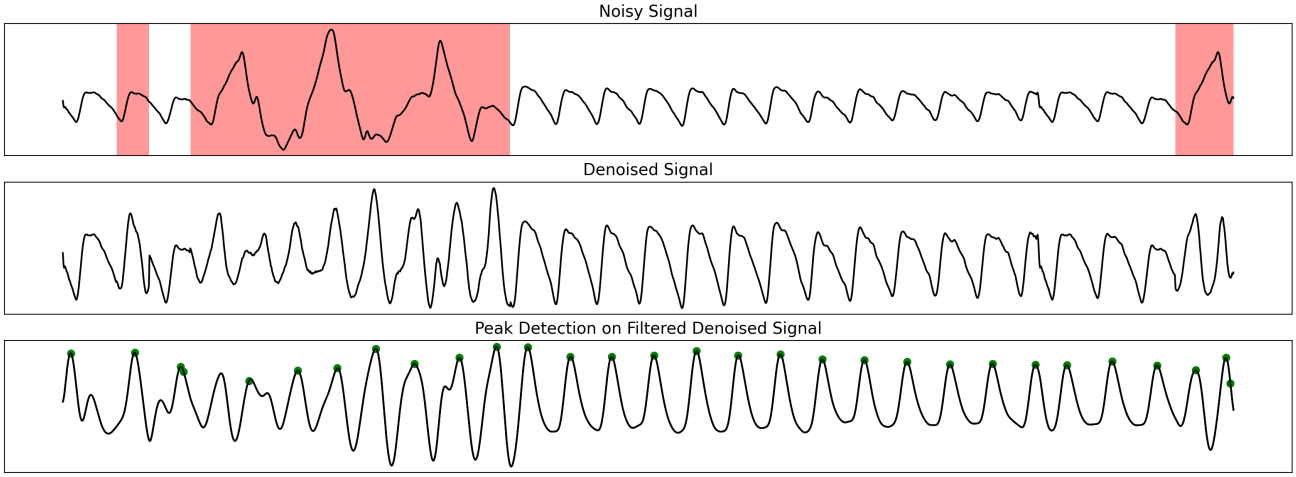

Figure 3. Example 2 of denoising from Stanford test set.

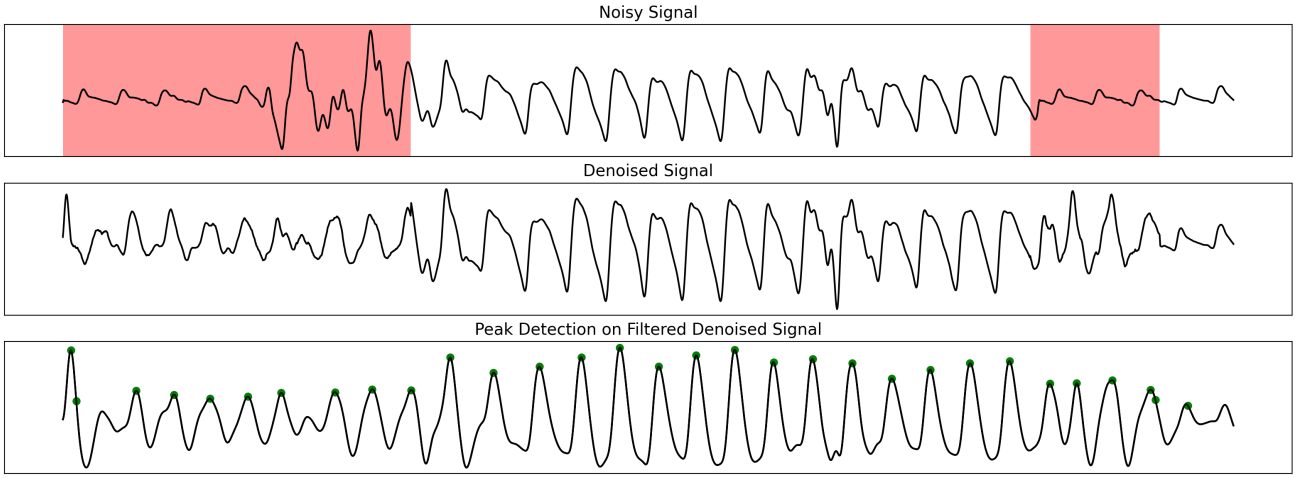

Figure 4. Example 3 of denoising from Stanford test set.

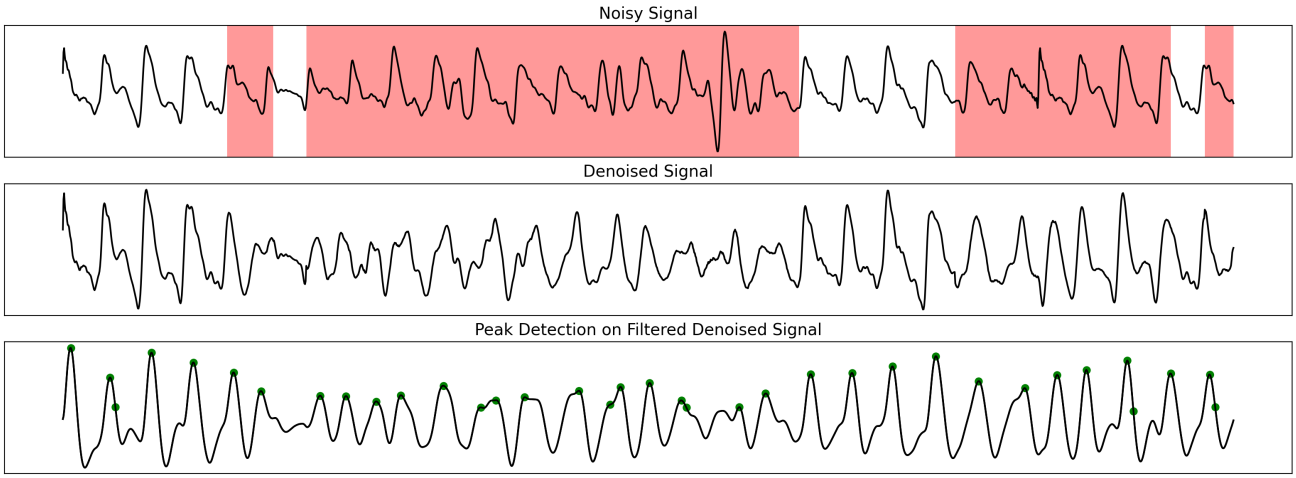

Figure 5. Example 4 of denoising from Stanford test set.

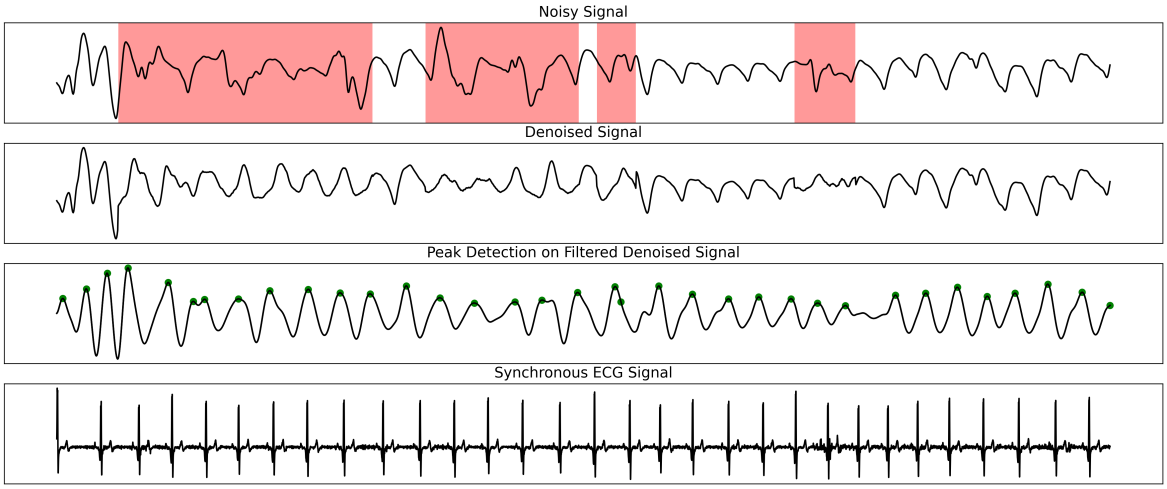

Figure 6. Example 1 of denoising from PPG DaLiA dataset.

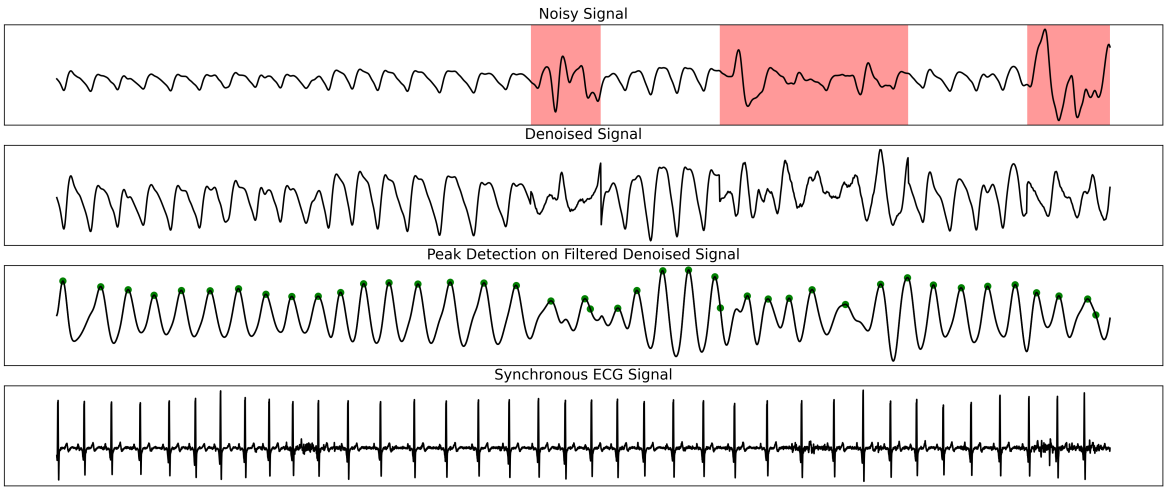

Figure 7. Example 2 of denoising from PPG DaLiA dataset.

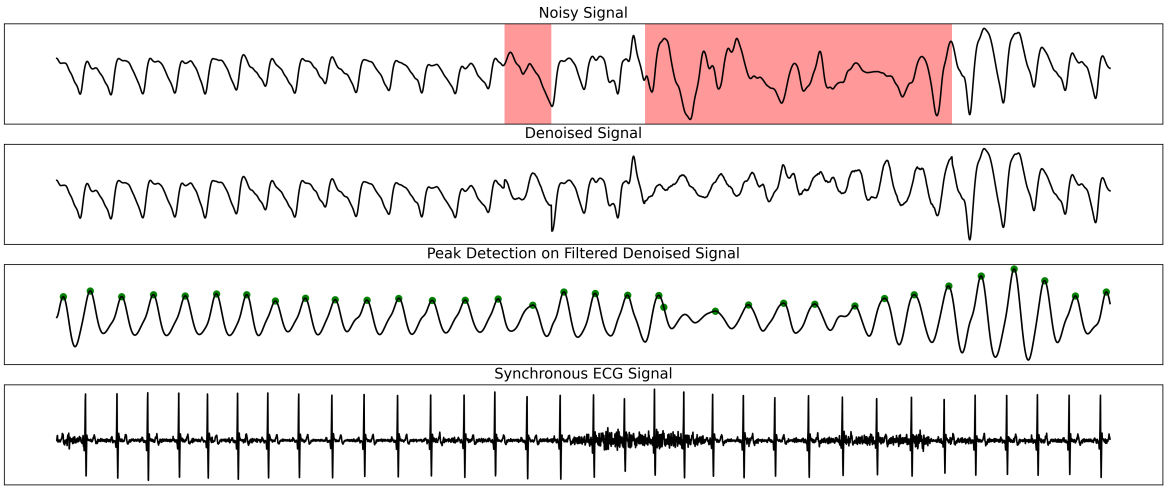

Figure 8. Example 3 of denoising from PPG DaLiA dataset.

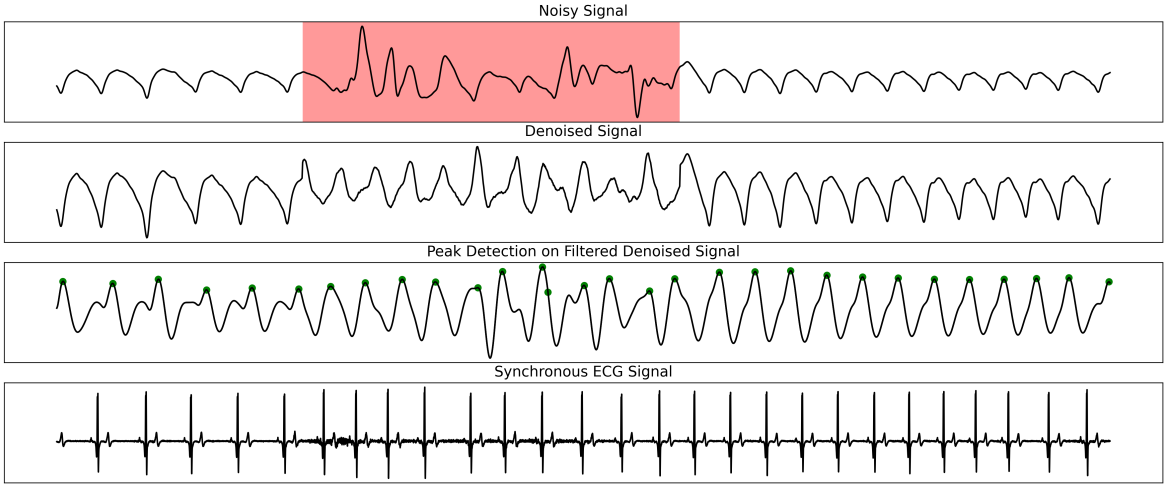

Figure 9. Example 4 of denoising from PPG DaLiA dataset.