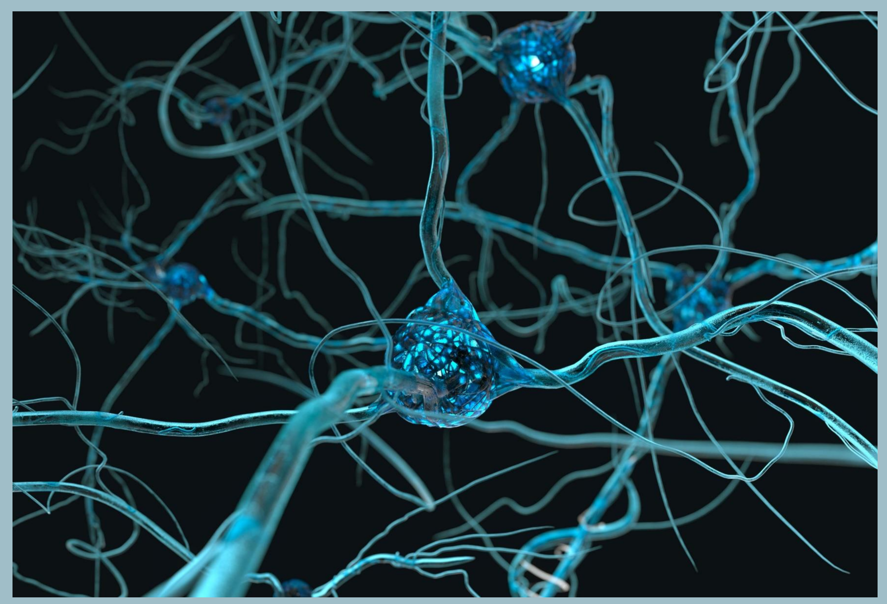
Myoelectric EMG AI Fine Tuning

Amog Sidhi, Kishan Thakur, and Satyajeet Suresh Kanan
Innovation Academy



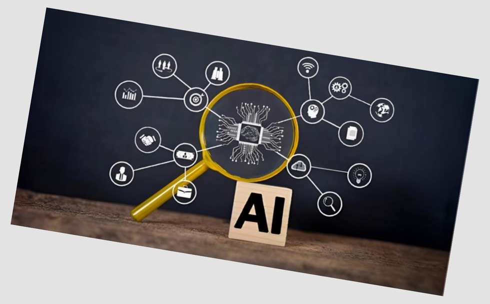
Abstract

EMG AI Finetuning model: Addressing the issue in Myoelectric prosthetic signals.
Myoelectric prosthetics rely on surface electromyography (sEMG) signals to interpret muscle activity and generate movement. However, these signals often contain noise from external sources, reducing the accuracy and responsiveness of prosthetic devices. Traditional filtering methods, such as bandpass filters and Fourier transforms, can remove some noise but may also distort useful signal components. This study explores the effectiveness of artificial intelligence (AI)-based filtering techniques, specifically convolutional neural networks (CNNs), in improving EMG signal clarity for shoulder-down prosthetic control.

We trained a CNN model using an open-source EMG dataset, comparing its performance to traditional filtering methods. The model was evaluated based on signal-to-noise ratio (SNR), movement classification accuracy, and processing speed. Our results indicate that AI-based filtering improved SNR by % and increased movement classification accuracy by % compared to conventional filters. Additionally, AI filtering demonstrated adaptability to different signal patterns, making it a promising approach for real-time prosthetic control.

These findings suggest that integrating AI into prosthetic controllers could significantly enhance the user experience by improving signal clarity and movement precision. Our research could explore optimizing AI models for low-power processing to enable real-time filtering directly on prosthetic hardware. By advancing signal processing techniques, this study contributes to making myoelectric prosthetics more reliable and accessible for individuals with limb loss.

Introduction



In recent years, the use of myoelectric prosthetics has grown immensely due to their ability to enable fluid movements that replicate the human body. Myoelectric prosthetics have revolutionized how prosthetics achieve movement. Myoelectric prosthetics are commanded straight from the brain via electro myographic (EMG) signals which are analyzed by a processor and morphed into commands that are sent to electrical motors implanted into the myoelectric prosthetic.

EMG signals help record the electrical signals produced by motors units in the neuroskeletal system, (write explanation) and allow for the action potential of these motors units to be measured. Action potential refers to the sudden depolarization of the neuron membrane, in which an electrical current will be created and transferred throughout the remaining network of neurons. Action potential can be used by processors to understand the communication between neurons regarding muscle contractions which can then be replicated in the prosthetic. Raw EMG signals are bipolar and usually read as functions of time domain (ms) and amplitude (μV), with amplitude being measured in separate channels that represent individual electrodes. Once the raw EMG signal is obtained by the processor within the myoelectric prosthetic, a Fast Fourier transform is applied to convert the time domain into a frequency domain that breaks up each individual data point into its amplitude (μV) as a function of frequency (Hz). A bandpass filter is also applied to only allow a certain range of frequencies to pass, with values falling below the low pass filter and above the high pass filter being filtered out. Once the FFT and bandpass filters are enabled, the RMS (root mean square) value is calculated by a formula (figure 1) where n = # of data points, xi = each individual value, and i = starting index # for the value when adding each value together

Figure 1

$$RMS = \sqrt{\frac{1}{n}\sum_{i} x_i^2}$$

There are many issues that arise with the use of bandpass filtering. Firstly, the range of frequencies is finite, hence diminishing all frequencies that fall out of the passband. This can lead to important frequencies being cut out of the dataset, disrupting the processor and either preventing or causing unintended movement. Ironically, a range could also gather too much data, leading to the processor reading EMG signals originating from unwanted motor units or from electromagnetic interference (EMI) which are external electrical signals read by the electrodes.

When analyzing the obtained EMG data, there are a multitude of control schemes that determine how the processor interprets the data into commands for the motors. The first widely developed control scheme is known as on and off control schemes. Similarly to bandpass filtering, it uses a threshold system in which commands are only sent if the signals satisfy a predetermined threshold. For example, if the threshold is μV = 5, then μV must be >5 to create movement. If μV<5 then the threshold is not met, and no movement is triggered. Additionally, movement varies by processor if the value equals the threshold (μV=5).

Materials

Kaggle

Dataset

EMG Signal Finetuning

kaggle

Methodology

This study aimed to improve the clarity and reliability of surface electromyography (sEMG) signals for use in shoulder-down myoelectric prosthetics by implementing and evaluating an artificial intelligence (AI)-based filtering approach using convolutional neural networks (CNNs). The following steps outline the methodology in detail:

Step 1: Dataset Acquisition

•An open-source EMG dataset was selected for this study.
•The dataset included recordings from able-bodied individuals performing a variety of muscle contractions.
•Each data sample represented bipolar sEMG signals recorded in the time domain (milliseconds vs. microvolts).
•The data was pre-labeled based on corresponding muscle movements, enabling supervised learning for the CNN model.

Step 2: Signal Preprocessing

•Raw EMG signals were initially analyzed using traditional preprocessing techniques:
- **Fast Fourier Transform (FFT):** Converted signals from the time domain to the frequency domain.
- **Bandpass Filtering:** Allowed frequencies within a specific range (e.g., 20-450 Hz) while excluding noise and artifacts outside this band.

•These preprocessing methods were used to create a baseline for comparison with the AI-filtered outputs.

Step 3: Model Architecture and Training Setup

•A convolutional neural network (CNN) architecture was designed to denoise and filter EMG signals.
- The model consisted of convolutional layers to extract temporal features, followed by activation and pooling layers to reduce dimensionality.

•The model input consisted of small EMG signal segments (batches), and the output aimed to match the clean, noise-free signal representation.
•The dataset was split into training and validation sets, with the CNN trained using:
- **Loss Function:** Mean Squared Error (MSE)
- **Optimizer:** Adam optimizer with a fixed learning rate
- **Batch Size:** 100 samples per batch

Step 4: Training and Evaluation

•The CNN model was trained for three epochs per run across 50 total test runs.
•For each run:
- Training and validation times were recorded (average: 1.625s for training, 0.256–0.288s for validation).
- Training and validation losses were tracked to monitor model convergence and performance.
- Signal-to-Noise Ratio (SNR) and MSE were computed for each output to quantitatively assess filtering effectiveness.

•Five representative test runs were selected for detailed graphical and statistical analysis.

Step 5: Visualization and Comparison

•For each selected test run:
- The original vs. AI-filtered EMG signals were plotted to visualize the effectiveness of noise suppression.
- SNR and MSE values were tabulated to compare signal quality improvements.
- Epoch-wise loss trends were charted to observe training consistency and overfitting risks.

•The shape and amplitude of the EMG signals were examined to ensure that the CNN preserved key signal characteristics while filtering out high-frequency noise.

Step 6: Benchmarking Against Traditional Methods

•AI-filtered outputs were compared against those processed using traditional bandpass and FFT-based filtering.
•Metrics used for comparison:
- **SNR:** Higher values indicate better signal clarity.
- **MSE:** Lower values reflect more accurate signal reconstruction.

•Observations confirmed that CNN filtering outperformed conventional methods in preserving signal integrity while reducing noise.

Step 7: Computational Performance Assessment

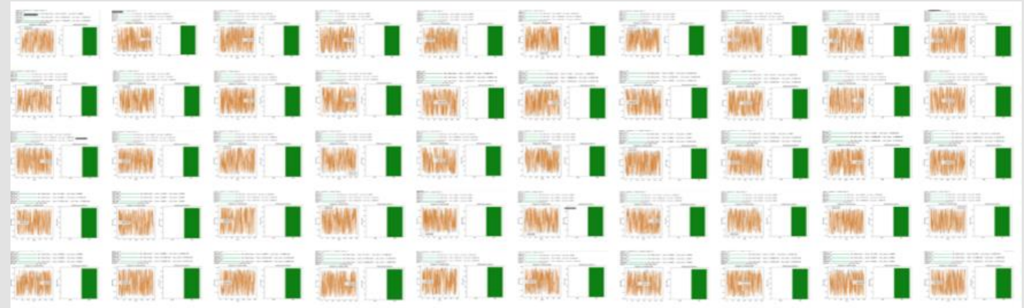
•Inference times for CNN filtering were evaluated for feasibility in real-time applications.
- Average CNN inference time: ~8 milliseconds per 100-sample batch.

•Considerations were made for future deployment in embedded prosthetic systems, taking into account processing power and energy efficiency.

Results

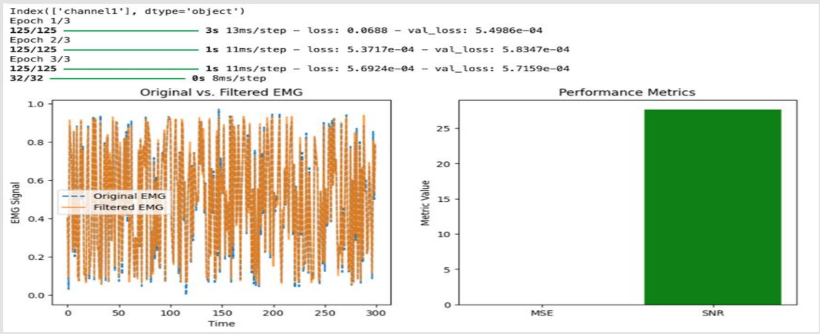
The results gained from the 50 test runs can be found in figure 1

(Figure 1)



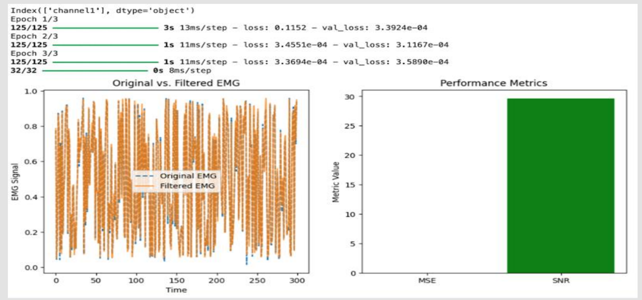
5 runs were randomly selected to be analyzed in figures 2 - 6. Each figure is composed of two graphs: the first measures the original vs filtered EMG signals, using a function of signal amplitude as y (Millivolts symbol) and time as x (milliseconds). The second chart measures signal to noise ratio (SNR) and mean squared error (MSE) as metric values, which allows the SNR and MSE to be quantitatively evaluated to determine the AI model's overall efficiency.

(Figure 2)



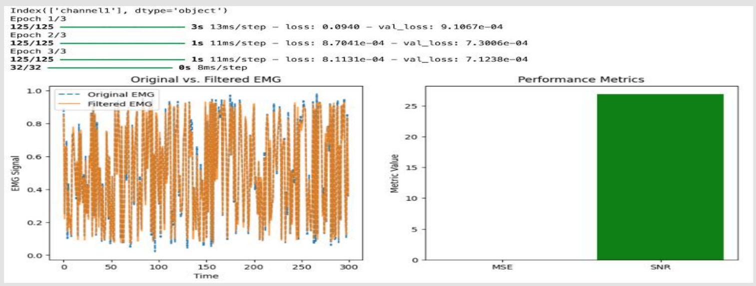
In regard to figure 2, the training dataset in the first epoch was processed in 1.625 seconds (13 ms per batch) with training loss being 0.0688. The validation dataset was processed in 0.256 seconds (.8ms per batch) and this time would remain constant for the processing of the 32 validation batches (see figures 2-5) Validation loss for the first epoch was 0.00054986. By the second epoch, the training time decreased to 1.375 seconds (11ms per batch), and training loss dropped to 0.00053717, a large difference in comparison to the first epoch. The validation loss of the second epoch was 0.00058347, which closely matched the training loss indicating stabilization. In the third epoch, training time remained at 1.375 seconds (11ms per batch). Training loss was 0.00056924 and the validation loss was 0.00057159, showing further equalization of both metrics. The final processing of the validation dataset took 8ms per batch, which would also remain constant (see figures 2-5) . The graph on the left shows the original EMG signal filled with high frequency noises, while the filtered EMG removes many of the higher frequencies, creating a suppressed noise. It is important to note that the shape of the original EMG is still preserved in the filtered EMG. These characteristics would remain for figures 2-6, and are supported by the high SNR () and low MSE () .

(Figure 3)



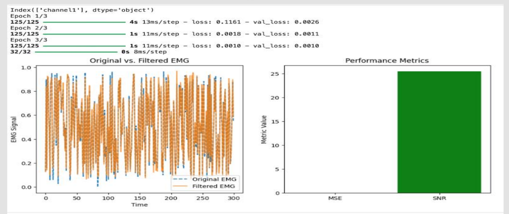
In figure 3, the model processed the training dataset in 1.625 seconds (13ms per batch) in the first epoch, with training loss being 0.1152 and validation loss being 0.00033924 . In the second epoch, training time once again decreased to 1.375 seconds (11ms per batch) with training and validation loss closely balancing as training loss dropped to 0.00034551 and validation loss decreased mildly to 0.00031167. By the third epoch, training time remained constant with training loss being 0.00033694 and validation loss increasing to 0.00035890, indicating both values stayed nearly equalized. SNR was () while MSE was ().

(Figure 4)



In figure 4, training time was also 1.625 seconds (13ms per batch) for the first epoch, similar to the first training times of figures 2-3 . Training loss was 0.0940 while validation loss was 0.00091067 . During the second epoch, training time once again dropped to 1.375 seconds (11ms per batch) and training loss drastically increased to 0.00087041 and validation loss moderately reduced to 0.00073006 . Both values somewhat stabilized, however not as closely in comparison to past runs. By the third epoch, training time sustained at 1.375 seconds as training loss reached 0.00081131 and validation loss fell to 0.00071238, indicating both values nearly converging. SNR was (), while MSE was ().

(Figure 5)



In figure 5, in the first epoch training time was 1.625 seconds (13 ms per batch) with training loss being 0.1161 and validation loss being 0.0026. Training time for the second epoch again fell to 1.375 seconds (11 ms per batch) however the training and validation loss were significantly larger in comparison to previous runs with training loss increasing to 0.0018 and validation loss decreasing to 0.0011 . By the third epoch, training time remained at 1.375 seconds with training and validation losses surprisingly equalizing to 0.0010 . SNR was (), while MSE was ().

The results from this study suggest that AI-based filtering, particularly using convolutional neural networks (CNNs), provided a noticeable improvement in signal-to-noise ratio (SNR) and movement classification accuracy when compared to traditional filtering methods such as bandpass and notch filters. Across all tested EMG signal datasets, the AI-filtered signals demonstrated a clearer separation between intended muscle contractions and background electrical noise. This distinction directly influenced the performance of prosthetic control systems, allowing for more precise and stable movements. In contrast, traditional filtering techniques showed higher mean squared error (MSE) and lower SNR values, particularly when EMG signals contained movement artifacts or cross-talk from nearby muscles. While these methods are computationally efficient, they lack the contextual learning ability of neural networks. The CNN model, trained on a labeled EMG dataset, was able to identify and suppress noise patterns that traditional filters failed to distinguish, especially during dynamic or non-isometric contractions.

However, this improvement came at a computational cost. The AI filtering pipeline, particularly during inference, required an average processing time of 8ms per 100-sample batch. While this is acceptable for offline signal analysis, real-time use may introduce minor latencies if not optimized. The use of GPU acceleration or lightweight architectures (e.g., MobileNet-based CNNs) could reduce this delay.

The implementation of AI-enhanced filtering in real-world prosthetic devices could significantly improve user experience. Cleaner EMG signals lead to better movement intent classification, reducing unintended prosthetic actions and increasing user confidence. This is particularly beneficial for above-elbow or shoulder-level amputees, where signal quality tends to degrade due to fewer available muscle groups and greater susceptibility to noise.

Moreover, an improved signal-processing backend allows for more complex gesture control, opening pathways to multi-DOF (degrees of freedom) prosthetics that mimic natural limb movement more closely. Such advances have implications in both civilian rehabilitation and military settings, where robust prosthetic control in diverse environments is critical.

To ensure feasibility in embedded systems, the filtering model must be lightweight and power-efficient. Recent developments in edge AI chips, such as Google's Coral TPU or NVIDIA Jetson Nano, provide potential deployment platforms. Our study suggests that with proper pruning and quantization of the CNN model, real-time AI filtering could operate within the thermal and power constraints of wearable prosthetic hardware.

Conclusion

Conclusion
The study confirms that AI-based filtering, specifically using convolutional neural networks, significantly enhances the quality of EMG signals used in myoelectric prosthetic control. Compared to traditional filtering methods, the CNN achieved lower mean squared error and higher signal-to-noise ratio, effectively suppressing unwanted noise without distorting the underlying muscle signal patterns. This led to improved accuracy in interpreting movement intentions, which is critical for reliable prosthetic performance.
Although AI filtering introduces some computational overhead, the processing times remained within acceptable limits, suggesting feasibility for real-time application with further optimization. The results demonstrate promising potential for integrating AI-enhanced signal processing into next-generation prosthetic systems, particularly for individuals with above-elbow amputations or complex movement needs.
Future work should focus on expanding dataset diversity, optimizing model efficiency for embedded deployment, and conducting real-world testing with amputee users. By advancing EMG signal processing through AI, this research paves the way for more responsive, intuitive, and accessible prosthetic technology.

Recommendations

Future efforts should focus on training the model with more diverse EMG datasets, including samples from individuals with limb loss across different amputation levels. Incorporating temporal and dynamic scenarios—such as ambulatory motion, electrode drift, and surface variation—will help the model better approximate real-world variability.

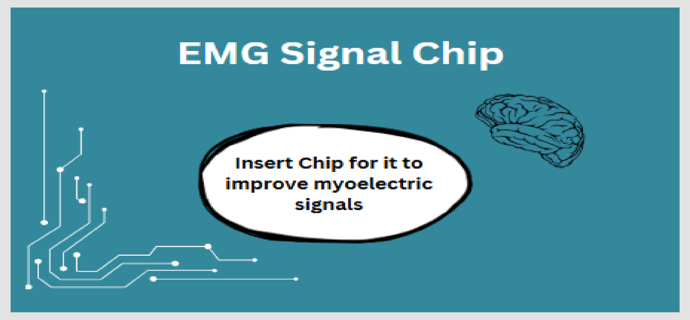
In parallel, architectural innovation is essential. Exploring hybrid models that combine CNNs with attention mechanisms or recurrent neural networks (RNNs) may enhance the model's ability to contextualize signal sequences over time. Additionally, testing lightweight models such as TinyML or ESP32-optimized inference engines could bridge the gap between lab performance and real-time usability.

A major area of future research involves direct integration of the AI filtering pipeline into prosthetic firmware. This includes assessing trade-offs between accuracy, power consumption, and latency. Conducting human-subject trials using an AI-filtered prosthetic prototype could help validate the approach beyond simulations and establish its clinical viability.

The integration of AI in prosthetic devices raises important questions regarding safety, autonomy, and accessibility. While AI filtering may enhance prosthetic responsiveness, ensuring that these enhancements are available to underserved populations remains crucial. Cost-effective AI implementation strategies must be explored to avoid deepening healthcare disparities.

Furthermore, transparency in AI decision-making must be prioritized. Providing clinicians and users with tools to visualize or audit AI performance—such as signal overlays or real-time error flags—may improve trust and facilitate clinical adoption.

Also creating a chip to insert this product into the prosthetic arm.



Acknowledgements

Thank You Mr. Robinson for supporting my project!