TECHNISCHE MECHANIK
an open access journal

journal homepage: www.ovgu.de/techmech

Tech. Mech., Vol. 44, Is. 1, (2024), 47–60

Received: 15.01.2024
Accepted: 03.05.2024
Available online: 03.05.2024

Convolutional Neural Networks for Time Series Data Processing Applicable to sEMG Controlled Hand Prosthesis

Golam Gause Jaman*, and Marco P. Schoen

Measurement and Control Engineering Research Center, Idaho State University, Pocatello, ID, 83209, USA

Abstract: Surface electromyography (sEMG) signals are often used to control prosthetics, but accurately interpreting these stochastic signals remains challenging. Deep learning tools like convolutional neural networks (CNNs) have shown promise for complex classification problems, yet CNN applications for time series data are limited. This work explores adapting CNNs to sEMG time series for improved classification, addressing two questions: 1) Can a CNN trained on cross-subject data generalize without individualization? 2) Can a small individualized dataset sufficiently train an accurate control model? To investigate, sEMG data is formatted into images using handcrafted features, with pixels representing multichannel time series. A ResNet50 architecture is trained on two datasets: individual and cross-subject. Results show cross-subject models fail to provide accurate subject-specific control due to high inter-subject variability of sEMG. However, ResNet50 trained on individual data produces highly accurate offline and near real-time classification. The proposed method is also tested on an external dataset and compared to similar published methods, demonstrating strong performance. In summary, CNNs show promise for prosthetic control from sEMG but require individualized training data. The proposed data formatting and ResNet50 architecture can enable precise control from minimal data, overcoming barriers to clinical implementation. Further research into cross-subject generalizability is warranted to understand the sources of variability and improve model robustness.

Keywords: Convolutional Neural Network, Feature Engineering, Residual Neural Network, Spatial Feature, Surface Electromyography, Time Series Data

1 Introduction

Surface electromyography (sEMG) signals are used to control assistive devices like prostheses, wheelchairs Li et al. (2017); Geng et al. (2018); Al-Angari et al. (2016), speech interfaces Scheme et al. (2007, 2006), and virtual reality systems Summa et al. (2019). These signals originate from skeletal muscles and reflect motor neuron activity. However, accurately sensing these small voltage potentials (-5 to +5 mV) is challenging due to cross-talk from other muscles Côté Allard et al. (2019). Much research has focused on extracting motion intent from sEMG for prosthesis control through methods like artificial neural networks (ANNs) Cheron et al. (1996), fuzzy logic Chan et al. (2000), and optimization algorithms Fernandez et al. (1996). Typically, ANNs are trained offline then utilized for real-time control Moron et al. (2018). An advanced ANN enabled intuitive cursor control using eight sEMG sensors Olsson et al. (2020a), indicating promise for deep learning in prosthetic devices by addressing issues like latency Olsson et al. (2020a); Atzori and Müller (2019).

Non-invasive sEMG systems are preferred for prostheses Atzori et al. (2016). However, sEMG signals are stochastic, complicating motion prediction Xiong et al. (2021). Often inertial sensors complement sEMG features to reduce dependency on additional sensors Xiong et al. (2021), but this increases costs. Standardized sEMG datasets enable comparison of offline performance, but replicating real-time hardware is difficult, limiting benchmarking Côté Allard et al. (2019). Acquiring local data can optimize feature engineering and classification research.

Recently, time series sEMG data has been formatted as images then classified using techniques like convolutional neural networks (CNNs) Xiong et al. (2021). This branched into either short-term motion predictions or longer full motion forecasts Xiong et al. (2021). Handcrafted features utilize domain knowledge to construct informative, low-dimensionality images to enable explainable predictions with reduced computation Xiong et al. (2021).

Time series classification (TSC) is a challenging data mining problem Esling and Agon (2012). A common approach is nearest neighbor classifiers with distance functions like dynamic time warping Begnall et al. (2017). Feature engineering can improve TSC outcomes Begnall et al. (2017). The current state-of-the-art is the computationally intensive HIVE-COTE algorithm utilizing an ensemble of 35 classifiers Lucas et al. (2018).

Deep learning shows promise for tackling complex classification. Convolutional neural networks (CNNs) revolutionized computer vision Krizhevsky et al. (2012) and are now being applied to natural language processing Bahdanau et al. (2015). For TSC, CNNs could enable improved feature learning Neamtu et al. (2018), with work exploring CNN solutions using formatted

* E-mail address: jamagola@isu.edu

doi: 10.24352/UB.OVGU-2024-053

sEMG data Tsinganos et al. (2020). An automated CNN topology selection method has also been proposed but requires significant computing resources Olsson et al. (2020b). We propose a simpler data formatting approach to leverage existing image classification techniques.

Recurrent neural networks (RNNs) are well-suited for time series data modeling given their memory capacities Zeroual et al. (2020). Long short-term memory (LSTM) networks are a type of RNN that can store and forget information selectively. Deep residual networks (ResNets) are another effective deep learning architecture, demonstrating advantages for image recognition Hinton et al. (2012); LeCun et al. (2015) and language processing Bordes et al. (2014); Jean et al. (2014). Benefits of ResNets include enabling deeper network training, improving performance, reducing overfitting, and easier implementation He et al. (2016). Their residual connections contribute to widespread ResNet adoption.

In this work, ResNet, as depicted in Fig. 1, a variant of CNN, is adapted to accommodate time series data rather than using traditional choices such as RNN or LSTM. The motivation for the adaptation of this image classification tool for use in time series data is given by the prospect of crafting a real-time solution and the potential to incorporate different sets of classification methods. As the CNN methods gained popularity in the field of computer vision, GPUs such as NVIDIA-based GPUs are optimized for tasks defined in CNN where LSTM draws less benefit. Using specialized hardware results in shorter training and testing durations for ResNet versus LSTM for similar problem complexity Weytjens and De Weerdt (2020). In addition, LSTM networks demand longer signal durations for accurate motion classification Toro-Ossaba et al. (2022).

The proposed method accumulates temporal and spatial sEMG signal components into images using 100ms windows. The handcrafted features enable explainable predictions with reduced computation. These changes make deep learning suitable for real-time applications. Optimal parameters for image creation are found through trial-and-error.

Comparing multi-class motion classification performance is difficult due to differences in motion classes, hardware, objectives, signal capture techniques, etc. This work uses a common external database with 40 classes and equal class examples to compare the proposed method against literature techniques. The 2.5% random prediction accuracy for this database is far lower than the 25-33% seen in other investigations, enabling rigorous benchmarking. The external database investigation is conducted using the cross-subject motion generalization and individual subject motion classification with the data generated from ten participants with a healthy upper limb as detailed in section 6.3.

2 The Hardware Configuration

The MyoMex armband from Thalmic Labs is used for data collection. It contains eight medical-grade stainless steel single differential sEMG electrodes and a 9-axis inertial measurement unit (IMU) sampling at 200Hz with 8-bit resolution over Bluetooth. Only the sEMG data is used to align with research objectives. The armband is synchronized for each participant using the Myo Connect application. MATLAB® initiated and managed sEMG streaming. Identical MyoMex hardware and configurations sourced internal and external datasets used in this study. Resources for interfacing the armband with MATLAB® are available from the referenced GitHub repository (https://github.com/mark-toma/MyoMex).

3 The Experiments

Four specific hand motions are considered for dataset generation: fist, cylindrical grasp, spherical grasp, and tip pinch. Data is recorded without restrictions on shoulder or elbow positions (dynamic) but emphasizing the predefined motions De la Cruz-Sanchez et al. (2019). Motions are performed using the MyoMex armband sensor shown in Fig. 2. Multi-subject data collected under University of the Mixteca approval has been used previously De la Cruz-Sanchez et al. (2019). Additional single subject experiments (section 5.2) had participants repeatedly perform the gestures, constructing training/testing datasets while taking breaks to prevent fatigue influence. Raw 5-second sEMG data is stored for each gesture. A second experiment quantified near real-time response and system latency stemming from data acquisition and classification.

4 The Data and the Pre-Process

Two separate data sets are used - one collected in this study, and one from prior work De la Cruz-Sanchez et al. (2019). The time series sEMG data (”raw data”) undergoes feature extraction and formatting to conform to image classification criteria. Recordings are trimmed to three seconds by dropping the first and last seconds. The trimmed data is made absolute to remove polarity. It is then sectioned into sliding windows of twenty samples with a five sample stride. Window and stride sizes are set through trial and error; optimization is left for future work. The windowed samples are averaged to smooth the data by removing rapid alternations. Eq. (1) shows the smoothed sEMG data where w and s are the window size and stride size respectively. $x_{n,m}$ is the sampled sEMG at the n^{th} instance for m^{th} sensor producing $g_{j,m}$ is the j^{th} smoothed sample. The moving averaged sEMG data is further processed to extract the two key features (details are found in sections 4.1 and 4.2) used in the construction of the feature image for the hand motion classification investigation. All the pre-processing routines to format the raw data to the feature image with directions are available at (https://github.com/jamagola/sEMG).

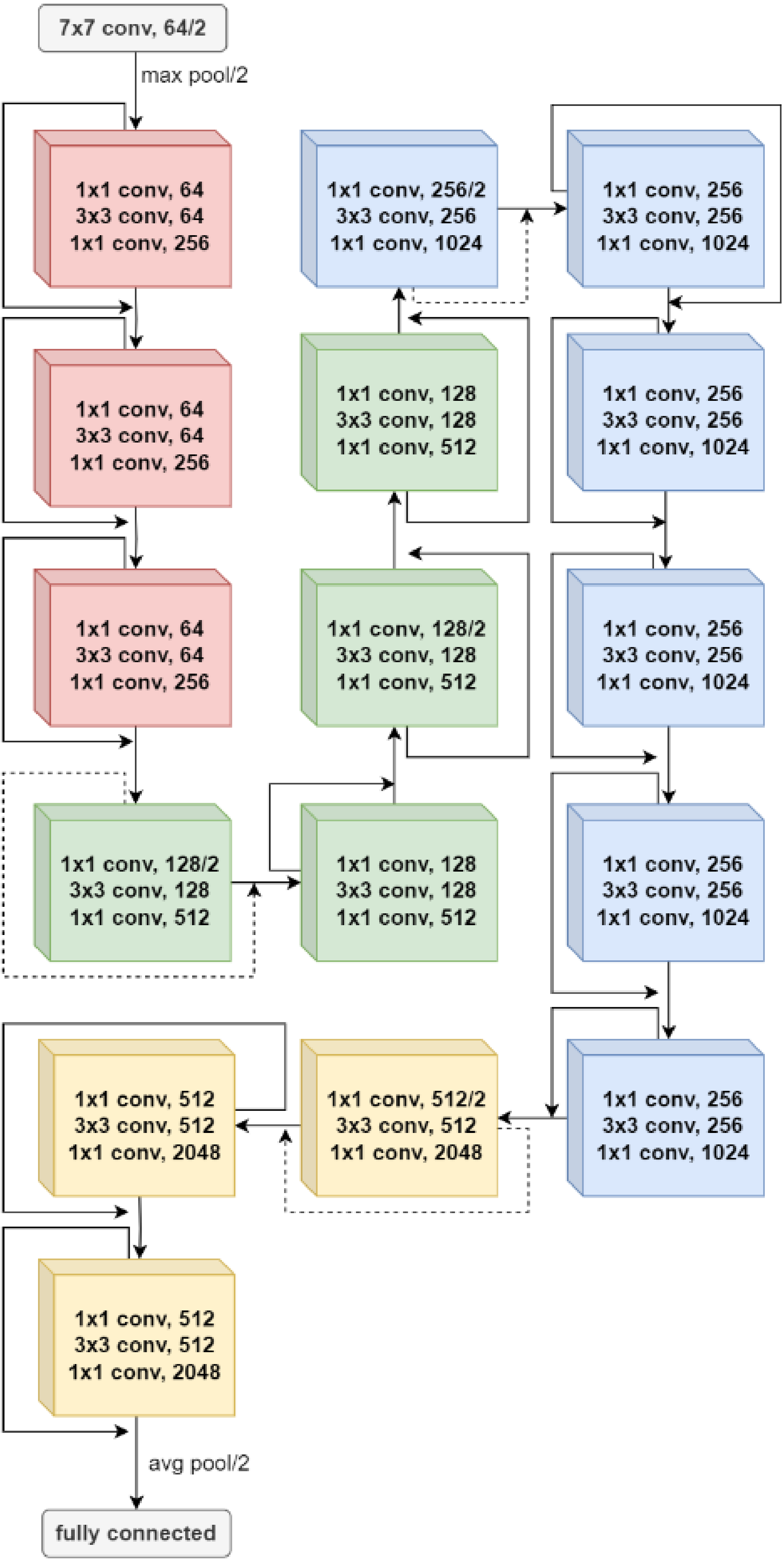


Fig. 1: CNN architecture is utilized for a given sEMG classification problem (ResNet50) He et al. (2016).

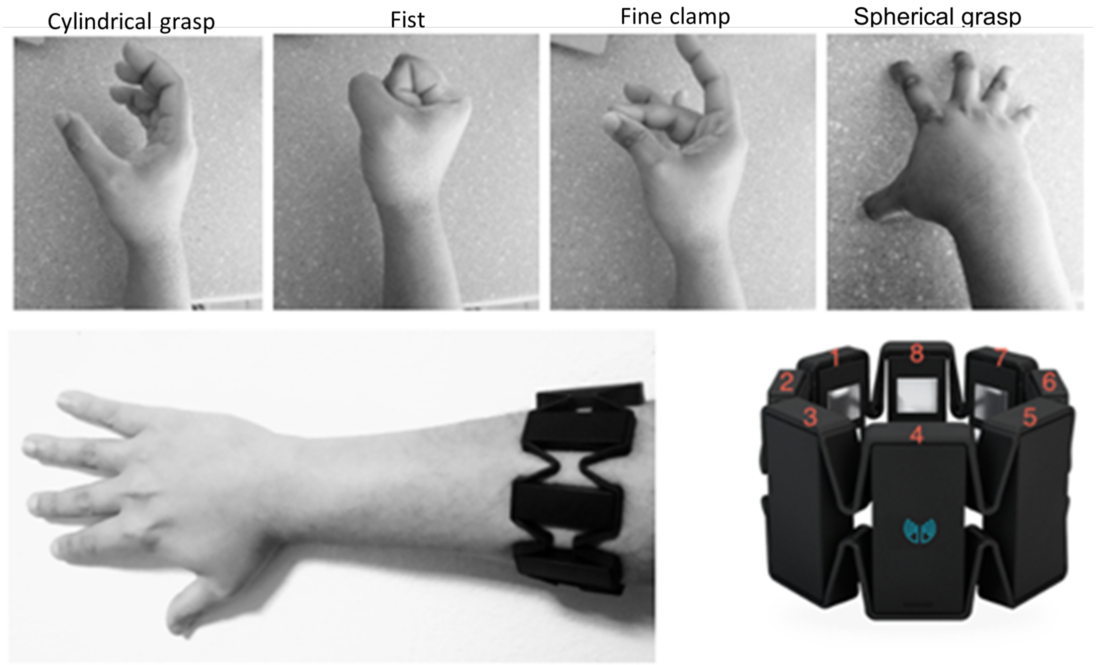


Fig. 2: Set of four different hand gestures and MyoMex armband.

$$g_{j,m} = \sum_{n=j\times s}^{w+j\times s-1} \frac{|x_{n,m}|}{w} \tag{1}$$

4.1 Simple moving average (MA)

The smooth sEMG data is sequentially sectioned and averaged using a second sliding window composed of twenty data points with a stride size of five data points. The resultant smoothed data leads to a dimensional reduction in the number of samples from the raw data. The reduced-dimension data from all eight sEMG sensors is accumulated and formatted to a 2D array of numerical values with each row presenting temporal states and each column indicating individual source/sensor (spatial locations). The resulting matrix is addressed as a feature matrix for the moving average (MA) in this text. Eq. (3) shows the MA feature matrix composed of elements from p steps defined in Eq. (2), where k is the index for the second window.

$$f_{k,m} = \sum_{j=k\times s}^{w+k\times s-1} \frac{g_{j,m}}{w} \tag{2}$$

$$M_{MA} = \begin{bmatrix} f_{1,1} & \cdots & f_{1,8} \\ \vdots & \ddots & \vdots \\ f_{p,1} & \cdots & f_{p,8} \end{bmatrix} \tag{3}$$

A filled contour map composed of spatial and temporal information can be generated using the feature matrix suitable for applying machine learning techniques. The spatio-temporal feature contains amplitude or response from the space time coordinate. The response is then presented as a surface plot that includes the contour map for 2D presentation. The filled contour map figures are exported to images with desired resolution and treated as single data point corresponding to a single hand motion. The generic process for processing raw sample data into an image-based feature is depicted in Fig. 3 and Fig. 4.

Contour mapping further emphasizes spatiotemporal sEMG patterns. Filled contour maps generalize groups of values, reducing noise influence so final feature image resolution can be decreased without losing key patterns. This handcrafted image is based on observations of the raw sEMG data matrix during acquisition. A twenty level filled contour map with normalized color axis is applied to the feature matrix, essentially quantizing the sEMG features through zero-order hold interpolation of contour levels. Quantization is followed by color scale normalization. The map is then converted to a 300x300 pixel image for offline and near real-time classification.

4.2 Spatial Feature (SF)

Consider an irregular polygon with eight vertices. Eight straight lines are drawn from the center of the irregular octagon to the vertices. Let the length of the lines connecting the vertices to the center be defined by the magnitude of the smooth sEMG data

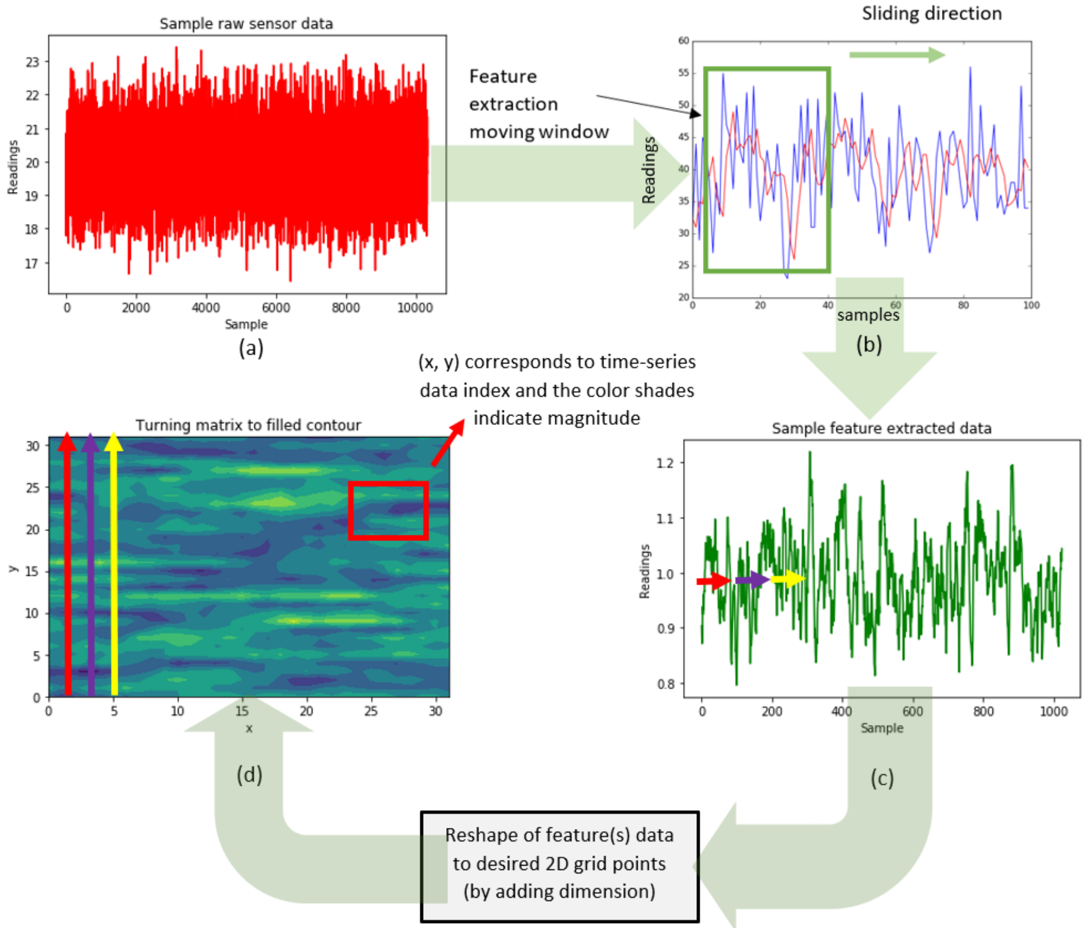

Sample raw sensor data
Readings
Sample
(a)
Feature
extraction
moving window
Sliding direction
Readings
samples
(b)
(x, y) corresponds to time-series
data index and the color shades
indicate magnitude
Turning matrix to filled contour
y
x
Sample feature extracted data
Readings
Sample
(c)
(d)
Reshape of feature(s) data
to desired 2D grid points
(by adding dimension)

Fig. 3: Process sequences for image feature extraction from single time series data (a) Single time series samples acquired (longer duration than data acquisition step for visualization purpose). (b) Feature extraction from the raw sensor readings is accommodated in the sliding window. (c) Data after the feature extraction process. (d) Re-arrange of 1D feature data to 2D grid points (since only one sensor data is considered) followed by feature image construction.

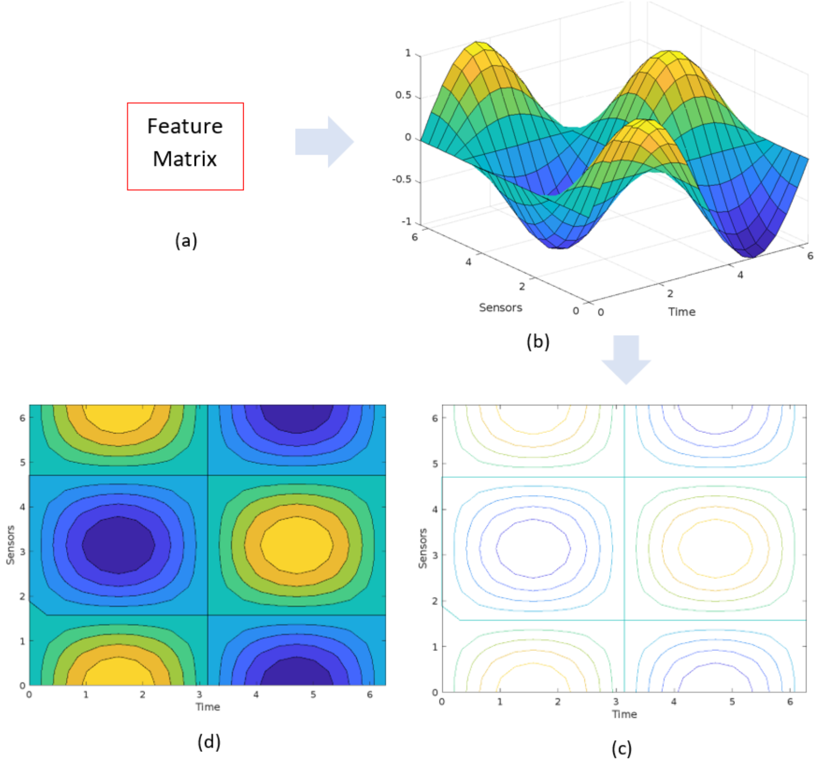


Fig. 4: Process steps for feature matrix to fill contour conversion. (a) Feature matrix derived from sEMG data acquisition. (b) Feature matrix plot as surface. (c) Contour plot generation from the surface plot. (d) Filled contour plot.

found in Eq. (1). Here, each of the eight lines corresponds to one of the eight sEMG probes. As a result, each of the interior angles of the octagon produced is a function of three sEMG sensors readings including the probes placed corresponding to each of the vertices and the two neighbors such that the interior angle $\theta_{j,m} = f(g_{j,m-1}, g_{j,m}, g_{j,m+1})$. The interior angles of the octagon are depicted in Fig. 5. A 2D array is constructed by placing the interior angle information along with the varying time instances in a similar manner as described in the MA feature section (4.1) except for the columns which correspond now to the vertices instead of the sensors. The array or feature matrix is defined in Eq. (5) addressed as Spatial Feature (SF) in the paper and elements in the feature matrix are defined in Eq. (4). Each of the matrix elements is the variance of the interior angle calculated from the $g_{j,m}$. The composed array is processed by the color mapping scheme and image construction steps described in the section 4.1. The process flow from data acquisition to image construction is shown in Fig. 6.

$$\phi_{k,m} = \sum_{j=k\times s}^{w+k\times s-1} \frac{(\theta_{j,m} - \bar{\theta}_{k,m})^2}{w-1} \tag{4}$$

$$M_{SF} = \begin{bmatrix} \phi_{1,1} & \cdots & \phi_{1,8} \\ \vdots & \ddots & \vdots \\ \phi_{p,1} & \cdots & \phi_{p,8} \end{bmatrix} \tag{5}$$

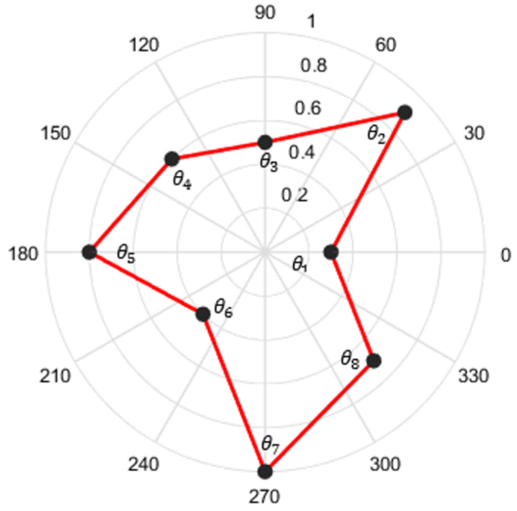


Fig. 5: Geometric visualization of spatial feature.

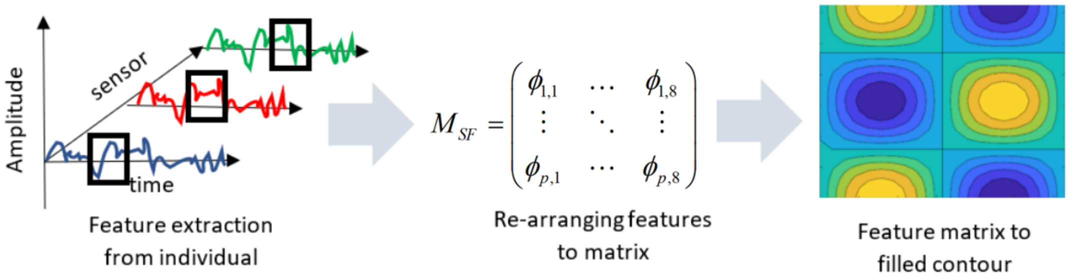

Amplitude
sensor
time
Feature extraction
from individual
$M_{SF} = \begin{pmatrix} \phi_{1,1} & \cdots & \phi_{1,8} \\ \vdots & \ddots & \vdots \\ \phi_{p,1} & \cdots & \phi_{p,8} \end{pmatrix}$
Re-arranging features
to matrix
Feature matrix to
filled contour

Fig. 6: The process sequence of extracted features to fill contour plot-based image features.

5 Image Classification

A key concept is converting time series data into images showing parallel temporal and spatial activity using 3-second measurements with 100ms windows, without inertial data. In a typical 2D convolutional neural network (CNN), labeled images are filtered by tunable 2D arrays called convolution operations defined by stride size and padding. Outputs are subsampled by pooling operations like mean pooling. Paired filtering and pooling layers with appropriate sizing compose a convolution layer. Layers are sequenced then flattened into a vector fed into a fully connected network. Training minimizes error via algorithms like stochastic gradient descent. Activations, batch normalization etc. can be inserted. Residual networks route copies of intermediate outputs to later

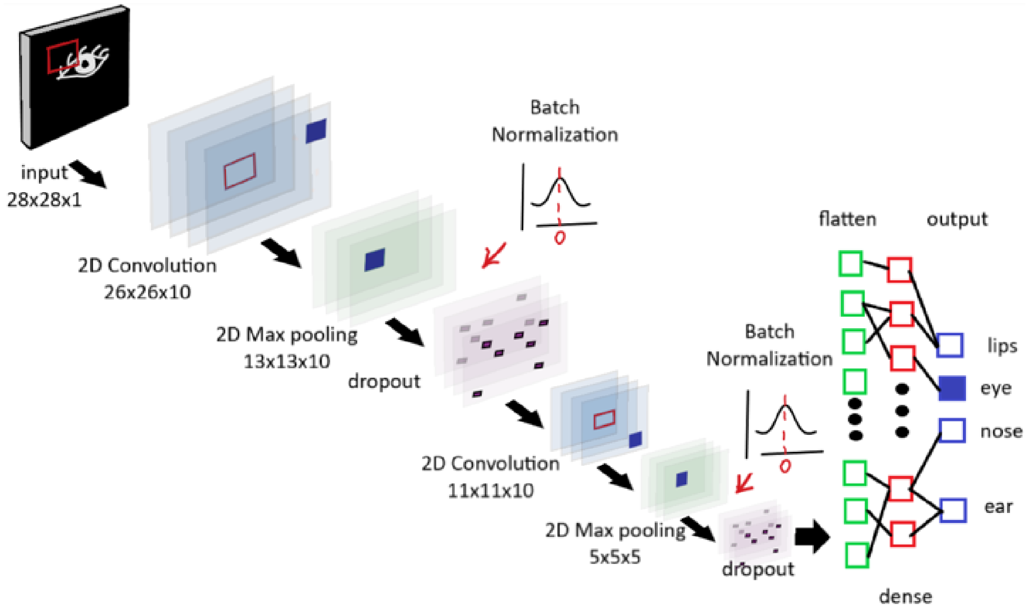


Fig. 7: Key processes involved in CNN-based classification.

layers, preserving signals. Fig. 7 depicts the key processes involved in CNN-based classification.

ResNet50 He et al. (2016) is used here (Fig. 1). No pre-trained transfer learning is used to avoid negative transfer Wang et al. (2019). Instead, MATLAB®'s toolbox trained ResNet50 from scratch with predefined architecture but untrained weights. The process flow (Fig. 8) shows the data acquisition to classification steps. Two routes stem from the described pre-processing features.

ResNet50 is trained on 300x300 pixel images embedding pre-defined motion information. The training epoch is 1000 with a batch size of 8. Total training iterations ranged from 800-9000 depending on dataset size. Additional hyperparameters like a 0.001 learning rate and 80/20 train/test split are manually set without optimization. Validation data is separated 5:1 from training with balanced classes. Training applied MATLAB®'s k-fold cross validation depending on data size. Two primary investigations are conducted: (1) cross-subject investigation (CSI) and (2) single-subject investigation (SSI).

5.1 Cross Subject Investigation or CSI

Two sEMG data sets are used. The first is from 12 subjects aged 25-55 of both genders, collected at the authors' lab with local approval. The second is sourced from a databank in De la Cruz-Sanchez et al. (2019) for cross-subject investigation (CSI). This contained 900 time series vectors from 225 participants performing four predefined motions once each, using identical hardware/configurations. Sourced data underwent smoothing per the pre-processing procedure, then MA and SF feature extraction and image construction separately. Data augmentation mitigated the lack of multiple trials per motion for individual participants. The extracted images are used to separately train and test ResNet50 networks based on the MA and SF features. The Cross Subject Investigation outcomes are discussed in detail in section 6.1.

5.2 Single Subject Investigation or SSI

Raw offline sEMG data is collected per section 3. ResNet50 models are trained on SF and MA image features separately for each of the 12 single subject investigation (SSI) participants, so 24 networks total without data augmentation. Each SSI subject performed each of the four motions 20 times, generating 80 images. 20% are randomly selected for testing, distributed equally across classes. So 64 training images per subject-network (16 per class) are used. A near real-time experiment then had subjects randomly perform the gestures sequentially. The classification pipeline included data recording, feature extraction, image formatting, and inference by the corresponding trained network to predict motion class. Process delays are measured from data acquisition start to prediction end.

6 Performance Analysis

The performance corresponding to every investigation considered in this paper is measured based on validation accuracy. The trained networks are configured using identical hyperparameters across all the investigations and tests that include, loss functions, activation functions, learning rate, optimization methods (stochastic gradient descent), etc.

6.1 Cross Subject Investigation

Cross-subject or multi-subject investigations (CSI) are tested using three different data configurations. Each of the configurations contains non-identical sizes of data points. Regardless of the configurations, the dimension of the feature images applied to every

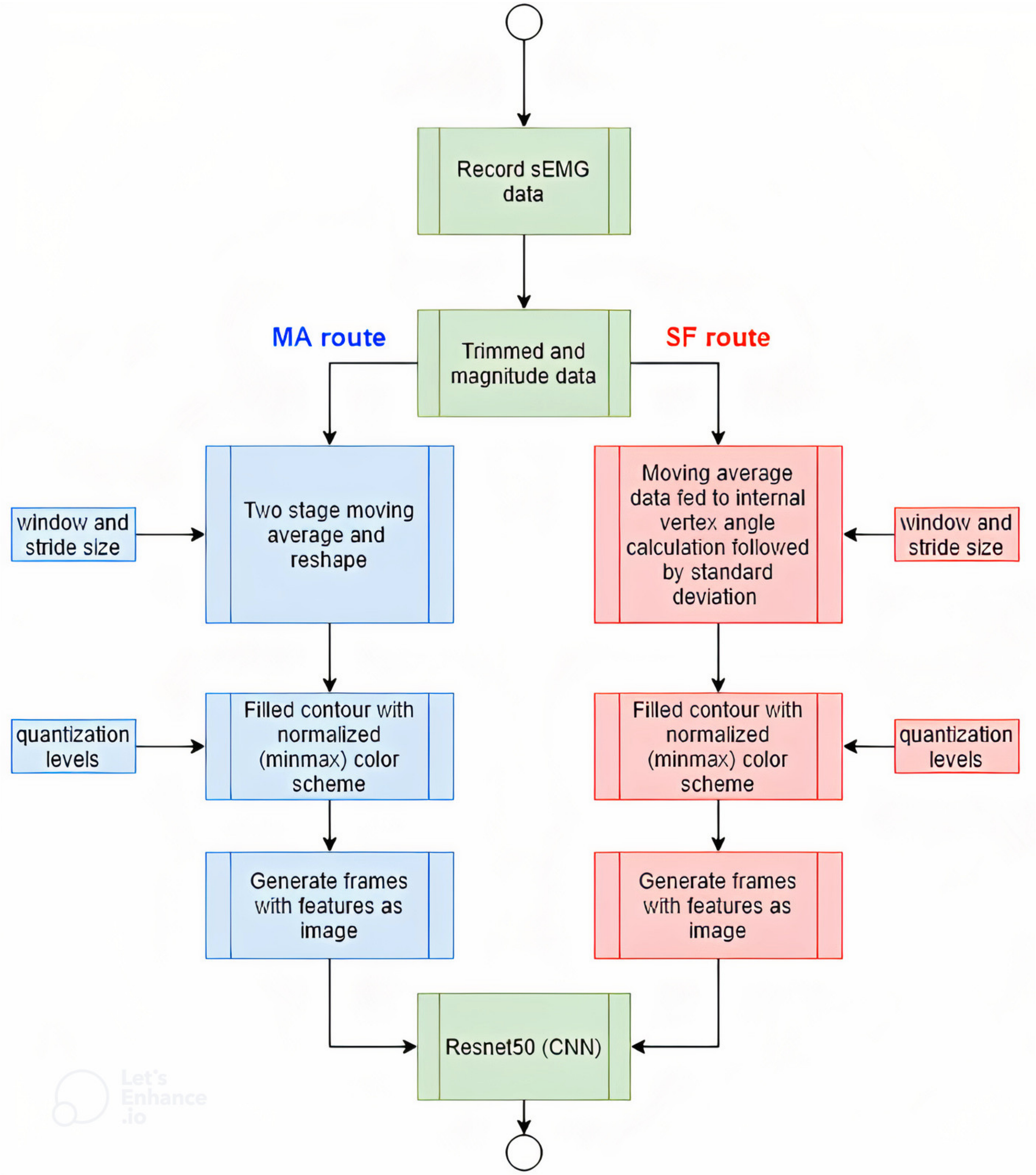

Record sEMG data
Trimmed and magnitude data
MA route
SF route
window and stride size
Two stage moving average and reshape
Moving average data fed to internal vertex angle calculation followed by standard deviation
window and stride size
quantization levels
Filled contour with normalized (minmax) color scheme
Filled contour with normalized (minmax) color scheme
quantization levels
Generate frames with features as image
Generate frames with features as image
Resnet50 (CNN)

Fig. 8: Proposed feature engineering and classification routes.

investigation in this paper remains unchanged. In this section, each CSI outcome is discussed separately in the following.

6.1.1 Configuration A

In this setup, image features are used to train the network offline without any additional data generations and formations. At the end of training with 1,000 epochs, the MA-based trained network settled with higher validation accuracy for offline classifications relative to SF. The MA-based model on average converged to 40% accuracy whereas the SF-based model to 36%. In Fig. 11a, the average validation accuracy curve is shown against the epoch for both features.

6.1.2 Configuration B

Image features are used to train models offline with data augmentation after train/test splitting. Training images are rotated 90°, 180°, and 270° clockwise to generate three additional features per data point. Rotations alter electrode sequence direction (Fig. 9). 180° rotations mimic muscle activity as if measured from different electrodes, capturing signal characteristics from diverse locations for cross-subject generalization. 90° and 270° rotations aim to integrate correlations between time and spatial domains under the same motion label. Investigations used both SF and MA features. MA converged to 35% validation accuracy after epochs versus 32% for SF (Fig. 11b).

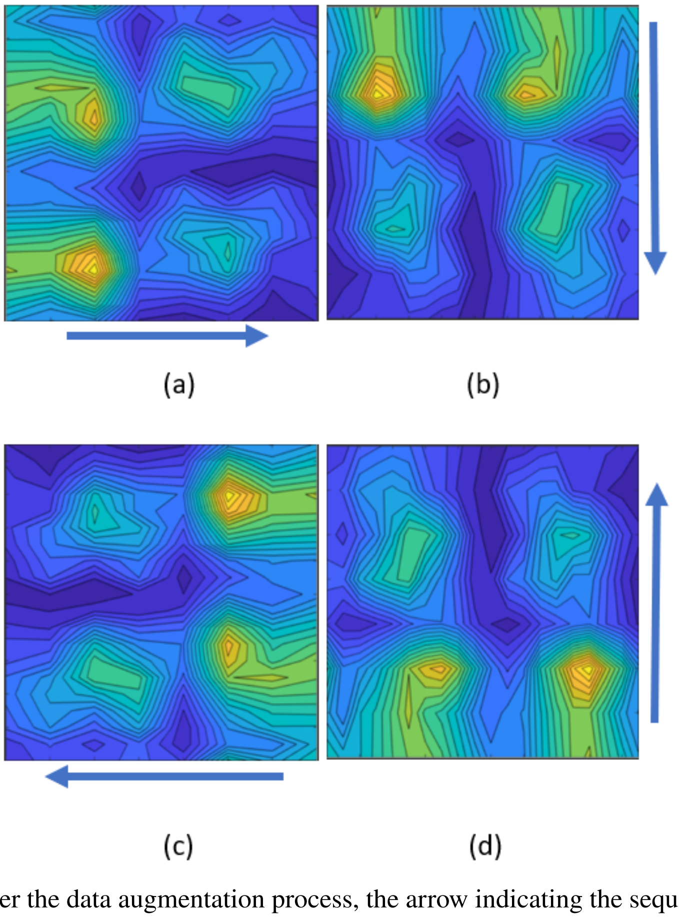


Fig. 9: Single motion images after the data augmentation process, the arrow indicating the sequence of sensor location. (a) No rotation, (b) 900 clockwise, (c) 1800 clockwise and (d) 2700 clockwise.

6.1.3 Configuration C

Cross-subject feature images contain motion and subject-specific signals, the latter acting as classification noise. An additional investigation aimed to mitigate participant characteristics and amplify generalizable motion attributes. The first motion class for each subject is used to embed subject information into other class features. Training/testing predicted only three motion classes, unlike other investigations. This concatenates one feature image to others from that subject, generating relative feature (RF) images before train/test splitting to relatively represent motions. Augmentation is also applied (Fig. 10). RF images used both MA and SF features. Fig. 11c and Fig. 11d shows validation accuracy of RF networks (a) without and (b) with data augmentation (DA). MA-based networks outperformed SF, achieving 52% and 50% final accuracy without and with DA respectively, versus 48% and 44% for SF.

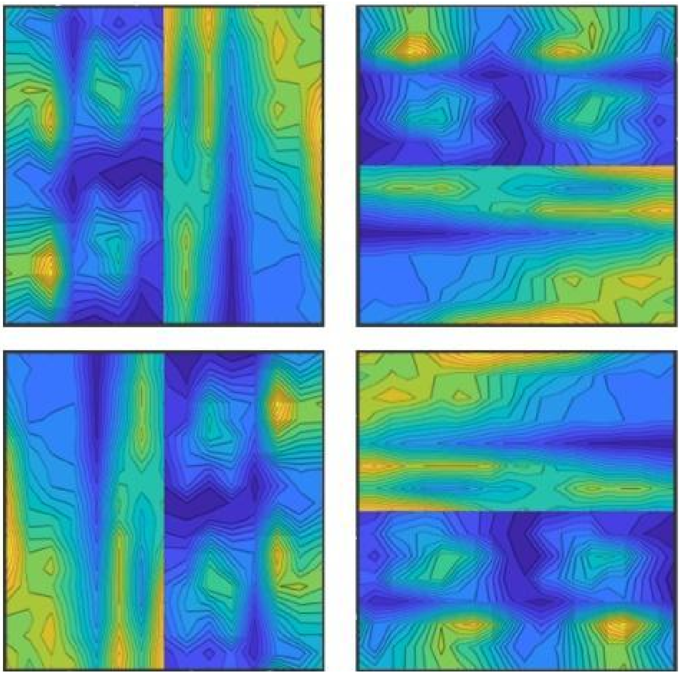


Fig. 10: Relative feature image of a single class of motion with data augmentation.

6.2 Single Subject Investigation (+Real-Time)

Trained MA and SF networks are tested on offline data, achieving 90% and 100% mean validation accuracy after maximum epochs respectively (Fig. 11e). Near real-time testing had subjects randomly perform ten motions with < 5s rest between. Total latency from acquisition to classification is 3.1s, dominated by the 3s data collection and feature extraction. Inference computation is 0.1s, hardware-dependent. MA and SF networks achieved 85% and 100% prediction accuracy in near real-time conditions.

6.3 External Database

The proposed MA feature method is applied to an external NinaPro DB5 database Pizzolato et al. (2017) using two adjacent Myo armbands with a 16 electrode, 22.5 degree configuration sampled at 200Hz Atzori et al. (2014). Ten subjects with 40 classes (excluding rest) and six trials per motion are used, totaling 2400 datasets with sEMG and IMU. MA features are chosen since IMU alters the spatial-temporal configuration for SF. The method is first applied to all classes collectively, achieving 80% cross-subject accuracy (Fig. 12a). Single subject testing then achieved 72.5% accuracy, surpassing the 69.04% best accuracy in Pizzolato et al. (2017) using mDWT and SVM.

The database is further reduced to ten motions from Sadhu (2019) and compared to 91% CNN accuracy in that work. Proposed CSI and SSI methods achieved 95% validation accuracy on this subset, with a higher 10% chance accuracy versus 2.5% for 40 classes. For full 40 class evaluation, subject models used two of six repetitions for testing (67/33 split) while ten subject models used 80/20 splitting. Results are summarized in Tab. 1. Fig. 12b compares validation accuracy of proposed and external methods.

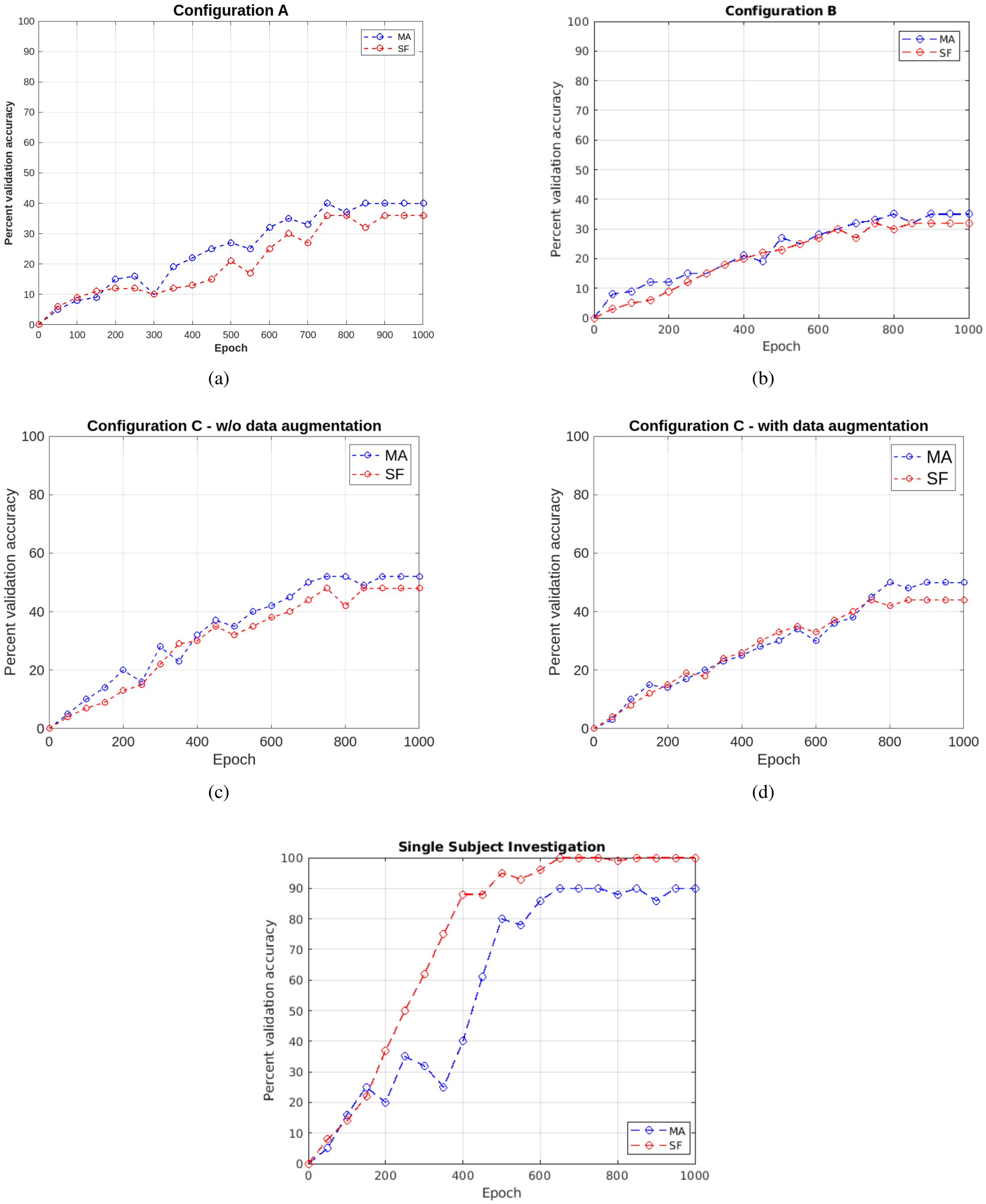
6.4 Results Summary

Studies often use ≤ 10 subjects with ≥ 6 motion trials Arozi et al. (2020); Javaid et al. (2021). Results show lower cross-subject prediction accuracy for the proposed methods and network versus ≥ 90% individual subject performance regardless of feature. SF outperformed MA, including in real-time. Further tests on an external database with IMU showed greater cross-subject accuracy versus other methods. Single subject external tests are limited to six repetitions per motion versus 60 for cross-subject, impacting ResNet50 training. Tab. 1 summarizes all investigation results. The ResNet50 network requires 22-25MB memory for four motion classification from 300x300 images, readily met by many embedded systems.

7 Prospects and Challenges

The hand-engineered features provide model interpretability and reduce computational load, desirable for sEMG systems. Future work includes optimizing window/stride sizes. A key limitation is the small subject/trial size versus typical deep learning data scales. Cross-subject classification would benefit from more subjects and motion trials.

The 200Hz MyoMex sampling rate is below EMG signal bandwidth. This undersampling challenges motion classification but reduces data volume and computation. The techniques presented could transform other time series stochastic data. Merits of machine learning are tested under hardware constraints common in research Pizzolato et al. (2017).



(a)

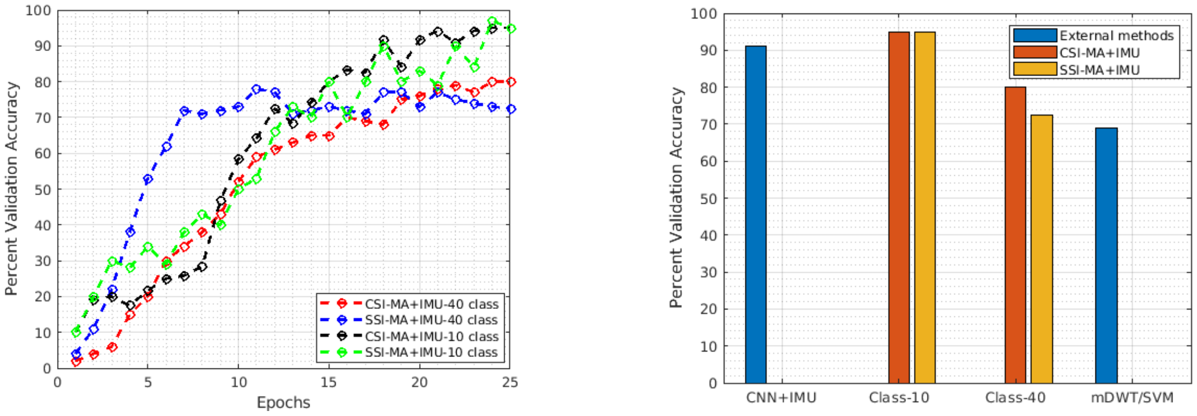
(b)

(c)

(d)

(e)

Fig. 11: Average performance of MA and SF features in terms of percent validation accuracy for (a) configuration A, (b) configuration B, (c) configuration C without data augmentation, (d) and with data augmentation, (e) and for single subject investigation (SSI).



(a) (b)

Fig. 12: (a) Mean validation accuracy profile on external database Pizzolato et al. (2017). (b) Comparison between methods in terms of validation accuracy reported in Pizzolato et al. (2017); Sadhu (2019) and proposed.

Tab. 1: A summary of all the investigations performed in this paper (RF: Relative Feature, DA: Data Augmentation, CSI: Cross Subject Investigation, and SSI: Single Subject Investigation).

Experiment	Data Configuration	IMU	RF	DA	Feature	Test Size	Validation accuracy
CSI	Default	NO	NO	NO	MA	20%	40.00%
	motion classes	NO	NO	NO	SF	20%	36.00%
	(offline learning)	NO	NO	YES	MA	20%	35.00%
		NO	NO	YES	SF	20%	32.00%
	Reduced	NO	YES	NO	MA	20%	52.00%
	motion classes	NO	YES	NO	SF	20%	48.00%
	(offline learning)	NO	YES	YES	MA	20%	50.00%
		NO	YES	YES	SF	20%	44.00%
	External datasets						
	40 Classes	YES	NO	NO	MA	33%	80.12%
	10 Classes	YES	NO	NO	MA	20%	95.00%
SSI	Individual Model	NO	NO	NO	MA	20%	90%
	(offline)	NO	NO	NO	SF	20%	100%
	Real-Time	NO	NO	NO	MA	N/A	85.00%
	(3s processing latency)	NO	NO	NO	SF	N/A	100.00%
	External datasets						
	40 Classes	YES	NO	NO	MA	33%	72.50%
	10 Classes	YES	NO	NO	MA	20%	95.00%

8 Conclusion

A key prosthetics research question is incorporating intelligent systems, with ML/DL enabling core functions like motion classification from sensor arrays. This paper offers a methodology to leverage CNN image classification resources like ResNet for time series data. Multi-channel time series are treated as integrated time-spatial images through feature design.

Two main questions are addressed: generalizing motions user-independently without IMUs is challenging and likely requires more data than utilized here. However, user-specific models may outperform cross-subject ones for personalized control given limited motion data per subject. Hence this paper used in-house facilities for single-subject investigations.

Results suggest subject-specific models predict motions with 90-100% accuracy excluding IMUs using only 64 training images. This enables less experimentation for prosthesis fitting, patient-specific optimization, offline training, and near real-time embedded deployment with little computation. Additional tests found 85-100% accuracy under three second latency constraints.

Comparisons on an external database with IMUs indicate superior performance over existing methods, though significant optimization potential remains.

Acknowledgment

This research is conducted with support and resources received from the Department of Mechanical Engineering, Idaho State University. The research used databases from external sources which is available at (http://ninaweb.hevs.ch/) and (https://data.mendeley.com/datasets/sg9kwnwcp6/1). The research uses MATLAB$^{®}$ routine and setups for MyoMex armband communication which is available at (https://github.com/mark-toma/MyoMex).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Haitham Al-Angari, Gunter Kanitz, Sergio Tarantino, and Christian Cipriani. Distance and mutual information methods for emg feature and channel subset selection for classification of hand movements. *Biomedical Signal Processing and Control*, 27:24–31, 2016. doi: 10.1016/j.bspc.2016.01.011.

M. Arozi, W. Caesarendra, M. Ariyanto, M. Munadi, J.D. Setiawan, and A. Glowacz. Pattern recognition of single-channel semg signal using pca and ann method to classify nine hand movements. *Symmetry*, 12:541, 2020. doi: 10.3390/sym12040541.

M. Atzori and H. Müller. Pawfe: Fast signal feature extraction using parallel time windows. *Frontiers in Neurorobotics*, 13, 2019. doi: 10.3389/fnbot.2019.00074.

M. Atzori, A. Gijsberts, C. Castellini, B. Caputo, A.G.M. Hager, S. Elsig, et al. Electromyography data for non-invasive naturally-controlled robotic hand prostheses. *Scientific Data*, 1, 2014.

M. Atzori, M. Cognolato, and H. Müller. Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands. *Frontiers in neurorobotics*, 10:9, 2016.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*, 2015.

A. Begnall, J. Lines, A. Bostrom, J. Large, and E. Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 31(3):606–660, 2017.

A. Bordes, S. Chopra, and J. Weston. Question answering with subgraph embeddings, 2014. Preprint, https://arxiv.org/abs/1406.3676v3.

F. H. Y. Chan, Yong-Sheng Yang, F. K. Lam, Yuan-Ting Zhang, and P. A. Parker. Fuzzy emg classification for prosthesis control. *IEEE Transactions on Rehabilitation Engineering*, 8(3):305–311, 2000. doi: 10.1109/86.867872.

G. Cheron, J-. Draye, M. Bourgeios, and G. Libert. A dynamic neural network identification of electromyography and arm trajectory relationship during complex movements. *IEEE Transactions on Biomedical Engineering*, 43(5):552–558, 1996. doi: 10.1109/10.488803.

Ulysse Côté Allard, Gabriel Gagnon-Turcotte, Angkoon Phinyomark, Kyrre Glette, Erik Scheme, Francois Laviolette, and Benoit Gosselin. Virtual reality to study the gap between offline and real-time emg-based gesture recognition, 2019.

Berith De la Cruz-Sanchez, M. Arias Montiel, and Marco P. Schoen. Emg pattern recognition to control a hand exoskeleton for assisted rehabilitation. In *Memorias del XXI Congreso Mexicano de Robotica 2019*, Manzanillo, Colima, Mexico, November 2019.

P. Esling and C. Agon. Time series data mining. *ACM Computing Surveys*, 45(1):12:1 – 12:34, 2012.

J. J. Fernandez, K. A. Farry, and J. B. Cheatham. Waveform recognition using genetic programming: the myoelectric signal recognition problem. In *Proceedings of the First Annual Conference of Genetic Programming*, pages 63–71, Stanford University, Stanford, California, USA, 1996.

Y. Geng et al. A robust sparse representation based pattern recognition approach for myoelectric control. *IEEE Access*, 6: 38326–38335, 2018. doi: 10.1109/ACCESS.2018.2851282.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, Las Vegas, NV, 2016.

G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Process. Mag.*, 29:82–97, 2012.

H.A. Javaid, M.I. Tiwana, A. Alsanad, J. Iqbal, M.T. Riaz, S. Ahmad, and F.A. Almisned. Classification of hand movements using myo armband on an embedded platform. *Electronics*, 10:1322, 2021. doi: 10.3390/electronics10111322.

S. Jean, K. Cho, R. Memisevic, and Y. Bengio. On using very large target vocabulary for neural machine translation, 2014. Preprint, https://arxiv.org/abs/1412.2007v2.

A. Krizhevsky, I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25, pages 1097–1105, 2012.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436–444, 2015.

X. Li, O.W. Samuel, X. Zhang, et al. A motion-classification strategy based on semg-eeg signal combination for upper-limb amputees. *J NeuroEngineering Rehabil*, 14(2):1–13, 2017. doi: 10.1186/s12984-016-0212-z.

B. Lucas, A. Shifaz, C. Pelletier, L. O'Neill, N. Zaidi, B. Goethals, F. Petijean, and G.I Webb. Proximity forest: An effective and scalable distance-based classifier for time series. *Data Mining and Knowledge Discovery*, 28:851–881, 2018.

Jonathan Moron, Thomas DiProva, John Cochrane, In Ahn, and Yufeng Lu. Emg-based hand gesture control system for robotics. In *MWSCAS*, pages 664–667, 2018. doi: 10.1109/MWSCAS.2018.8624056.

R. Neamtu, R. Ahsan, E.A. Rundenstainer, G. Sarkozy, E. Keogh, H.A. Dau, C. Nguyen, and C. Lovering. Generalized dynamic time warping: Unleashing the warping power hidden in point-wise distances. In *Proceedings of the IEEE Conference on Data Engineering*, 2018.

A. Olsson, N. Malesevic, A. Bjorkman, and C. Antfolk. An algorithm calibrated with categorically labelled emg for end-to-end estimation of continuous hand kinematics, 2020a. MEC20 Symposium.

A. E. Olsson, A. Björkman, and C. Antfolk. Automatic discovery of resource-restricted convolutional neural network topologies for myoelectric pattern recognition. *Computers in Biology and Medicine*, 120:103723, 2020b.

S. Pizzolato, L. Tagliapietra, M. Cognolato, M. Reggiani, H. Müller, and M. Atzori. Comparison of six electromyography acquisition setups on hand movement classification tasks. *PLoS ONE*, 12(10):e0186132, 2017. doi: 10.1371/journal.pone.0186132.

S. R. Sadhu. Classification of semg signals and motion data using learning algorithms with application to post-stroke impairment, 2019.

E. Scheme, E. Castillo-Guerra, K. Englehart, and A. Kizhanatham. Practical considerations for real-time implementation of speech-based gender detection. In J.F. Martínez-Trinidad, J.A. Carrasco Ochoa, and J. Kittler, editors, *Progress in Pattern Recognition, Image Analysis, and Applications*, volume 4225 of *Lecture Notes in Computer Science*, Berlin, Heidelberg, 2006. Springer. doi: 10.1007/11892755_44.

E. J. Scheme, B. Hudgins, and P. A. Parker. Myoelectric signal classification for phoneme-based speech recognition. *IEEE Transactions on Biomedical Engineering*, 54(4):694–699, 2007. doi: 10.1109/TBME.2006.889175.

S. Summa, R. Gori, L. Freda, E. Castelli, and M. Petrarca. Development of a dynamic oriented rehabilitative integrated system (doris) and preliminary tests. *Sensors*, 19:3402, 2019.

A. Toro-Ossaba, J. Jaramillo-Tigreros, J.C. Tejada, A. Peña, A. López-González, and R.A. Castanho. Lstm recurrent neural network for hand gesture recognition using emg signals. *Appl. Sci.*, 12:9700, 2022. doi: 10.3390/app12199700.

P. Tsinganos, B. Cornelis, J. Cornelis, et al. Hilbert semg data scanning for hand gesture recognition based on deep learning. *Neural Comput & Applic*, 2020. doi: 10.1007/s00521-020-05128-7.

Z. Wang, Z. Dai, B. Póczos, and J. Carbonell. Characterizing and avoiding negative transfer. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11285–11294, Long Beach, CA, USA, 2019. doi: 10.1109/CVPR.2019.01155.

H. Weytjens and J. De Weerdt. Process outcome prediction: Cnn vs. lstm (with attention). In *Business Process Management Workshops*, volume 397 of *Lecture Notes in Business Information Processing*, Cham, 2020. Springer. doi: 10.1007/978-3-030-66498-5_24.

D. Xiong, D. Zhang, X. Zhao, and Y. Zhao. Deep learning for emg-based human-machine interaction: A review. *IEEE/CAA Journal of Automatica Sinica*, 8(3):512–533, 2021. doi: 10.1109/JAS.2021.1003865.

Abdelhafid Zeroual, Fouzi Harrou, Abdelkader Dairi, and Ying Sun. Deep learning methods for forecasting covid-19 time-series data: A comparative study. *Chaos, Solitons & Fractals*, 140:110121, 2020. doi: 10.1016/j.chaos.2020.110121.