PulseSort

EMG-Based Hand/Finger Gesture Recognition using Machine Learning

---

1. Introduction

Electromyography (EMG) signals are electrical signals generated by muscle activity. These signals can be used to identify hand and finger gestures, which has applications in human–computer interaction, prosthetics, and rehabilitation systems.

The objective of this project, **PulseSort**, is to design a machine learning–based system capable of recognizing **12 different hand/finger gestures (including rest)** using EMG data. Multiple classifiers are trained and compared to identify the most suitable model for gesture recognition.

---

2. Dataset Description

The dataset used in this project consists of EMG recordings stored in CSV files. Each file corresponds to a specific gesture class.

Dataset characteristics: - File format: CSV - Number of gesture classes: 12 - Gestures include: clenched, fist, four, index_finger, okay, peace, rest, rock, spread, three, thumb, up - Each row represents EMG signal samples with associated labels

All CSV files are stored inside the `data/EMG/` directory and are loaded programmatically.

---

3. Project Structure

The project follows a modular and organized structure:

```
PulseSort/
|── data/
|   └── EMG/
|       └── *.csv
|
|── src/
|   ├── load_data.py
|   ├── prepare_dataset.py
|   ├── feature_extraction.py
|   ├── train_models.py
```

```
│    ├── evaluate.py
│
├── main.py
├── requirements.txt
├── README.md
```

---

4. Methodology

4.1 Data Loading

All EMG CSV files are read and concatenated into a single Pandas DataFrame using the `load_emg_data()` function.

4.2 Data Preparation

- Gesture labels are encoded into numerical form
- Features (X) and labels (y) are separated
- Data is split into training and testing sets (80% / 20%)

4.3 Feature Scaling

Standardization is applied using **StandardScaler** to normalize EMG values. Scaling is performed **after train-test split** to avoid data leakage.

4.4 Machine Learning Models

Three classifiers are implemented: - Support Vector Machine (SVM) - K-Nearest Neighbors (KNN) - Logistic Regression

Each model is trained using the same training data for fair comparison.

---

5. Model Evaluation

Model performance is evaluated using **classification accuracy** on the test dataset.

Results obtained: - SVM Accuracy: ~0.08 - KNN Accuracy: ~0.08 - Logistic Regression Accuracy: ~0.08

The low accuracy indicates that raw EMG values alone are insufficient for accurate gesture recognition.

---

6. Discussion

The identical accuracy across models suggests that: - The dataset is likely balanced across 12 classes (random guess ≈ 1/12 ≈ 0.083) - No advanced time-domain EMG features (RMS, MAV, WL, etc.) were extracted

This confirms the importance of **feature extraction** in EMG-based classification tasks.

---

7. Conclusion

This project successfully demonstrates the complete pipeline for EMG-based gesture recognition, including: - Dataset handling - Preprocessing and scaling - Training multiple machine learning classifiers - Comparative evaluation

Although the current accuracy is low, the system provides a strong foundation for future improvements.

---

8. Future Work

To improve performance, the following enhancements are recommended: - Time-domain feature extraction (RMS, MAV, Zero Crossing) - Frequency-domain analysis - Deep learning approaches (CNN, LSTM) - Real-time EMG signal acquisition

---

9. Technologies Used

- Python
- NumPy
- Pandas
- Scikit-learn
- VS Code

---

10. Acknowledgement

This project was developed as part of an academic learning initiative to understand EMG signal processing and machine learning–based classification.

End of Report