MANE 4962 Final Report

PREDICTING USER INTENT FROM sEMG SIGNALS

Paul Khoury

Rensselaer Polytechnic Institute

Prof. Chowdhury

April 23rd, 2025

Executive Summary

This project developed a machine learning system that can accurately interpret muscle signals (EMG) to recognize hand gestures, with the goal of being applied to control of robotic prosthetics. Unlike conventional prosthetics that require manual setup and have limited movement capabilities, our approach uses modern machine learning to create more responsive artificial limbs.

Current commercial prosthetic systems face three key limitations that restrict their effectiveness. They typically require extensive manual calibration, which is time-consuming and must be repeated frequently. They offer limited movement capabilities compared to natural human motion. Additionally, they often don't adapt well to different users, making them less accessible. Our solution addresses these issues by quickly recognizing a user's intended hand movements directly from muscle signals measured on the skin surface.

We analyzed a comprehensive dataset collected from 36 individuals wearing a commercial EMG bracelet while performing six basic hand gestures. After processing millions of raw data points, we trained a model to recognize which signals correspond to which gestures. First, we extracted meaningful statistical features (similar to averages) from the electrical signals to make the data more manageable. Next, we tested different machine learning approaches to classify the hand gestures accurately. Finally, we compared performance between traditional methods and neural networks to determine the most effective solution.

Our neural network model was 94% accurate in recognizing predicting which hand gesture a muscle signals corresponded to. This performance was better than multiple published benchmarks and our first models, demonstrating the effectiveness of our approach. We discovered that even by using relatively simple signal features, we could produce accurate results. We confirmed which signal features were most effective for this specific application, with Mean Absolute Value (MAV) providing surprisingly strong results.

This research implies that machine learning is a good solution for signal recognition in prosthetic controls. Overall, research like this is bringing artificial limb functionality closer to natural human capabilities.

1. Introduction

The primary objective of this project is to develop a machine learning (ML) model capable of accurately interpreting surface electromyographic (sEMG) signals to classify user intent, with applications in smart prosthetic control. sEMG signals capture muscle activation potentials non-invasively through electrodes placed on the skin. They offer a promising but complicated means of decoding voluntary limb movement. This technology forms the foundation for “smart” prosthetics, enabling artificial limbs that can truly act as natural biological limbs as far as actuation is concerned.

This research addresses key limitations in today's commercial prosthetic systems, which typically need manual setup, have limited movement capabilities, or don't work well for different users. Using supervised learning and improved signal analysis methods, this project develops an automated approach to recognize specific hand gestures from raw muscle signals. While earlier studies have shown good results with custom-designed features and simpler classification methods, this work compares traditional and newer machine learning approaches, tests how well they work across different situations, and carefully reviews assumptions made in previous research.

The ultimate goal, which is beyond the scope of this report, is contributing to the development of systems that can work in real-time and accommodate the full range of human motor abilities. If fully realized, this technology could seamlessly interpret “thoughts” controlling the performance of fine motor actions such as typing or sewing.

A comprehensive literature review informed key aspects of the study design. Kok et al. evaluated multiple feature extraction methods (MAV, RMS, DWT) and compared different classification models (SVM, KNN, Naïve Bayes) for anatomical motion prediction. Their work demonstrated the efficacy of statistical features like root mean square (RMS) and mean absolute value (MAV) for EMG classification, reinforcing the decision to apply time-domain statistical measures for collapsing temporal structures into formats suitable for time-independent classifiers.

Shakya and Ranjitkar explored convolutional neural networks (CNNs) and principal component analysis (PCA) for enhancing pattern recognition in forearm biomedical signals, providing insights into using multiple data forms to train classifiers. Their work with the GRABMyo dataset offered a useful reference point for data collection methodology.

Additionally, a review by Mhiriz et al. synthesized current challenges in sEMG-based classification, including inter-subject variability and noise robustness. Their paper addressed common issues like low signal-to-noise ratios and suggested methods to combat these problems using various models, including Support Vector Machines and Random Forests. This comprehensive overview guided preprocessing strategies and model selection for the project.

The dataset used originated from the UC Irvine Machine Learning Repository and was collected via the Myo Thalmic Bracelet, a commercially available (company is now defunct) sEMG acquisition device with eight EMG channels and consistent electrode placement. Ample data was collected and provided for analyses.

The project began with thorough data preparation and cleaning. Millions of raw data points were refined into a more manageable dataset by removing noise and calculating statistical features from each channel. The data was then standardized and visualized to better understand the patterns among different hand gestures.

Starting with a support vector machine model as a baseline, the project evaluated performance using standard accuracy metrics. This approach drew on techniques identified in the literature review. Later, additional features were tested and performance was finally compared with a neural network model.

Results showed the neural network achieved 94% accuracy, outperforming both the published benchmark and the baseline model. These findings confirm that even relatively simple statistical features from muscle signals can produce highly accurate gesture classification when paired with the right models. However, questions about how well these models work across different people remain open and will be explored in future work.

This project demonstrates the feasibility of creating intelligent prosthetic control systems using machine learning to interpret sEMG signals. The report details the process and the results.

2. Problem Definition

The problem addressed in this project is the reliable identification of the gesture that a patient intends to make based only on surface EMG signals. The primary challenge here is to interpret the user's sEMG signal which are complex, noisy, and highly variable. This task is not feasible using hardcoded rules or simple statistical tests, as the data is multi-dimensional, user-specific, and does not have easily defined boundaries. Thus, machine learning is required to detect non-obvious patterns and to generalize across gesture types and individuals.

Older generations of prosthetic limbs that respond to EMG signals often rely on simplistic threshold-based approaches or require painstaking calibration for each individual, which limits their effectiveness. By using ML, particularly supervised classification techniques, this project seeks to automate gesture recognition with greater precision and minimal setup. For this course, the scope of the project will be limited – in the interest of realistic goal-setting – such that the aim is to successfully classify sEMG signals into one of 6 predefined static hand/wrist positions that correspond to the sEMG waveforms. The 6 hand gestures are as follows:

1. Hand at rest
2. Fist clench
3. Wrist flexion
4. Wrist extension
5. Radial deviation
6. Ulnar deviation

3. Methods and Procedure

I. Data Aggregation

The original dataset consisted of 72 text files (two per subject across 36 individuals), each representing time-series sEMG recordings from eight channels. Every row had a corresponding gesture label which was assigned on-site and real-time (concurrently to data collection in the lab).

The first step in making the data usable and remotely convenient for this project is that all .txt files should be converted into a csv file, which makes them much easier to utilize and manipulate with python. The first step for that would be to place all the text files in a single folder. Then, using a custom program (txttocsvmaster.py script located in MANE-4962-FINAL-PROJECT/notebooks), combine all the text files into one 'master' file in CSV format. This program also ensures all the formatting is consistent that any improperly formatted data is discarded. It is worth noting that in the case of this dataset, this accounts for much less than 1 percent of the total data.

II. Data cleanup and Feature Extraction

Next, the final_data_processer.py script should be executed. This script performs multiple preprocessing tasks: First, it removes invalid or noisy gesture labels (data labeled 0, indicating unmarked data, and data tagged with a 7, which is a gesture that not all subjects performed. This inconsistency was not addressed in the original paper or the ReadMe.)

> *Note: This stage would have been an ideal point to apply signal processing filters—specifically, a Notch filter to suppress power-line interference, along with low-pass and high-pass filters to reduce muscular noise and other signal artifacts. However, upon reviewing my project, I realized that I inadvertently skipped this filtering step in the final classification. Although the final model still performed well, it's possible that classification accuracy could have been further improved with proper filtering. Therefore, the author strongly recommends that anyone attempting to reproduce or extend this study should not omit this step during preprocessing.*

Next, the script computes three statistical features—Root Mean Square (RMS), Mean Absolute Value (MAV), and Slope Sign Changes (SSC)—for each instance of an uninterrupted classified gesture in each signal channel. This reduces the dimensionality of the data by making it time-independent. Effectively, it takes the several seconds worth of signal data (comprising thousands of data points) and summarizes it in a statistical measure. This is repeated for every occurrence of each gesture (dozens to hundreds per subject since gestures were repeated over the course of data collection for each subject.

As a consequence of this preprocessing, the output is a cleaned, time-independent dataset suitable for classification. This process reduces the original >36 million data points down to ~15,000 samples while preserving the essential information.

Each data point in the final processed dataset consists of a feature vector, which is composed of continuous, numerical values. The length of each feature vector can vary—typically containing 8, 16, or 24 values per sample. This variation depends on the combination of statistical features selected for a given model run. For example, using only the Root Mean Square (RMS) feature results in 8 values (one for each EMG channel), while including Mean Absolute Value (MAV) and Slope Sign Changes (SSC) adds additional sets of 8 values each, bringing the total to 16 or 24 features, respectively.

These numerical feature vectors are each paired with a categorical target label, an integer which represents the intended hand gesture. The target variable is a discrete integer value ranging from 1 to 6, corresponding to one of six predefined static gestures defined in the problem section. This label serves as the classification target for the machine learning models.

III. Standardization

All features were scaled using Scikit-learn's StandardScaler, normalizing them to have zero mean and unit variance. This step is highly recommended for SVMs and neural networks, since SVM could be sensitive to input scaling.

IV. Classification

Two classification pipelines were developed during the course of this project. The first was a Support Vector Machine (SVM) classifier, implemented using Scikit-learn. This choice was motivated by existing literature, which highlighted the success of SVMs in EMG signal classification due to their ability to handle high-dimensional and non-linearly-separable data. The model was evaluated using accuracy, precision, recall, and a confusion matrix.

The second and more advanced pipeline involved a fully connected neural network, built using Keras. This feedforward model is described in greater detail in the Results and Discussion section of this report. The neural network was trained on the same preprocessed dataset and achieved a peak accuracy of 94%, outperforming the SVM. This confirmed that NNs could offer improved performance.

V. Methods and Procedure Summary

All scripts and notebooks are organized in the MANE-4962-FINAL-PROJECT directory. They include:

- txttocsvmaster.py for data aggregation
- final_data_processer.py for filtering and feature extraction
- proj_prog_4_initial_classification.ipynb for SVM training
- final_classification.ipynb for neural net classification
- Workflow.txt provides a useful high-level overview of the steps to take in order to recreate this project.

4. Dataset and Visualization

Several techniques were used to visually inspect the data. While it is important to emphasize that this did *not* equate to classification or impact the data nor the results in any meaningful way, it was useful for the experimenter to develop an intuition about the data. As mathematics communicator Grant Sanderson once said, “The space of all possible logical moves you can make [in solving a problem] is often too vast to explore in a reasonable time. Intuition is what offers the guiding lights telling you which paths are worth trying.” Exploratory visualization served as that guiding light throughout this project.

t-SNE plots provided a two-dimensional representation of the high-dimensional feature space, offering some insight into how well the gesture classes could be separated. Despite the inherent variability and noise in EMG signals, these plots revealed there to be a surprising amount or underlying structure in the data. However, this comes with a caveat; it's important to note that t-SNE is a non-linear, stochastic algorithm, and while it is useful for clustering, it does not preserve global distances or scale. It also varies with the seeding. Thus, it should be interpreted qualitatively, rather than as a definitive measure of separability.

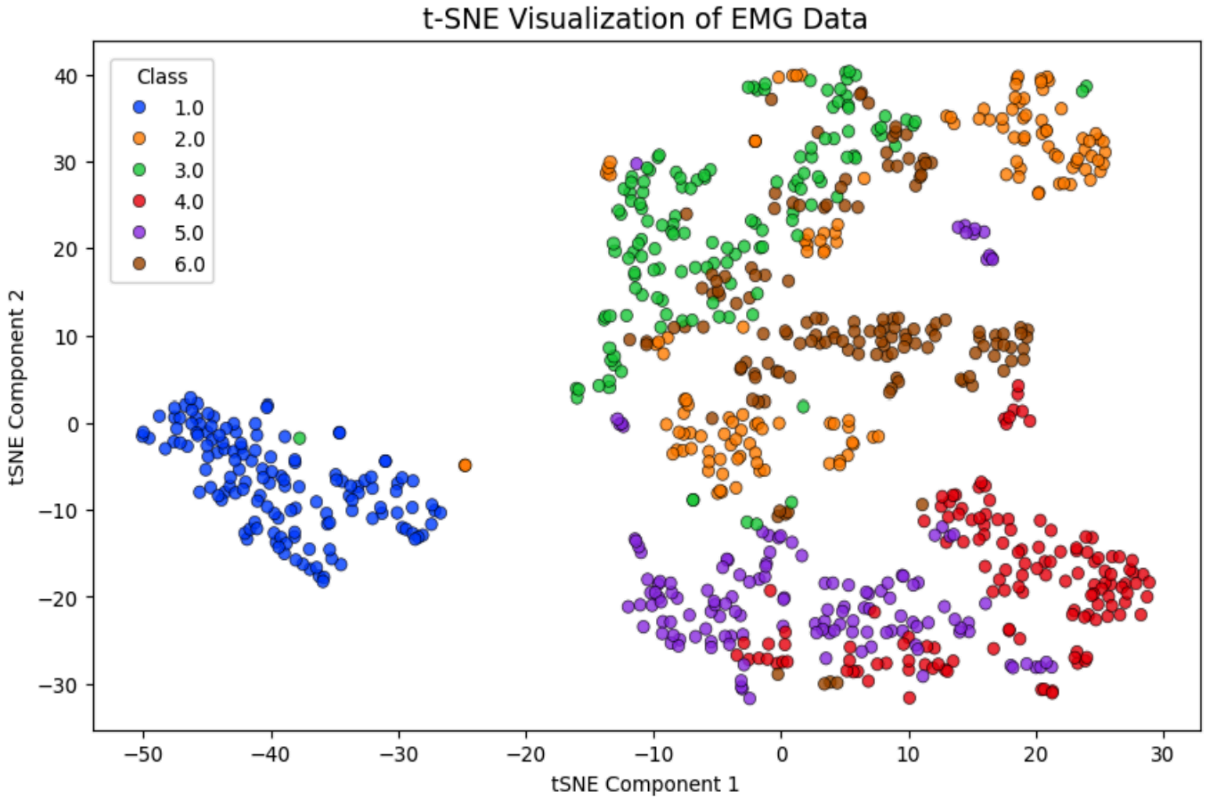

Fig. 1

That being said, the t-SNE plots helped highlight unexpected patterns. For example, while previous predictions suggested that the "fist clench" gesture (class 2) might be the most difficult to classify, visualization showed it to be quite distinct. Conversely, wrist flexion (class 3) demonstrated significant overlap with radial and ulnar deviations, both visually and in the model's confusion matrix.

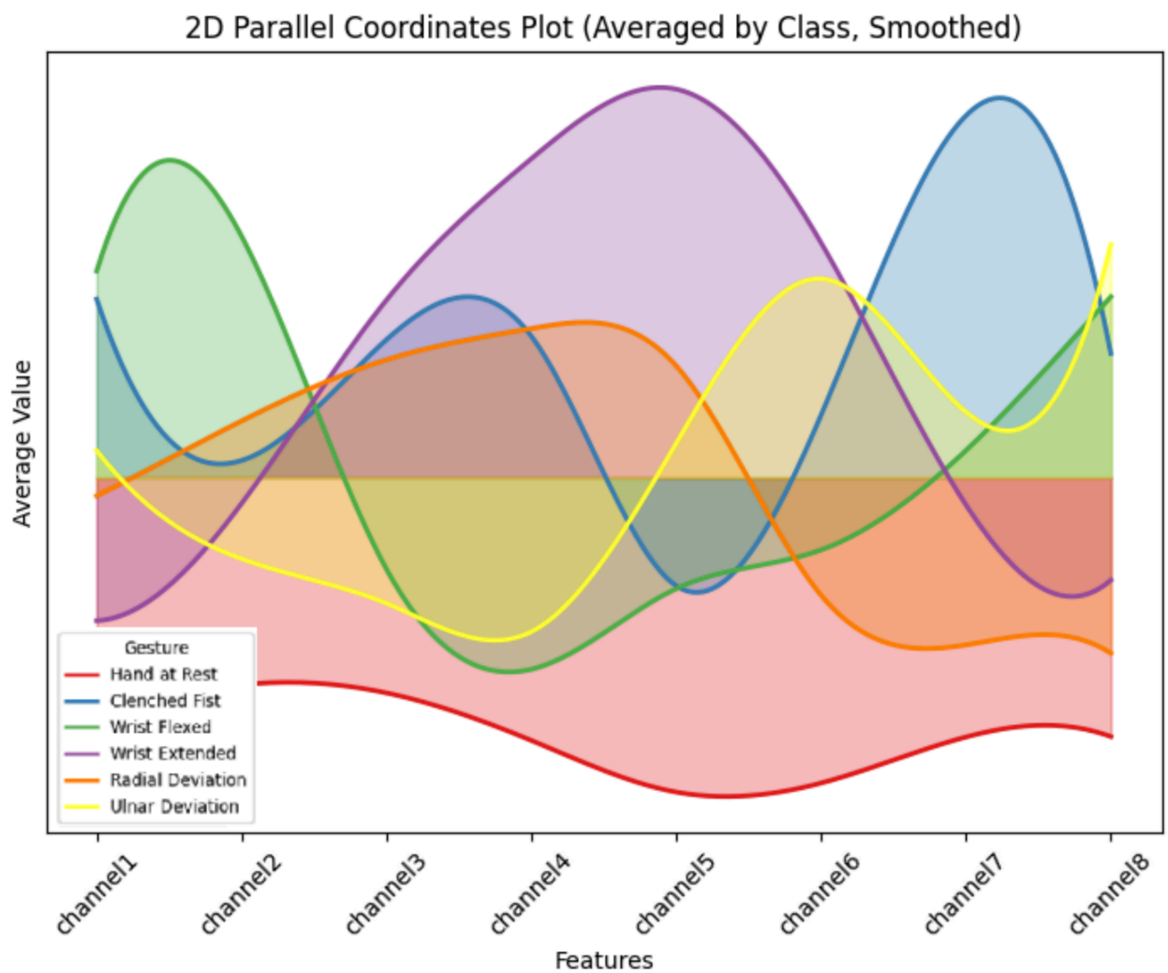

Fig. 2

Next, line plots and pairplots helped realize the EMG "signatures" of each of the various gestures. Initially, I realized that these are useful abstractions for understanding differences between classes. I previously wrote that these visualizations do not directly map to anatomical movement or muscular physiology; they merely serve as a practical aid in grasping how the ML model might be perceiving the data space. However, upon further reflection, I now claim that they do in fact exhibit meaningful real-world analogues. For instance, the EMG "signature" for wrist extension appears to be nearly the opposite—flipped across the y-axis—of the "signature" for wrist flexion. This makes intuitive sense given the data collection setup, where electrodes were arranged in a circular ring around the forearm, capturing opposing muscle activations in a spatially-consistent and spatially-aware manner.

Overall, the data visualization validated my assumption that statistical measures of sEMG wave signals preserve enough information to allow for successful classification without explicitly modeling time-dependent features (the temporal dynamics).

5. Results and Discussion

After preprocessing the dataset and selecting appropriate features (initially RMS), I began with training an SVM classifier using Scikit-learn. Using only RMS for each channel, the model achieved an overall accuracy of 91.07%, a result that not only met the original performance target outlined in the proposal but also matched the benchmark found in related literature. For instance, Kok et al. (2024) reported similar classification accuracy using RMS features and SVMs on a generalized sEMG dataset, which reinforced the credibility of these initial results.

A peer commented that it was interesting and perhaps suspicious that both of Kok et al. and this study independently reported achieving exactly 91% classification accuracy with our respective sEMG classifications. While the similarity in results initially did alert attention, it is in fact a coincidence. Although we were working on similar problems, we used different datasets, had differing preprocessing methods, and even settled on different features. Thus I can confidently assert that this shared accuracy value is incidental and not meaningful.

The performance metrics were visualized and tabulated in a classification report (Fig. 3) and a confusion matrix (Fig. 4), which offered further insight into the classifier's behavior. The confusion matrix revealed that while most gesture classes were classified accurately, gesture 3 (wrist flexion) exhibited the lowest precision and recall. This was consistent with earlier t-SNE visualizations, which showed that flexion overlapped significantly with neighboring classes such as radial and ulnar deviations. Conversely, gesture 2 (fist clench), which was initially predicted to be the most difficult to classify due to perceived lack of separability in t-SNE space, turned out to be among the best-performing classes in the final model. This result highlights the limitations of dimension-reduced visuals like t-SNE.

```
Accuracy: 0.9107142857142857

Classification Report:
              precision    recall  f1-score   support

         1.0       0.97      1.00      0.98        28
         2.0       0.93      0.93      0.93        28
         3.0       0.85      0.82      0.84        28
         4.0       1.00      0.89      0.94        28
         5.0       0.87      0.93      0.90        28
         6.0       0.86      0.89      0.88        28

    accuracy                           0.91       168
   macro avg       0.91      0.91      0.91       168
weighted avg       0.91      0.91      0.91       168
```

Fig. 3

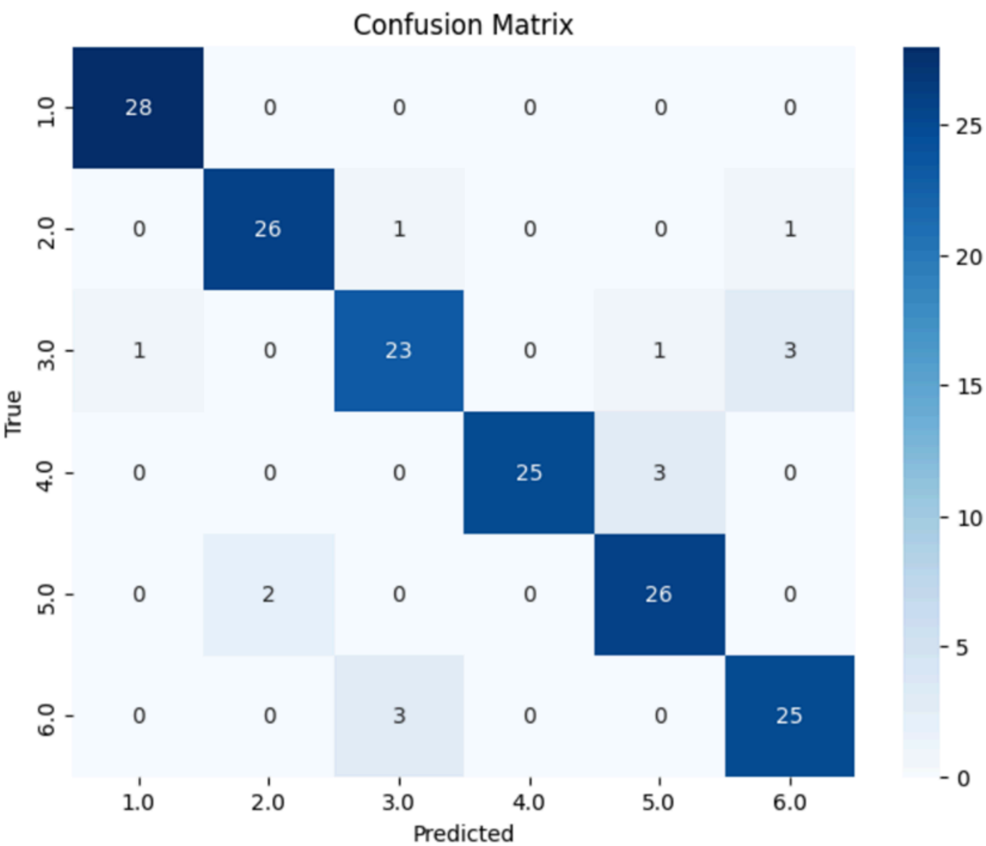

Fig. 4

While the SVM yielded unexpectedly strong initial results (most likely owing to the robust reprocessing the experimenter performed), the classifier's performance appeared to hit a ceiling—particularly when dealing with gestures that naturally shared similar muscle activation patterns.

Before progressing onto a more complex model, I wanted to test the model on different features since they had already been extracted anyway. My next step was to try utilizing all 24 features extracted (RMS, MAV, and SSC for each channel). This yielded a lower classification accuracy of 88.09%. Then I Attempted just using SSC. This yielded the worst results, with an accuracy of just 32.7%. This indicated that SSC is what was affecting the classification ability of the model. So then I tried RMS and MAV in conjunction (for a total of 16 features – 8 and 8). This yielded a new best classification accuracy of 92.3%. Out of curiosity, I tested the model once more just using MAV. Surprisingly, this outperformed even RMS combined with MAV, for a classification accuracy of 92.9%. This result contradicted common findings in the literature, which typically favor RMS. It suggests that MAV may be even better suited to this type of problem under certain conditions.

This variation in SVM classification accuracy based on feature selection is detailed in the table below.

Features Selected	Accuracy Achieved with SVM	Relative Rank
RMS	91.07%	3
SSC	32.71%	5
MAV	92.86%	1
RMS + MAV	92.26%	2
RMS + MAV + SSC	88.09%	4

Table 1

While these results were exciting, curiosity about more complex model types spurred a transition. The main alternative classification method discussed in the available sEMG classification literature was Neural Networks. They are particularly well-suited for tasks involving non-linear relationships. Plus, they are less constrained by complexities with feature distribution or linear separability. Therefore, the next phase of experimentation focused on evaluating whether a Neural Net could outperform the limited SVM.

The neural net at hand is a fully connected neural network using TensorFlow's Keras API. In terms of its architecture, it begins with an input layer that accepts feature vectors of length equal to the number of columns in the training data. The first hidden layer contains 128 neurons and uses the ReLU activation function. This is followed by a dropout layer with a dropout rate of 30%, which helps prevent overfitting by randomly deactivating a portion of the neurons during training. The model continues with two additional hidden layers, each with 64 neurons and using the ReLU activation function. Each of these layers is also followed by a dropout layer with a 30% dropout rate, which help the model generalize. The output layer contains as many neurons as there are classes in the target variable (six), and it uses the softmax activation function, which is appropriate for multi-class classification tasks.

Prior to training, the target variable is first encoded into integer labels using a label encoder, and then transformed into one-hot encoded vectors. This is necessary because the categorical crossentropy loss function requires the target labels to be in one-hot encoded format. The model was compiled with the Adam optimizer. Training was conducted for 1000 epochs (this value was iteratively increased to ensure I was not missing out on any increased accuracy) using a batch size of 32, an industry standard. 10% of the training data is set aside as a validation set to monitor the model's performance during training.

Both training and validation accuracy and loss were tracked and plotted. These curves, shown in Fig. 5 show rapid convergence. By around the 100th epoch, validation accuracy already surpassed 90% and its performance continued to improve slowly before finally plateauing at about 94%. Notably, there was minimal divergence between training and validation accuracy curves, suggesting that overfitting was not an issue, and the model could generalize.

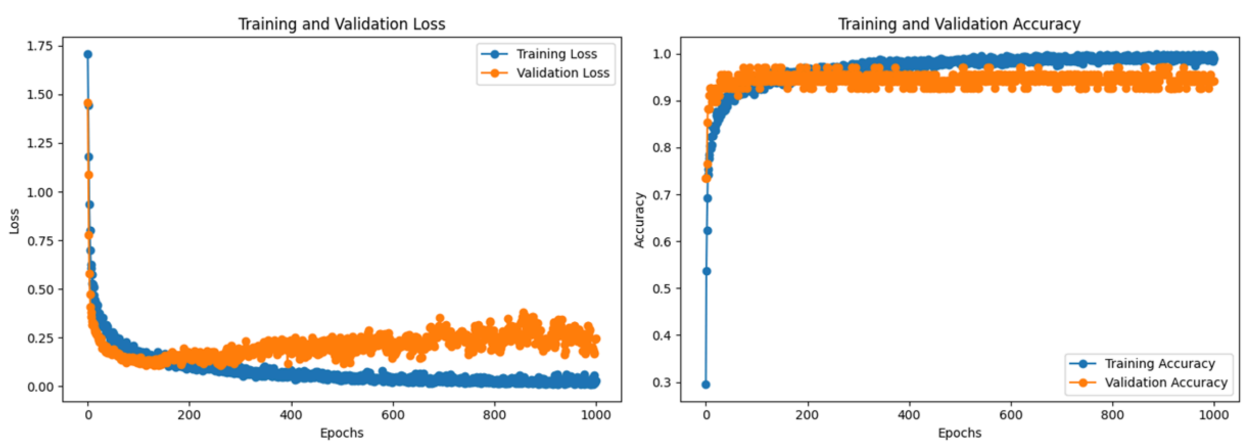

Fig. 5

The final classification accuracy of 94% represents a significant improvement over earlier models. Compared to the baseline SVM, which reached 91%, and even the optimized MAV-only SVM which achieved almost 93%, the neural network clearly demonstrated superior discriminative capability. This gain in accuracy was especially important for gesture classes that were previously difficult to separate, such as wrist flexion (class 3) and ulnar or radial deviations (classes 5 and 6). These gestures tend to have overlapping EMG activation profiles, but the neural net was able to tease apart the very subtle differences in signal patterns and succeed where the SVM could not.

Classification Report:

	precision	recall	f1-score	support
1.0	0.97	1.00	0.98	28
2.0	1.00	0.96	0.98	28
3.0	0.86	0.86	0.86	28
4.0	0.96	0.96	0.96	28
5.0	0.96	0.96	0.96	28
6.0	0.89	0.89	0.89	28
accuracy			0.94	168
macro avg	0.94	0.94	0.94	168
weighted avg	0.94	0.94	0.94	168

Fig. 6

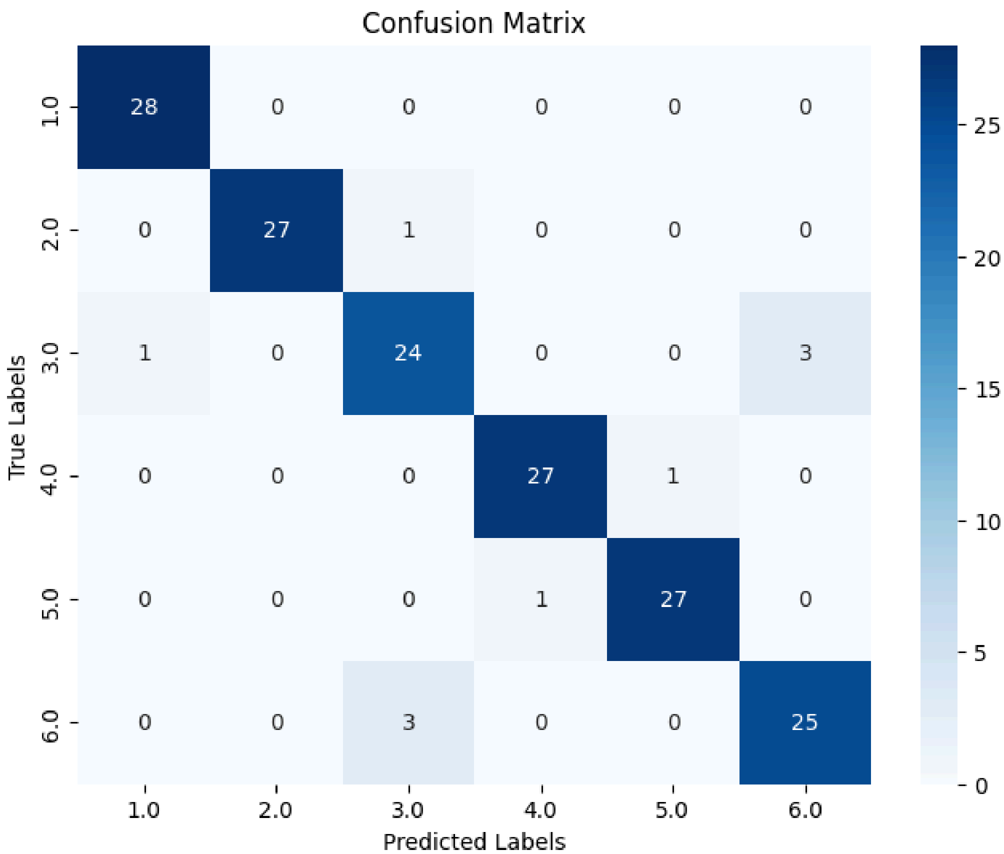

Fig. 7

6. Conclusion

This project set out to develop a machine learning solution for interpreting surface electromyographic (sEMG) signals to classify hand gestures for smart prosthetic control. Our goal was to create a classifier capable of being implemented into robotic prosthetic limbs and interpreting user intent.

The project was highly successful, exceeding my initial expectations. By extracting meaningful statistical features from the data and applying supervised learning techniques, we achieved 94% classification accuracy. This result surpassed both my own initial findings and multiple published benchmarks in the literature.

A significant finding was the effectiveness of relatively simple statistical features, particularly Mean Absolute Value (MAV), which outperformed more complex feature combinations when paired with appropriate machine learning models. Furthemore, the neural network implementation was superior compared to traditional SVM, especially for gestures with similar muscle activation patterns.

While there remains questions about new-user generalizability, this project establishes a solid foundation for future work on the topic. In conclusion, highly-accessible machine learning engineering has the potential to create more responsive, intuitively-behaving prosthetics that improve quality of life and alleviate human suffering. My work on this will continue for the foreseeable future.

7. Data and Code:

Data Source: https://archive.ics.uci.edu/dataset/481/emg+data+for+gestures

Link to Github repository for this project (public):
https://github.com/khourypaul/MANE-4962-FINAL-PROJECT

8. References

1. Kok, C. L., Ho, C. K., Tan, F. K., & Koh, Y. Y. (2024). *Machine learning-based feature extraction and classification of EMG signals for intuitive prosthetic control*. **Applied Sciences**, *14*(13), 5784. https://doi.org/10.3390/app14135784

2. Shakya, S., & Ranjitkar, B. (2022). *Forearm bio-medical signal processing*. **International Journal of Emerging Trends in Engineering and Technology**, *2*(1), 24–37. https://doi.org/10.3126/injet.v2i1.7251849

3. Mhiriz, Z., Bourhaleb, M., & Rahmoune, M. (2024). *Leveraging machine learning for signal processing in surface electromyography (sEMG) for prosthetic control*. In **Advances in Artificial Intelligence and Data Engineering** (pp. 125–137). Springer. https://doi.org/10.1007/978-3-031-68650-4_11

4. Di Nardo, F., Nocera, A., Cucchiarelli, A., Fioretti, S., & Morbidoni, C. (2022). *Machine learning for detection of muscular activity from surface EMG signals*. **Sensors**, *22*(9), 3393. https://doi.org/10.3390/s22093393

5. Ojha, A. (2023). *An introduction to electromyography signal processing and machine learning for pattern recognition: A brief overview*. **Extensive Reviews**, *3*(1), 24–37.

https://doi.org/10.21467/exr.3.1.8382

6. Simar, C., Colot, M., Cebolla, A.-M., Petieau, M., Cheron, G., & Bontempi, G. (2024). *Machine learning for hand pose classification from phasic and tonic EMG signals during bimanual activities in virtual reality*. **Frontiers in Neuroscience**, *18*, Article 1329411. https://doi.org/10.3389/fnins.2024.1329411

9. Appendix

Authorial Disclaimer: *I used ChatGPT\Claude\Other AI tool to rephrase some sentences for clarity. The executive summary was also generated in part with the help of generative AI in order to create an effective summary for the target audience. Furthermore, Gemini was utilized for some of the code generation, particularly scripts used in the preprocessing stage. The author remains committed to transparency regarding the use of AI in an academic setting.*