



Guidage non-intrusif d'un bras robotique à l'aide d'un bracelet myoélectrique à électrode sèche

Thèse

Ulysse Côté Allard

Doctorat en génie électrique
Philosophiæ doctor (Ph. D.)

Québec, Canada

Guidage non-intrusif d'un bras robotique à l'aide d'un bracelet myoélectrique à électrode sèche

Thèse

Ulysse Côté-Allard

Sous la direction de:

Benoit Gosselin, directeur de recherche
François Laviolette, codirecteur de recherche

Résumé

Depuis plusieurs années la robotique est vue comme une solution clef pour améliorer la qualité de vie des personnes ayant subi une amputation. Pour créer de nouvelles prothèses intelligentes qui peuvent être facilement intégrées à la vie quotidienne et acceptée par ces personnes, celles-ci doivent être non-intrusives, fiables et peu coûteuses. L'électromyographie de surface fournit une interface intuitive et non intrusive basée sur l'activité musculaire de l'utilisateur permettant d'interagir avec des robots. Cependant, malgré des recherches approfondies dans le domaine de la classification des signaux sEMG, les classificateurs actuels manquent toujours de fiabilité, car ils ne sont pas robustes face au bruit à court terme (par exemple, petit déplacement des électrodes, fatigue musculaire) ou à long terme (par exemple, changement de la masse musculaire et des tissus adipeux) et requiert donc de recalibrer le classifieur de façon périodique.

L'objectif de mon projet de recherche est de proposer une interface myoélectrique humain-robot basé sur des algorithmes d'apprentissage par transfert et d'adaptation de domaine afin d'augmenter la fiabilité du système à long-terme, tout en minimisant l'intrusivité (au niveau du temps de préparation) de ce genre de système.

L'aspect non intrusif est obtenu en utilisant un bracelet à électrode sèche possédant dix canaux. Ce bracelet (3DC Armband) est de notre (Docteur Gabriel Gagnon-Turcotte, mes co-directeurs et moi-même) conception et a été réalisé durant mon doctorat. À l'heure d'écrire ces lignes, le 3DC Armband est le bracelet sans fil pour l'enregistrement de signaux sEMG le plus performant disponible. Contrairement aux dispositifs utilisant des électrodes à base de gel qui nécessitent un rasage de l'avant-bras, un nettoyage de la zone de placement et l'application d'un gel conducteur avant l'utilisation, le brassard du 3DC peut simplement être placé sur l'avant-bras sans aucune préparation. Cependant, cette facilité d'utilisation entraîne une diminution de la qualité de l'information du signal. Cette diminution provient du fait que les électrodes sèches obtiennent un signal plus bruité que celle à base de gel. En outre, des méthodes invasives peuvent réduire les déplacements d'électrodes lors de l'utilisation, contrairement au brassard. Pour remédier à cette dégradation de l'information, le projet de recherche s'appuiera sur l'apprentissage profond, et plus précisément sur les réseaux convolutionnels.

Le projet de recherche a été divisé en trois phases. La première porte sur la conception d'un classifieur permettant la reconnaissance de gestes de la main en temps réel. La deuxième porte

sur l'implémentation d'un algorithme d'apprentissage par transfert afin de pouvoir profiter des données provenant d'autres personnes, permettant ainsi d'améliorer la classification des mouvements de la main pour un nouvel individu tout en diminuant le temps de préparation nécessaire pour utiliser le système. La troisième phase consiste en l'élaboration et l'implémentation des algorithmes d'adaptation de domaine et d'apprentissage faiblement supervisé afin de créer un classifieur qui soit robuste au changement à long terme.

Abstract

For several years, robotics has been seen as a key solution to improve the quality of life of people living with upper-limb disabilities. To create new, smart prostheses that can easily be integrated into everyday life, they must be non-intrusive, reliable and inexpensive. Surface electromyography provides an intuitive interface based on a user's muscle activity to interact with robots. However, despite extensive research in the field of sEMG signal classification, current classifiers still lack reliability due to their lack of robustness to short-term (e.g. small electrode displacement, muscle fatigue) or long-term (e.g. change in muscle mass and adipose tissue) noise. In practice, this means that to be useful, classifier needs to be periodically re-calibrated, a time consuming process.

The goal of my research project is to propose a human-robot myoelectric interface based on transfer learning and domain adaptation algorithms to increase the reliability of the system in the long term, while at the same time reducing the intrusiveness (in terms of hardware and preparation time) of this kind of systems.

The non-intrusive aspect is achieved from a dry-electrode armband featuring ten channels. This armband, named the *3DC Armband* is from our (Dr. Gabriel Gagnon-Turcotte, my co-directors and myself) conception and was realized during my doctorate. At the time of writing, the 3DC Armband offers the best performance for currently available dry-electrodes, surface electromyographic armbands. Unlike gel-based electrodes which require intrusive skin preparation (i.e. shaving, cleaning the skin and applying conductive gel), the 3DC Armband can simply be placed on the forearm without any preparation. However, this ease of use results in a decrease in the quality of information. This decrease is due to the fact that the signal recorded by dry electrodes is inherently noisier than gel-based ones. In addition, other systems use invasive methods (intramuscular electromyography) to capture a cleaner signal and reduce the source of noises (e.g. electrode shift). To remedy this degradation of information resulting from the non-intrusiveness of the armband, this research project will rely on deep learning, and more specifically on convolutional networks.

The research project was divided into three phases. The first is the design of a classifier allowing the recognition of hand gestures in real-time. The second is the implementation of a transfer learning algorithm to take advantage of the data recorded across multiple users, thereby

improving the system’s accuracy, while decreasing the time required to use the system. The third phase is the development and implementation of a domain adaptation and self-supervised learning to enhance the classifier’s robustness to long-term changes.

Contents

Résumé	ii
Abstract	iv
Contents	vi
List of Tables	x
List of Figures	xi
Remerciements	xvii
Foreword	xix
Included Publications	xix
Other publications	xxiii
IEEE Copyright Notice	xxv
Introduction	1
1 {Supervised, Transfer, Domain Adaptation, Self-supervised} Learning	7
1.1 Supervised Learning	8
1.2 Transfer Learning	17
1.3 Unsupervised Domain Adaptation	18
1.4 Self-supervised Learning	18
1.5 Bibliography	19
2 A convolutional neural network for robotic arm guidance using sEMG based frequency-features	22
2.1 Reference	22
2.2 Context	22
2.3 Résumé	23
2.4 Abstract	23
2.5 Introduction	24
2.6 Proposed guidance system overview	26
2.7 Classifier overview	28
2.8 Experimental Results	31
2.9 Conclusion	36
2.10 Bibliography	37

3	Deep learning for electromyographic hand gesture signal classification using transfer learning.	41
3.1	Reference	41
3.2	Context	41
3.3	Résumé	42
3.4	Abstract	43
3.5	Introduction	43
3.6	Related Work	45
3.7	sEMG datasets	46
3.8	Classic sEMG Classification	49
3.9	Deep Learning Classifiers Overview	50
3.10	Transfer Learning	54
3.11	Classifier Comparison	57
3.12	Real-Time Classification and Medium Term Performances (case study) . . .	60
3.13	Discussion	62
3.14	Conclusion	64
3.15	Acknowledgements	65
3.16	Bibliography	65
3.A	Data Augmentation	73
3.B	Deep Learning on Embedded Systems and real-time classification	74
3.C	Feature Engineering	75
3.D	Hyperparameters selection for state of the art feature sets.	81
3.E	Dimensionality Reduction on the Myo Armband Dataset for State of the Art Feature Set	81
3.F	Reducing the number of EMG channels on the target dataset	82
4	A Low-Cost, Wireless, 3-D-Printed Custom Armband for sEMG Hand Gesture Recognition	83
4.1	Reference	83
4.2	Context	83
4.3	Résumé	84
4.4	Abstract	84
4.5	Introduction	85
4.6	Overview of Surface EMG Acquisition Systems	86
4.7	The 3DC Armband (Prototype)	88
4.8	Comparison Dataset	94
4.9	Comparison Methods	96
4.10	Results	99
4.11	Discussion	103
4.12	Conclusions	104
4.13	Authors Contribution	104
4.14	Funding	105
4.15	Acknowledgments	105
4.16	Conflicts of interest	105
4.17	Ethical statement	105
4.18	Abbreviations	105
4.19	Bibliography	106
4.A	Confusion Matrices	111

5	Interpreting Deep Learning Features for Myoelectric Control: A Comparison with Handcrafted Features	116
5.1	Reference	116
5.2	Context	116
5.3	Résumé	117
5.4	Abstract	117
5.5	Introduction	118
5.6	Material and Methods	120
5.7	Results	134
5.8	Discussion	144
5.9	Conclusion	147
5.10	Bibliography	149
6	Virtual Reality to Study the Gap Between Offline and Real-Time EMG-based Gesture Recognition	158
6.1	Reference	158
6.2	Context	158
6.3	Résumé	158
6.4	Abstract	159
6.5	Introduction	159
6.6	Long-term sEMG Dataset	161
6.7	Deep Learning Classifiers	166
6.8	Transfer Learning	169
6.9	Results	170
6.10	Discussion	174
6.11	Conclusion	175
6.12	Bibliography	176
7	Unsupervised Domain Adversarial Self-Calibration for Electromyographic-based Gesture Recognition	180
7.1	Reference	180
7.2	Context	180
7.3	Résumé	180
7.4	Abstract	181
7.5	Introduction	181
7.6	Datasets and Network's Architecture	184
7.7	Unsupervised Domain Adaptation	188
7.8	Unsupervised Self-Calibration	192
7.9	Experiments and results	195
7.10	Discussion	197
7.11	Conclusion	202
7.12	Bibliography	203
7.A	ConvNet's comparison with Handcrafted feature sets	208
7.B	Pseudo-labeling Heuristic	211
	Conclusion	214
8.3	Discussion	215
8.4	Scientific Contributions Overview	215

8.5 Future works	216
Bibliography	218

List of Tables

2.1	Participants general information and first two tasks results	33
3.1	Transfer learning comparison on the Evaluation Dataset	58
3.2	Classifier comparison on the Evaluation Dataset	58
3.3	Transfer learning comparison on the NinaPro DB5	59
3.4	Classifiers comparison on the NinaPro DB5	60
3.5	Comparison of the five data augmentation techniques proposed.	74
3.6	Classification accuracy on the <i>Evaluation dataset</i> for the feature sets with and without dimensionality reduction.	82
4.1	Comparison of surface electromyography acquisition systems	87
5.1	Handcrafted features extracted for topological landmarks sorted by functional group.	124
5.2	Members of nodes labeled in Figure 5.6. LeFX refers to a <i>Learned Feature</i> from block X	141
5.3	Accuracy obtained on the test set using the handcrafted features and the learned features from their respective block. The <i>Single Feature</i> accuracies are given as the average accuracy over all the features of their respective block/category. . .	142
6.1	Leap motion ConvNet’s architecture	168
7.1	Offline accuracy for seven gestures	196
7.2	Offline accuracy for eleven gestures	197
7.3	Dynamic dataset’s accuracy for eleven gestures using Training Recordings as unlabeled data	199
7.4	Dynamic dataset’s accuracy for eleven gestures using the first Evaluation Recording as unlabeled data	200
7.5	Accuracy for the Recalibration and Recalibration SCADANN with eleven gestures on the dynamic dataset using the first Evaluation Recording as unlabeled data	201
7.6	Comparison between the ConvNet employed in this work and Handcrafted feature sets	210

List of Figures

1.1	What is hard to do? An XKCD illustration	7
1.2	underfitting versus overfitting	10
1.3	Feedforward neural network	14
1.4	Inside a ConvNet	16
2.1	The 7 gestures considered in this article	28
2.2	The convolutional network employed in this article	30
2.3	Box plot of participant's completion time of task 3	33
2.4	Bar plot of the speed test	34
2.5	Inter-day classification: Average accuracy over a 6-day trials	35
2.6	Muscle fatigue: Average accuracy over a five-minute period	36
3.1	The 7 hand/wrist gestures considered in the <i>Myo Dataset</i>	48
3.2	Examples of the range of armband placements on the subjects' forearm	48
3.3	Slow-fusion convolutional network architecture	51
3.4	Proposed spectrogram convolutional network architecture	52
3.5	Proposed continuous wavelet transform network architecture	53
3.6	The raw EMG convolutional network	54
3.7	Proposed enhanced raw EMG convolutional network	54
3.8	Diagram of the proposed transfer learning architecture	57
3.9	Out-of-sample classification accuracy on the NinaPro DB5 with the proposed transfer learning algorithm	61
3.10	Inter-day classification: Average accuracy over 14 days	62
3.11	Muscle fatigue: Average accuracy over a five minutes period	63
3.12	Visual comparison between continuous wavelet transform and short-time Fourier transform	80
4.1	The 3DC Armband	88
4.2	System-level concept of the multichannel wireless sEMG sensor	89
4.3	Photos of the proposed electromyographic sensor	90
4.4	Bandwidth comparison between the 3DC and the Myo Armband	91
4.5	3D model of the 3DC Armband form factor	93
4.6	The two different armband configuration	95
4.7	The eleven hand/wrist gestures employed in this article	95
4.8	Comparison of the signals recorded via the Myo and 3DC Armband	96
4.9	The proposed raw convolutional network architecture	98
4.10	The proposed spectrogram convolutional network architecture	99

4.11	Comparison of the average accuracy obtained by the linear discriminant classifier in respect to the Myo and 3DC Armband over 1, 2, 3 and 4 cycles of training	100
4.12	Confusion Matrices for the Myo and the 3DC Armband employing linear discriminant analysis for classification and four cycles of training.	100
4.13	Comparison of the average accuracy obtained by the <i>Raw</i> convolutional network in respect to the Myo and 3DC Armband over 1, 2, 3 and 4 cycles of training	101
4.14	Confusion Matrices for the Myo and the 3DC Armband employing the <i>Raw</i> ConvNet for classification and four cycles of training	101
4.15	Comparison of the average accuracy obtained by the <i>Spectrogram</i> convolutional network in respect to the Myo and 3DC Armband over 1, 2, 3 and 4 cycles of training	102
4.16	Confusion Matrices for the Myo and the 3DC Armband employing the <i>Spectrogram</i> ConvNet for classification and four cycles of training	102
4.17	Confusion Matrices for the Myo and the 3DC Armband employing linear discriminant classifier for classification and one cycle of training	111
4.18	Confusion Matrices for the Myo and the 3DC Armband employing linear discriminant classifier for classification and two cycles of training	111
4.19	Confusion Matrices for the Myo and the 3DC Armband employing linear discriminant classifier for classification and three cycles of training	112
4.20	Confusion Matrices for the Myo and the 3DC Armband employing the <i>Raw</i> ConvNet for classification and one cycle of training	112
4.21	Confusion Matrices for the Myo and the 3DC Armband employing the <i>Raw</i> ConvNet for classification and two cycles of training	113
4.22	Confusion Matrices for the Myo and the 3DC Armband employing the <i>Raw</i> ConvNet for classification and three cycles of training	113
4.23	Confusion Matrices for the Myo and the 3DC Armband employing the <i>Spectrogram</i> ConvNet for classification and one cycle of training	114
4.24	Confusion Matrices for the Myo and the 3DC Armband employing the <i>Spectrogram</i> ConvNet for classification and two cycles of training	114
4.25	Confusion Matrices for the Myo and the 3DC Armband employing the <i>Spectrogram</i> ConvNet for classification and three cycles of training	115
5.1	Diagram of the workflow of this work. The <i>3DC Dataset</i> is first preprocessed before being used to train the network using standard training and the proposed ADANN training procedure. The handcrafted features are directly calculated from the preprocessed dataset, while the deep features are extracted from the ConvNet trained with ADANN. In the diagram, the blue rectangles represent experiments and the arrows show which methods/algorithms are required to perform them.	121
5.2	The eleven hand/wrist gestures recorded in the <i>3DC Dataset</i> (image re-used from [19])	122
5.3	The ConvNet’s architecture, employing 543,629 learnable parameters. In this figure, B_i refers to the i th feature extraction block ($i \in \{1, 2, 3, 4, 5, 6\}$). Conv refers to Convolutional layer. As shown, the feature extraction is performed after the non-linearity (leaky ReLU).	126

5.4	Overview of the training steps of ADANN (identical to DANN) for one labeled batch from the source ($\{x_s, y_s\}$, blue lines) and one unlabeled batch from the target ($\{x_t\}$, red dashed lines). The purple dotted lines correspond to the backpropagated gradient. The gradient reversal operation is represented by the purple diamond.	128
5.5	An example of step 3 of the Mapper algorithm with $W = 2$. The purple dots represent the elements of \mathcal{W} . In (A), the red square corresponds to \mathfrak{C} . In (B), \mathfrak{C} is subdivided using k^2 squares of length H (with $k = 2$ in this case). The orange diamonds, in both (B) and (C), represent the elements of \mathcal{V} . Finally, the square $\mathfrak{c}_{\vec{v}}$ of length D is shown on the upper left corner of (C), overlapping other squares centered on other elements of \mathcal{V} (dotted lines).	132
5.6	Topological network generated exclusively for the handcrafted features, where nodes are colored to indicate percent composition of: (a) signal amplitude and power features (SAP), (b) nonlinear complexity (NLC), (c) frequency information features (FI), (d) time series modeling features (TSM), and (e) unique features (UNI). Dashed boxes highlight dense groupings of the specified functional group in each of the networks.	135
5.7	Classification results of deep learning architectures. A) Per-participant test set accuracy comparison when training the network with and without ADANN, B) Confusion matrices on the test set for cross-subject training with and without ADANN.	136
5.8	Output of Guided Grad-CAM when asked to highlight specific gestures in an example. For all graphs, the y-axis of each channel are scaled to the same range of value (indicated on the first channel of each graph). Warmer colors indicate a higher 'importance' of a feature in the input space for the requested gesture. The coloring use a logarithmic scale. For visualization purposes, only features that are within three order of magnitudes to the most contributing feature are colored. (A) The examples shown are real examples and correspond to the same gestures that Guided Grad-CAM is asked to highlight. (B) A single example, generated using Gaussian noise of mean 0 and standard deviation 450, is shown three times. While the visualization algorithm does highlight features in the input space (when the requested gesture is not truly present in the input), the magnitude of these contributions is substantially smaller (half or less) than when the requested gesture is present in the input.	137
5.9	Topological network generated for exclusively the learned features, where nodes are colored to indicate percent composition of: (A) Block 1's features, (B) Block 2's features, (C) Block 3's features, (D) Block 4's features, (E) Block 5's features, and (F) Block 6's features. Dashed boxes highlight dense groupings of the specified block features in each of the networks.	139
5.10	Topological network generated for all features, where nodes were colored to indicate percent composition of learned features. The dashed boxes highlight dense grouping of handcrafted features with their associated type.	140

5.11	Confusion matrices using the handcrafted features and the learned features from the first, penultimate and last block as input and a LDA as the classifier. The first column, denoted as <i>All features</i> , shows the confusion matrices when using all 64 learned features of Block 1, 5 and 6 respectively (from top to bottom) and the set of UNI handcrafted features. The next five columns, denoted as <i>Single Feature</i> , show the confusions matrices for handcrafted feature exemplars and from the same network's blocks but when training the LDA on a single feature. The subset of learned features was selected as representative of the typical confusion matrices found at each block. The exemplars of the handcrafted features were selected from each handcrafted features' category (in order: SAP, FI, NLC, TSM and UNI).	142
5.12	Mean squared error of the regressions from learned features to handcrafted features, with respect to the number of blocks employed for the regression. The features are grouped with their respective functional groups.	143
6.1	Diagram of the workflow of this article	161
6.2	The eleven hand/wrist gestures considered in this work	162
6.3	The 3DC Armband	163
6.4	Screenshot of the proposed virtual reality software	164
6.5	Images fed to the leap motion's convolutional network	167
6.6	The architecture of the convolutional network used for electromyography-based gesture recognition	169
6.7	Accuracy over-time on the offline dataset	171
6.8	Score obtained by the participants during the evaluation sessions in respect to the classifier's accuracy	171
6.9	Accuracy over-time on the online dataset	172
6.10	Accuracy in respect to the participant's limb orientation	173
6.11	Accuracy in respect to the gesture intensity	173
7.1	The eleven hand/wrist gestures recorded in the <i>Long-term 3DC dataset</i> and the <i>3DC Dataset</i> . The gestures included within the <i>Reduced Long-term 3DC Dataset</i> are encompassed within the green line (7 gestures totals).	185
7.2	The ConvNet's architecture employing 206 548 learnable parameters. In this figure, B_i refers to the i th block ($i \in \{0, 1, 2, 3\}$). Conv refers to a convolutional layer. When working with the reduced dataset, the number of output neurons from the gesture-head are reduced to seven.	187
7.3	The VADA algorithm which simultaneously tries to reduce the divergence between the labeled source ($\{x_s, y_s\}$) and unlabeled target ($\{x_t\}$) dataset while also penalizing violation of the cluster assumption on the target dataset. . . .	191
7.4	Offline accuracy using the TSD DNN for the eleven gestures in respect to time. The values on the x-axis represent the average number of days elapsed across participants since the first session.	198
7.5	TSD DNN dynamic dataset's accuracy for eleven gestures in respect to time. Training is performed offline with the first Training Recording session. Adaptation takes place on the first Evaluation Recording of the corresponding tested session, while the test set comes from the second Evaluation Recording of the same tested session. The values on the x-axis represent the average number of days elapsed across participants since the first session.	198

*Til Alexandra, for **alltid***

46.810118, -71.226128

60.390531, 5.318735

59.925757, 10.714769

WERE ALL STARS TO
DISAPPEAR OR DIE,
I SHOULD LEARN TO LOOK
AT AN EMPTY SKY
AND FEEL ITS TOTAL DARK
SUBLIME,
THOUGH THIS MIGHT TAKE
ME A LITTLE TIME

M. H. AUDEN

Remerciements

Je remercie mon directeur de recherche, Prof. Benoit Gosselin de l'Université Laval pour m'avoir supporté tout au long de mes études graduées. Je le remercie surtout d'avoir si généreusement accueilli au sein de son laboratoire un étudiant qui n'avait pas les meilleures notes, mais des idées pleines la tête et une passion apparente pour la recherche. Je le remercie également de m'avoir supporté malgré des changements massifs dans mon domaine de recherche (originellement, j'allais travailler avec l'électroencéphalographie). Je tiens également à remercier mon directeur de recherche, Prof. François Laviolette d'avoir vu en moi un potentiel que je soupçonnais pas. Je le remercie profondément également de toujours avoir pris le temps de m'aider lorsque j'en avais besoin malgré un emploi du temps qui devait l'occuper plus de 26h par jour. Je veux également le remercier de m'avoir ouvert l'esprit quant aux mathématiques qui ont toujours été une faiblesse pour moi. Il m'a donné l'envie de réellement apprendre à utiliser ce merveilleux outil et bien que mon cheminement est loin d'être terminé (et ne le sera jamais), grâce à lui je me sens mieux équipé pour naviguer cette apprentissage. Je tiens également à remercier mes deux directeurs pour leur soutien et les nombreuses opportunités qu'ils m'ont offert. En particulier de m'avoir permis d'assister à de nombreuses conférences qui ont été des moments clés durant mes études graduées. Je suis privilégié de les avoir eu comme directeurs de recherche.

Je remercie Alexandra, kjæresten min, sans qui je n'aurais tout simplement jamais réussi à faire ce travail. Elle m'a accompagnée tout au long de mes études graduées malgré l'océan qui nous séparait. Elle était là et m'a supporté dans tous mes moments de doutes et d'échecs (trop nombreux pour quantifier) et dans mes succès (beaucoup plus facile à quantifier). Elle m'a également aidé à réviser tous mes articles afin que je communique clairement mes idées dans ceux-ci. Elle était souvent la seule personne qui pouvait me redonner le sourire dans les moments difficiles et la motivation de pouvoir finalement vivre ensemble de façons permanente sans avoir s'inquiéter des visas m'a poussé au-delà de mes limites. Alexandra, merci sincèrement d'être toi avec moi.

Je tiens également à remercier mes parents Pierre et Rachel, pour leur soutien et amour inébranlable tout au long de ma vie (littéralement). Il n'y a pas de manuel pour être parents, mais vous avez très bien fait cela et j'espère pouvoir continuer à vous rendre fier. Je veux

également remercier ma soeur, Sarine, pour avoir été là quand j'en avais besoin. Je remercie également mes amis pour leur soutien. Merci tout particulièrement à Antoine et Rosemarie pour nos grandes discussions dans un sauna qui ont réchauffé mon cœur et à Max, Fred avec qui j'ai grandi et qui m'ont fait grandir.

Je veux également remercier Prof Kyrre Glette et Prof Erik Scheme de m'avoir généreusement invité dans leur université respective (université d'Oslo et Université du Nouveau-Brunswick) et de m'avoir permis d'évoluer dans un nouvel environnement académique. Les leçons que j'ai apprises grâce à vous m'ont permis d'être un meilleur chercheur.

Merci également aux professeurs Philippe Giguère, Alexandre Campeau-Lecours, Richard Khoury, David St-Onge et François Nougrou, ce fut un véritable plaisir de collaborer avec vous. Merci de m'avoir aidé à mieux comprendre vos domaines et de m'avoir aidé à apporter une petite contribution dans ceux-ci.

Finalement je tiens à remercier toutes les personnes avec qui j'ai eu la chance de collaborer durant mes études graduées. Merci particulièrement à Luc Lamontagne d'avoir été le premier à me donner une chance en recherche et de m'avoir toujours épaulé dans mes choix. Merci à Gabriel Gagnon-Turcotte, Cheikh Latyr Fall, Angkoon Phinyomark, Evan Campbell et Gabriel Dubé d'avoir substantiellement contribué aux travaux présentés dans cette thèse. Je remercie également tous les membres du GRAAL qui m'ont fournis un environnement d'apprentissage merveilleux. Merci particulièrement à Alexandre Lacoste, Jean-François Roy, Pascal Germain et Alexandre Drouin de m'avoir servi de modèle vers ce que je veux accomplir comme chercheur. Merci aussi à {Jonathan Beregeron et Nicolas Garneau} (Jim Stoppani). Jonathan Gingras, Amélie Rolland, Frédéric Paradis, Gaël Letarte, Hana Ajakan, Mazid Abiodoun, Prudencio Tossou, Maxime Tremblay, Mathieu Alain, Mathieu Godbout, Frédéric Leblanc et Roxane Debruyker. Vous avez tous contribué, grâce à vos discussions et collaboration à ma formation en tant que chercheur.

Foreword

This thesis presents a selection of six articles written during my doctoral studies, which were published or are currently under peer-review. These works build on each other and form a clear path of my contributions to the field of surface electromyography-based gesture recognition. My contributions to these articles is as follows:

Included Publications

All article were added to the thesis as is, without modification from their published/submitted format, except to put them in the same format as this thesis.

A convolutional neural network for robotic arm guidance using sEMG based frequency-features

Ulysse Côté-Allard, François Nougrou, Cheikh Latyr Fall, Philippe Giguère, Clément Gosselin, François Laviolette, and Benoit Gosselin. In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2464-2470. IEEE, 2016.

Published:

yes (conference)

Date of publication:

09/10/2016

Author Contributions:

U.C.-A and F.N. designed the experiments. U.C.-A. conducted the experiments, acquired the data, devised the convolutional architecture and its implementation, and wrote the manuscript. C.L.F. wrote the communication protocol for the robotic arm. F.N. contributed to writing, revising, and enhancing the assistive and rehabilitation aspect of the manuscript. P.G., F.L. contributed to revising, and enhancing the technical descriptions in the manuscript. C.G., F.L.

and B.G. contributed by their ideas, guidance and supervision. All authors contributed to editing and revising the manuscript.

Deep Learning for Electromyographic Hand Gesture Signal Classification Using Transfer Learning

Ulysse Côté-Allard, Cheikh Latyr Fall, Alexandre Drouin, Alexandre Campeau-Lecours, Clément Gosselin, Kyrre Glette, François Laviolette[‡], and Benoit Gosselin[‡]. "Deep learning for electromyographic hand gesture signal classification using transfer learning." IEEE Transactions on Neural Systems and Rehabilitation Engineering 27, no. 4 (2019): 760-771. [‡]These authors share senior authorship

Published:

yes (journal)

Date of publication:

31/01/2019

Author Contributions:

U.C.-A designed and conducted the experiments, acquired the data, devised the transfer learning algorithm and its implementation, and wrote the manuscript. C.L.F. conducted a literature review on deep learning in embedded system. A.D. contributed to writing, revising, and enhancing the technical descriptions in the manuscript. A.C.L. contributed to and revised the assistive and rehabilitation aspect of the manuscript. C.G and K.G. contributed to the hardware required to conduct the experiment. F.L. and B.G. contributed by their ideas, guidance and supervision. All authors contributed to editing and revising the manuscript.

A Low-Cost, Wireless, 3-D-Printed Custom Armband for sEMG Hand Gesture Recognition.

Ulysse Côté-Allard[†], Gabriel Gagnon-Turcotte[†], François Laviolette, and Benoit Gosselin. Sensors 19, no. 12 (2019): 2811.

[†]These authors contributed equally to this work

Published:

yes (journal)

Date of publication:

24/06/2019

Author Contributions:

U.C.-A and G.G.-T designed and conducted the experiments, acquired the data, created the armband, and wrote the manuscript. G.G.-T designed and assembled all the microelectronics component and wrote the communication protocol. U.C.-A. designed and 3D printed the armband's form factor. U.C.-A. and G.G.-T assembled the armbands. U.C.-A. designed and implanted the machine learning experiments. B.G. contributed the necessary hardware for the armband's construction. F.L. and B.G. contributed by their ideas, guidance and supervision. All authors contributed to editing and revising the manuscript.

Interpreting Deep Learning Features for Myoelectric Control: A Comparison with Handcrafted Features.

Ulysse Côté-Allard[†], Evan Campbell[†], Angkoon Phinyomark, François Laviolette[‡], Benoit Gosselin[‡], and Erik Scheme[‡]. *Frontiers in Bioengineering and Biotechnology*, no. 8 (2020), p.158.

[†]These authors contributed equally to this work

[‡]These authors share senior authorship

Published:

Yes (journal)

Date of submission:

30/11/2019

Author Contributions:

U.C.-A., E.C., A.P., F.L., B.G. and E.S. designed the experiments. U.C.-A. and E.C. conducted the experiments and wrote the manuscript. U.C.-A. designed and implemented the deep learning algorithms. E.C. produced the MAPPER analysis. U.C.-A. wrote the mathematical description of the MAPPER algorithm. U.C.-A. and E.C. ran the hybrid experiments. A.P., F.L., B.G. and E.S. contributed by their ideas, guidance and supervision. All authors contributed to editing and revising the manuscript.

Virtual Reality to Study the Gap Between Offline and Real-Time EMG-based Gesture Recognition

Ulysse Côté-Allard, Angkoon Phinyomark, Erik Scheme[‡], François Laviolette[‡] and Benoit Gosselin[‡]. arXiv preprint arXiv (2019).

[‡]These authors share senior authorship

Published:

No (In peer-review)

Date of submission:

9/12/2019 U.C.-A., A.P. and E.S. designed the experiments. U.C.-A. conducted the experiments, designed and implemented the deep learning algorithms, performed the offline and real-time analysis and wrote the manuscript. A.P., F.L., B.G. and E.S. contributed by their ideas, guidance and supervision. All authors contributed to editing and revising the manuscript.

Unsupervised Domain Adversarial Self-Calibration for Electromyographic-based Gesture Recognition

Ulysse Côté-Allard, Angkoon Phinyomark, Erik Scheme[‡], François Laviolette[‡] and Benoit Gosselin[‡]. arXiv preprint arXiv (2019).

[‡]These authors share senior authorship

Published:

No (In peer-review)

Date of submission:

9/12/2019

Author Contributions:

U.C.-A. designed the experiments and SCADANN, wrote the implementation of all the algorithms, performed the test and analysis and wrote the manuscript. A.P., F.L., B.G. and E.S. contributed by their ideas, guidance and supervision. All authors contributed to editing and revising the manuscript.

Other publications

I have also contributed, either as first or second author, to seven other peer-reviewed, published articles not included into the main body of this thesis, as their topic was too divergent from that of the thesis. Below is a list of these articles in chronological order.

Optimizing Question-Answering Systems Using Genetic Algorithms.

Ulysse Côté-Allard, Richard Khoury, Luc Lamontagne, Jonathan Bergeron, François Laviolette, and Alexandre Bergeron-Guyard. In The Twenty-Eighth International Flairs Conference. 2015.

Published:

yes (conference)

Author Contributions:

U.C.-A. Analyzed the problem, designed the genetic algorithm-based solution and implemented it. R.K. wrote the manuscript. U.C.A. and J.B. annotated and ran the experiments. R.K., L.L., F.L. contributed by their ideas, guidance and supervision. A.B.-G. contributed financially.

Towards the use of consumer-grade electromyographic armbands for interactive, artistic robotics performances

Ulysse Côté-Allard, David St-Onge, Philippe Giguère, François Laviolette, and Benoit Gosselin. "Towards the use of consumer-grade electromyographic armbands for interactive, artistic robotics performances." In 2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pp. 1030-1036. IEEE, 2017.

Published:

yes (conference)

Author Contributions:

U.C.-A. and D.S.-O designed and conducted the data collection. U.C.-A. designed, implemented and ran the machine learning pipeline. U.C.-A. and D.S.-O wrote the manuscript. P.G., F.L. and B.G. contributed by their ideas, guidance and supervision.

Time adaptive dual particle swarm optimization

Ulysse Côté-Allard[†], Gabriel Dubé[†], Richard Khoury, Luc Lamontagne François Laviolette, and Benoit Gosselin. In 2017 IEEE Congress on Evolutionary Computation (CEC), pp. 2534-2543. IEEE, 2017.

[†]These authors contributed equally to this work

Published:

yes (conference)

Author Contributions:

U.C.-A. and G.D. designed the optimization algorithm, ran the experiments, perform the analysis and wrote the manuscript. R.K., L.L., F.L. and B.G. contributed by their ideas, guidance and supervision. All authors contributed to editing and revising the manuscript.

Intuitive adaptive orientation control of assistive robots for people living with upper limb disabilities

Dinh-Son Vu[†], **Ulysse Côté-Allard**[†], Clément Gosselin, François Routhier, Benoit Gosselin, and Alexandre Campeau-Lecours. In 2017 International Conference on Rehabilitation Robotics (ICORR), pp. 795-800. IEEE, 2017.

[†]These authors contributed equally to this work

Published:

yes (conference)

Author Contributions:

A.C.-L designed and implemented the new orientation algorithm and designed the experiments. U.C.-A. helped designed the algorithm. D.-S.V. conducted the experiment. A.C.-L, U.C.-A. and D.-S.V. wrote the manuscript. F.R., C.G., B.G. and A.C.-L. contributed by their ideas, guidance and supervision. All authors contributed to editing and revising the manuscript.

Intuitive Adaptive Orientation Control for Enhanced Human–Robot Interaction

Alexandre Campeau-Lecours, **Ulysse Côté-Allard**, Dinh-Son Vu, François Routhier, Benoit Gosselin, and Clément Gosselin. IEEE Transactions on Robotics 35, no. 2 (2018): 509-520.

Published:

yes (journal)

Author Contributions:

A.C.-L designed and implemented the new orientation algorithm, designed and conducted the experiments. U.C.-A. helped designed the algorithm and designed, conducted and analysed the Amazon Turk experiment. A.C.-L, U.C.-A. and D.-S.V. wrote the manuscript. F.R., C.G., B.G. contributed by their ideas, guidance and supervision. All authors contributed to editing and revising the manuscript.

Engaging with Robotic Swarms: Commands from Expressive Motion

David St-Onge, **Ulysse Côté-Allard**, Kyrre Glette, Benoit Gosselin and Giovanni Beltrame. IEEE Transactions on Robotics 35, no. 2 (2018): 509-520.

Published:

yes (journal)

Author Contributions:

D.S.-O. designed and implemented the algorithms relating to the robots, collected the data. U.C.-A. designed and implemented the machine learning algorithms and ran the experiments. D.S.-O. and U.C.-A. wrote the manuscript (D.S.-O. wrote the majority). K.G., B.G. and G.B. contributed by their ideas, guidance and supervision. All authors contributed to editing and revising the manuscript.

IEEE Copyright Notice

As this document contains multiples articles published in IEEE conferences/journals, here is the copyright message for this thesis diffusion:

"In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of University Laval's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink. If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation."

Introduction

Intrinsic to the human experience, is being limited by our physical attributes. As such, humans have sought to evolve their surroundings to fully leverage their body's capabilities. However in some cases, where surroundings have been manipulated to benefit the majority, people living with disabilities can be adversely affected as they must navigate an environments built for another standard. In this context, robotics and artificial intelligence offer an attractive avenue to improve quality of life by democratizing devices which enables users to perform a wider range of day-to-day tasks more efficiently [4]. In the case of hand amputees, this translates into smarter and less expensive prosthetics that can be controlled seamlessly.

For hand prosthetic control, muscle activity provides an intuitive interface on which to perform hand gesture recognition [9]. This activity can be recorded by surface electromyography (sEMG), a non-invasive technique widely adopted both in research and clinical settings. The sEMG signals, which are non-stationary, represent the sum of subcutaneous motor action potentials generated through muscular contraction [9]. Despite extensive research in the field, the currently available myoelectric prosthetics fail to perform sufficiently robust and accurate multi-hand gesture recognition for day-to-day tasks [2, 6]. On the other hand, research papers routinely report multi-gestures classifications accuracy above 95% accuracy. This disconnect in reported offline accuracy versus online usability mainly stems from the four main dynamic factors of sEMG signals [10]:

- **Gesture Intensity** - Users perform the same gesture with varying degree of muscle contraction
- **Limb Position** - In real-life use, gestures will have to be performed in different positions which will alter the nature of the sEMG signal due to different eccentric/concentric contractions, gravitational forces, muscle compression, etc.
- **Electrode Shift** - Each time the users dons a myoelectric system, the electrodes will likely be placed in slightly different position.
- **Transient Changes in EMG** - Additional factors that alter the sEMG signal over-time (e.g. muscle fatigue, electrode impedance changes, non-stationarity of the signal).

Labeled myoelectric signals are also extremely time-consuming to obtain and must be recorded for each user as extensive variability exist between subjects [5] This last factor means that, in practice, sEMG datasets used as benchmarks for offline classification rarely contain even a single of these dynamic factors. In fact, the chasm is such that some authors reported significant different between method in an online setting, when there was none offline (when obtained from a dataset without the main dynamic factors) [12]. Additionally, due to the non-stationarity of sEMG signals, even a classifier trained on a dataset containing the four main dynamic factors will degrade over time, forcing the user to re-calibrate the system periodically. Moreover, myoelectric prosthetic are expensive, costing between 20 000 and 100 000USD [8].

Transradial prosthesis

Most of the research in EMG pattern recognition are motivated by prosthetic control, and most often for people with transradial amputations [10]. Transradial prosthesis refer to artificial limbs designed to replace a missing arm below the elbow [8]. This thesis however, focuses on a general human-computer interaction approach for both amputees and non-amputees. Nevertheless, to be able to contextualize most of the body of works referenced throughout this thesis, it is important to understand the capabilities and functions of prosthesis currently available.

Roughly speaking, upper limb prosthetics can be divided into two categories based on their functionality: passive and active prostheses [7].

Passive prosthetics can be further divided into functional and cosmetics prostheses. The functional prosthetics' goal is to facilitate the completion of specific activities (e.g. sports) (see Fig 0.1).



Figure 0.1 – Example of a passive functional prosthetic. Image taken from <https://www.armdynamics.com>

Cosmetics prostheses are designed to look like a natural limb (see Fig 0.2 for an example). These prostheses are lightweight. Despite their lack of active movement, they can provide a

user a surface to carry/stabilize object. Additionally, while they do not have active movement, they can be equipped with multi-positional joints providing more options to the user in how to interact with its environment. Despite their apparent limited functions, cosmetic prostheses "tend to have a higher rate of permanent use both in adult and pediatric populations, with less temporary disuse stemming from decreased discomfort and maintenance requirements" [3]. They also tend to cost substantially less (between 3 000 and 5 000USD) [8].



Figure 0.2 – Example of a passive cosmetic prosthetic. Image taken from <https://www.armdynamics.com>

The second type of prostheses (active prostheses) are designed to be actuated by the user without having to physically move the prosthetic's joints. They are also divided into two sub-categories, that is, body-powered and externally powered. Body-powered prostheses employ a cable attached to a sound body part of the wearer. By moving this body part, or extending the amputated limb, the cable actuates the prosthetic hand [7]. This allows the user to easily control its prosthetic movement with the desired strength intensity. The main drawback of such a system is the high amount of energy from the wearer required to utilize this apparatus throughout the day [7]. Despite this, the most popular prosthetic to date remain the single degree of freedom, body-powered split hook [1], examples of which can be seen in Fig. 0.3.

Externally powered prosthetics, as their name indicates, employ an external source of power. Myoelectric prostheses fall within this category. Other forms include force myography [13] and ultrasound imaging [11]. The currently commercially available myoelectric prosthetics are extremely expensive [8] and generally feature a low amount of simultaneously recognized gesture [7]. Weight and general comfort associated with these devices are also an issue [7]. From a signal recognition point of view, newly developed solution must integrate multiple grasp/gesture recognition simultaneously, while contending with a limited amount of computational power (as to limit the weight of the prosthetic and increase comfort) [7]. Additionally, re-calibration requirement should be minimal as otherwise cost-benefit to the user will simply not warrant the use of this type of prostheses.



Figure 0.3 – Example of a body-powered split hook prosthetic. The prosthetic works by having a rubber band holding the split hook in the closed position. A cable is then attached on the opposite shoulder of the wearer which when pull (by rolling the shoulder or moving the arm wearing the prosthetic forward) open the split hook. Note that this behavior can be reversed so that the default position is open. Image from: wired.com/2012/03/ff_prosthetics/

Thesis Goal and Outline

The goal of this thesis is twofold. First, provides new tools (hardware, software and experimental protocols) from which to study sEMG signals, as to better understand both the main dynamic factors and the type of information that are pertinent to extract from such signals. Second, to propose new algorithms to tackle some of the obstacles preventing multi-gestures-based myoelectric control systems to be truly usable in practical, real-life applications.

The structure of this thesis is as follows. Chapter 1 gives an overview of machine learning and its related concepts, which are important in understanding this thesis. Chapter 2 presents, to the best of my knowledge, the first real-time control of a robotic arm using sEMG signal based on a deep learning classifier. Chapter 3 proposes a solution, through transfer learning, to reduce the amount of data required to train a deep network so that a new user can perform sEMG-based gesture recognition. A new wireless myoelectric recording armband which outperforms the previously available, most popular sEMG armband is presented in Chapter 4. An analysis of the information learn based on the raw sEMG signal within a deep network, using handcrafted features as landmarks is given in Chapter 5. Chapter 6 presents a study of the four main dynamic factors based on a new experimental protocol which is conducted in virtual reality. Chapter 6 also proposes to apply an updated version of the previously presented transfer learning algorithm over multiple recording session from the same participant to enhance multi-day re-calibration performances. Finally, Chapter 7 proposes to tackle the challenging problem of adapting a classifier to the over-time changes of sEMG signal when the

user does not provide explicit re-calibration sessions. More precisely, Chapter 7 considers the especially hard case where multiple days have elapsed since the last recording session, which proscribes the use of algorithms that rely on small, incremental changes in the signal to adapt.

0.1 Bibliography

- [1] Ravi Balasubramanian and Veronica J Santos. *The human hand as an inspiration for robot hand development*, volume 95. Springer, 2014.
- [2] Elaine A Biddiss and Tom T Chau. Upper limb prosthesis use and abandonment: a survey of the last 25 years. *Prosthetics and orthotics international*, 31(3):236–257, 2007.
- [3] Elaine A Biddiss and Tom T Chau. Upper limb prosthesis use and abandonment: a survey of the last 25 years. *Prosthetics and orthotics international*, 31(3):236–257, 2007.
- [4] Steven W Brose, Douglas J Weber, Ben A Salatin, Garret G Grindle, Hongwu Wang, Juan J Vazquez, and Rory A Cooper. The role of assistive robotics in the lives of persons with disability. *American Journal of Physical Medicine & Rehabilitation*, 89(6):509–521, 2010.
- [5] Claudio Castellini, Angelo Emanuele Fiorilla, and Giulio Sandini. Multi-subject/daily-life activity emg-based control of mechanical hands. *Journal of neuroengineering and rehabilitation*, 6(1):41, 2009.
- [6] Francesca Cordella, Anna Lisa Ciano, Rinaldo Sacchetti, Angelo Davalli, Andrea Giovanni Cutti, Eugenio Guglielmelli, and Loredana Zollo. Literature review on needs of upper limb prosthesis users. *Frontiers in neuroscience*, 10:209, 2016.
- [7] Francesca Cordella, Anna Lisa Ciano, Rinaldo Sacchetti, Angelo Davalli, Andrea Giovanni Cutti, Eugenio Guglielmelli, and Loredana Zollo. Literature review on needs of upper limb prosthesis users. *Frontiers in neuroscience*, 10:209, 2016.
- [8] Grant McGimpsey and Terry C Bradford. Limb prosthetics services and devices. *Bioengineering Institute Center for Neuroprosthetics Worcester Polytechnic Institution*, 2008.
- [9] Mohammadreza Asghari Oskoei and Huosheng Hu. Myoelectric control systems—a survey. *Biomedical signal processing and control*, 2(4):275–294, 2007.
- [10] Erik Scheme and Kevin Englehart. Electromyogram pattern recognition for control of powered upper-limb prostheses: state of the art and challenges for clinical use. *Journal of Rehabilitation Research & Development*, 48(6), 2011.

- [11] David Sierra González and Claudio Castellini. A realistic implementation of ultrasound imaging as a human-machine interface for upper-limb amputees. *Frontiers in neurorobotics*, 7:17, 2013.
- [12] Ann M Simon, Levi J Hargrove, Blair A Lock, and Todd A Kuiken. The target achievement control test: Evaluating real-time myoelectric pattern recognition control of a multifunctional upper-limb prosthesis. *Journal of rehabilitation research and development*, 48(6):619, 2011.
- [13] Michael Wininger, Nam-Hun Kim, and William Craelius. Pressure signature of forearm as predictor of grip force. *Journal of Rehabilitation Research & Development*, 45(6), 2008.

Chapter 1

{Supervised, Transfer, Domain Adaptation, Self-supervised} Learning



Figure 1.1 – Image taken from <https://xkcd.com/1425/>

The chapter aims to present the basic concepts related to machine learning which will be used throughout this thesis. Machine learning algorithms can be thought of as a way to make a system behave in a certain way without giving it explicit instructions to do so. Instead, data related to the task are used to build a model from which predictions/decisions can be derived. Put another way, given a task and no idea how to do it, get data related to said task and

let the computer **try** to figure it out. The actual usefulness of the learned model is highly dependent on the task, the machine learning algorithm and the data. The following sections present different learning paradigms associated with machine learning. The relevance of these paradigm is directly dependent on the type of data and task at hand. Note that this thesis is limited to classification problems. That is, given an input, the model will output a single value chosen within a discrete and finite set.

1.1 Supervised Learning

Supervised learning refers to a setting where the data is comprised of a set of examples containing a certain amount of contextual information (referred to as *features*) and a target (also known as *label*). The *target* represents the desired output. For instance, consider the classification task of predicting if a table (*example*) is from Quebec, Norway or Denmark (*label*) based on the table's color, primary building material and linear length (*features*). In a supervised learning setting one would have a dataset with multiple examples of tables described using the three previously mentioned features and their associated country of origin. The goal would then be to learn a model using this labeled dataset so as to be able to predict the labels of new examples.

More formally, the classification supervised learning setting could be defined as such:

Let $\mathbf{x} \stackrel{\text{def}}{=} (x_1, \dots, x_M)$ be a feature vector with M features, where $x_m \in \mathbb{R}, \forall m \in \{1, \dots, M\}$. Define the dataset $\mathcal{S} \stackrel{\text{def}}{=} (\mathbf{x}_n, y_n)_{n=1}^N \sim \mathcal{A}^N$, where $y_n \in \mathbb{N}$ is the label associated with \mathbf{x}_n , and where each example (\mathbf{x}_n, y_n) is sampled from an unknown distribution \mathcal{A} . \mathcal{S} is also referred to as the training set. In machine learning, the examples are in theory assumed to be independent and identically distributed (i.i.d.). However, within the context of myoelectric control, such an assumption rarely holds in practice as the example (i.e. myoelectric signal) obtained at time $t + 1$ is highly dependent with the example obtained at time t . Note also that as this thesis only considers classification problems, there is a $k \in \mathbb{N}$ such that $\forall(\mathbf{x}, y) \sim \mathcal{A}, y < k$.

Note also that as this thesis only considers classification problems, $(\exists k \in \mathbb{N} | (\forall(\mathbf{x}, y) \sim \mathcal{A} | k > y))$. The machine learning algorithm goal is then to learn a model $\hat{h} : \mathbb{R}^M \rightarrow \mathbb{N}$ so that $\hat{h}(\mathbf{x}) = y, \forall(\mathbf{x}, y) \sim \mathcal{A}$.

1.1.1 Model Training

In practice, learning \hat{h} is often impossible. Instead, the goal becomes to learn an approximation of \hat{h} , $h : \mathbb{R}^M \rightarrow \mathbb{N}$, such that $h(\mathbf{x}) \approx y, \forall(\mathbf{x}, y) \sim \mathcal{A}$. A model h is defined through the values of its *learnable parameters*, where a learnable parameter is defined as able to affect the output of h given a particular input. As such, $h_1 = h_2$ if and only if their learnable parameters are identical. A machine learning algorithm thus affects the parameters of a model to obtain a

"better" model. Consequently, a way to quantitatively qualify a "better model" is needed. This is achieved through a *loss function* (l). Within the classification context, l is defined as:

$$l : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{R}_{\geq 0} \quad (1.1)$$

So that $l(h(\mathbf{x}), y) = 0$ when $h(\mathbf{x}) = y$ and $l(h(\mathbf{x}), y) > 0$ otherwise. As a concrete example, the *zero-one loss* returns 1 when $h(\mathbf{x}) \neq y$ and 0 otherwise.

Oftentimes, the goal of a machine learning algorithm is to find a model h that generalizes well. That is, makes good predictions on examples drawn from \mathcal{A} even if the example is not contained within \mathcal{S} . Given l , which characterizes how good a model is, the goal is to find the model h which has the smallest expected loss on examples from \mathcal{A} . Unfortunately, as \mathcal{A} is unknown, it is not possible to directly minimize the expected loss. Instead, the dataset \mathcal{S} can be used to obtain an approximation of the *true* expected loss. This approximation, named the *empirical risk*, is defined as follows:

$$R_{\mathcal{S}}(h) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{n=1}^N l(h(\mathbf{x}_n), y_n) \quad (1.2)$$

When $N \rightarrow \infty$, $R_{\mathcal{S}}(h)$ converges to the true expected loss. Thus, for N *sufficiently large*, $R_{\mathcal{S}}(h)$ gives a good approximation of the true risk. There is a catch however. In practice, *sufficiently large* depends on the task and on the complexity (i.e. number of adjustable parameters) of model considered [5]. In fact, it is possible for a sufficiently complex h (i.e. large number of adjustable parameters) to learn \mathcal{S} perfectly (or almost perfectly) without h being able to generalize well to new examples generated from \mathcal{A} . This is referred to as overfitting. On the other hand, underfitting arises when a model is too simple, and thus lacks the ability to minimize in a meaningful way $R_{\mathcal{S}}(h)$. Figure 1.2 shows a *regression problem* for different dataset sizes and model complexity using the same machine learning algorithm. A regression problem is similar to a classification problem, except that the output space of the model is \mathbb{R} as opposed to a classification problem where the output space is a finite set of elements in \mathbb{N} . Note that the figure shows a regression problem as, in my opinion, it offers a better visual representation of overfitting/underfitting and the impact of model complexity.

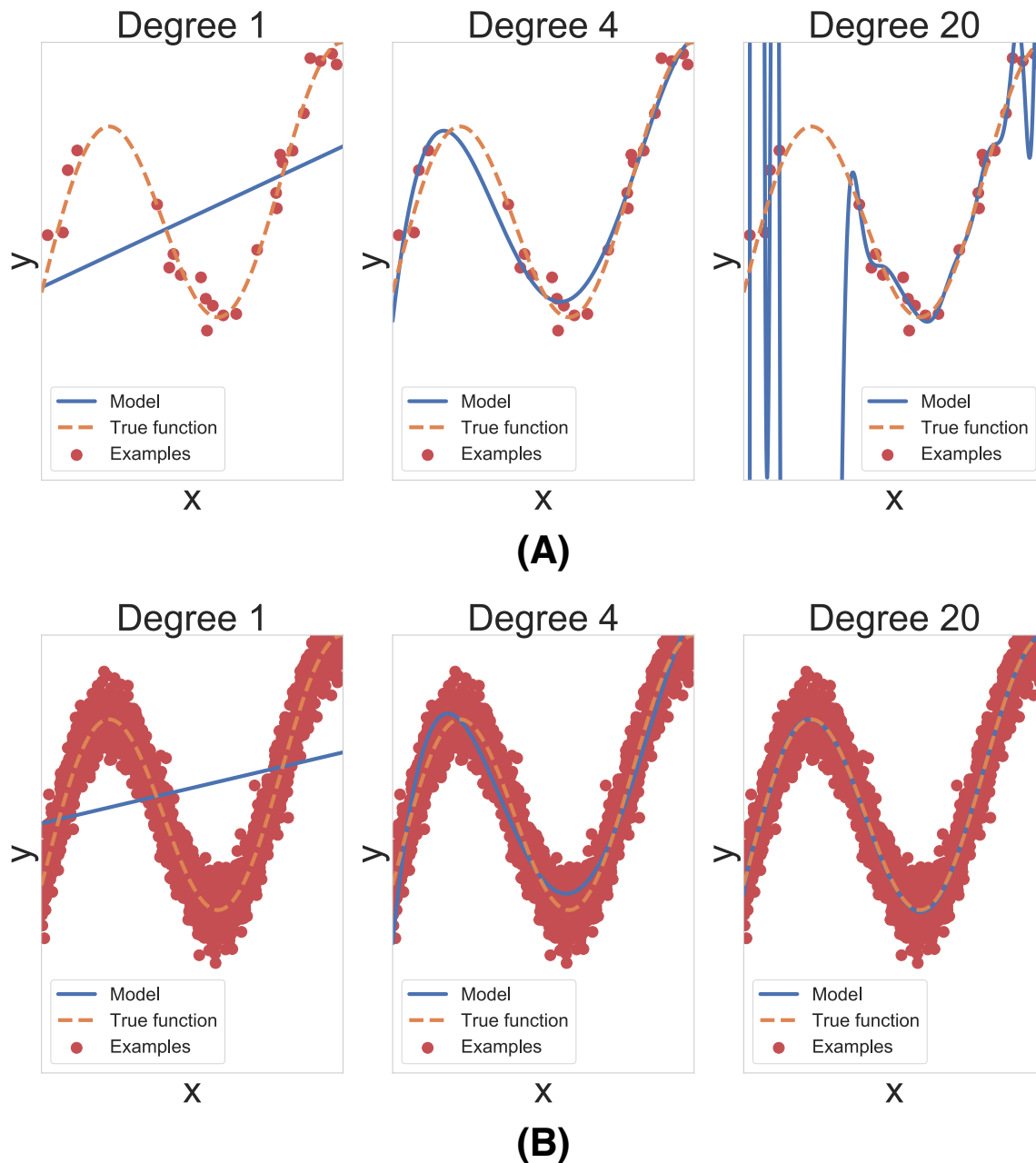


Figure 1.2 – Examples of a regression problem for different dataset sizes and model complexity. (A) Examples of underfitting (left) and overfitting (right). The model in the middle, although not perfect, shows a desirable, well fit behavior given the dataset size. (B) When the size of the dataset increases, the more complex model which previously overfitted, now predicts the true function almost perfectly.

The main takeaway from Figure 1.2 is that in machine learning size matters. That is, bigger datasets allow a machine learning algorithm to learn more complex h which are able to model more complex behaviors. Consequently, the ability of solving a task might be highly dependent in how much labeled data one can obtain.

Regularization

An unlimited amount of training data is, unfortunately, rarely available in practice. Consequently, other complementing methods are needed to address overfitting. One such way is through the use of regularization, which imposes a penalty based on the complexity of the model. In other words, a more complex model needs to outperform a simpler model by enough of a margin to warrant the augmentation in complexity. This behavior is achieved through the use of an additional term which is added to the empirical risk:

$$\frac{1}{n} \sum_{n=1}^N l(h(\mathbf{x}_n), y_n) + \lambda r(h) \quad (1.3)$$

Where r is a real-valued function which increases with the complexity of h and $\lambda \in \mathbb{R}_{\geq 0}$ is a hyperparameter which controls the importance of the regularization. In practice, the definition of r is highly dependent on the type of model considered for a given task.

In the literature, the term *regularization* is often used in a more general manner than defined previously, to refer to concepts or algorithms which directly or indirectly try to reduce overfitting. For example, given a machine learning algorithm that generates a new h at each iteration with increased complexity, setting a maximum allowed number of iterations would also be a form of regularization. This type of regularization is especially prevalent within deep learning (see Section 1.1.2).

Hyperparameters

As can be observed from Figure 1.2-(A), in this specific case, the model with four degrees of freedom models the true function the best even if it makes more mistakes on the training dataset than the model with 20 degrees of freedom. The number of degrees of freedom given to the model is an example of a *hyperparameter*. Hyperparameters are parameters which are fixed before the learning process begins, as their value affects the training process itself. Several types of models have hyperparameters that directly control the complexity of the model (such as specifying the number of degrees of freedom the model had access to in Figure 1.2).

The optimal hyperparameter values are task-specific and can have a large impact on the quality of the learned model. In practice, they are generally evaluated by splitting the training dataset in a training and validation set or by *cross-validation*. The former, which is the simplest, defines \mathcal{T} and \mathcal{V} , such that $\mathcal{T} \cup \mathcal{V} = \mathcal{S}$ and $\mathcal{T} \cap \mathcal{V} = \emptyset$. Different hyperparameter values can then be used to learn h on \mathcal{T} . Then, the model/hyperparameters which obtains the lowest empirical risk on the examples from \mathcal{V} is selected. Cross-validation works similarly, but instead of splitting \mathcal{S} into two non-overlapping sets, it splits it into q non-overlapping sets (called *folds*), where $2 \leq q \leq |\mathcal{S}|$. The machine learning algorithm is then trained q times, with the same hyperparameters, using $q - 1$ folds for training and the remaining one as a validation

set. At each new training, the previous validation fold is added to the training set and a fold which was never used for validation is selected as the new validation set. The score associated with the model/hyperparameters is then defined as the average score across all validation folds. Multiples models/hyperparameters can then be compared by computing their cross-validation’s score.

Specific strategies also need to be used to define the set of hyperparameters to choose from, as in practice there can be an infinite amount of combinations, and unfortunately grants are limited in time. The most popular are *manual search* [4], *grid search* [4], *random search* [4], evolutionary algorithms/swarm intelligence [26] and *Bayesian optimization* [23]. As its name suggests, manual search consists of manually selecting different combinations of hyperparameters. The remaining four all need to have intervals defined for each hyperparameter value (hyperparameters of hyperparameters if you will). With grid search, the user then define for each hyperparameters which values to test. The grid search then tests every hyperparameters combination. The main weakness of this approach is that it can be highly computationally prohibitive to perform depending on the number of hyperparameters. Random search works like grid search, but instead of testing every combination, it selects randomly a set amount of combinations to test. As a rule of thumb, random search should be preferred over grid search [4]. Evolutionary algorithms/swarm intelligence are a family of algorithms for global optimization which are inspired by biological evolution/collective behavior. They can be used as a happy medium between random search and grid search. Finally, Bayesian optimization tries to model the hyperparameters search space. With each new combination of hyperparameters tested, the model can be refined, which in turn can be used to propose a promising new combination of hyperparameters. Bayesian optimization thus tries to simultaneously improve its search space’s model and find the combination of hyperparameters which yield the highest validation score. This type of algorithm is often referred to as *high-cost function optimization*.

1.1.2 Deep Learning

Artificial neural networks [8] is a family of models employed in machine learning. They are characterized by the use of neurons as the building blocks of the model. At its simplest, a neuron is a function $\mathbb{R}^M \rightarrow \mathbb{R}$, which takes as input the features of a given example and return a weighted sum of the input. An additional term, referred to as *bias*, can also be added to the output. Finally, the output often goes through $\phi : \mathbb{R} \rightarrow \mathbb{R}$, a non-linear function. Mathematically, the combination of these three elements can be represented as follows:

$$\phi \left(\sum_{m=1}^M (w_m x_m) + b \right) \quad (1.4)$$

Where w_m are the weights of the neuron and b is the bias, all of which are learnable parameters.

Examples of non-linearity functions commonly used are the Rectified Linear Unit (ReLU)

$$ReLU(z) = \max(0, z) \tag{1.5}$$

and the sigmoid function

$$sigmoid(z) = \frac{1}{1 + e^{-z}} \tag{1.6}$$

The primary role of the non-linear function is to allow the neuron to learn a non-linear model. Additionally, it controls how much information passes through it. For simplicity sake and for the rest of this chapter, a neuron (also called unit) will refer to the combination of the three previously described elements as expressed in Equation 1.4.

A neural network is a model which uses multiple neurons simultaneously, by stacking them into *layers* and creating multiple layers one after the other. An illustration of a *feedforward neural network*, the simplest form of neural network, is given in Figure 1.3. Notice how the neurons are stacked into layers and that the layers between the input and output are referred to as *hidden layers*. The complexity of a neural network model can thus be controlled by changing the number of hidden layers and how many neurons are in each of these hidden layers. Note that, the number of neurons, number of layers and activation functions are all hyperparameters.

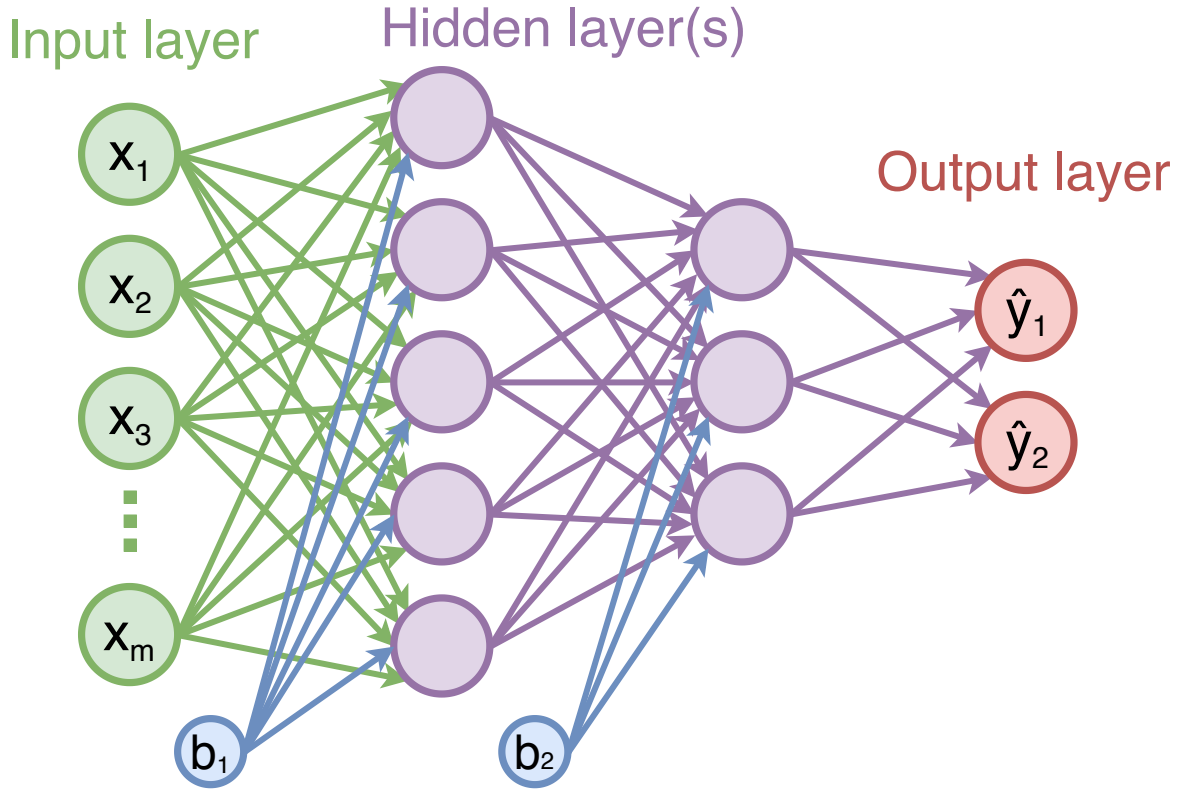


Figure 1.3 – An illustration of a feedforward neural network with two *hidden layers* for a two-class classification problem. In this figure, b_1 and b_2 are the bias associated with the first and second hidden layers respectively. The network will predict class 1 if $\hat{y}_1 > \hat{y}_2$ and class 2 in the second case.

Artificial neural networks are trained using an iterative process. At first, the training dataset is randomly divided into non-overlapping subsets called *batches*. At the first iteration, the first batch is fed to the network. The gradient of the loss function with respect to the network's weights is then computed using the backpropagation algorithm [18]. The weights are then updated to minimize the empirical risk by using a stochastic gradient descent algorithm such as Stochastic Gradient Descent (SGD) [3] or Adam [15]. Then a new batch is fed to the network and the process begins anew. Once all the batches have been used, the training dataset is again randomly divided into new batches. This iteration where new batches are generated is referred to as an epoch, so that the third epoch means that the network has seen all the training examples three times.

Deep learning refers to a type of neural network with "many" hidden layers. They are a multi-level representation learning method (i.e. methods that learn an embedding from an input to facilitate detection or classification), where each level (layer) generates a higher, more abstract representation of the input [17]. Over the years, deep learning algorithms have become increasingly more prominent for their unparalleled ability to automatically learn discriminant

features from large amounts of data. As stated in Section 1.1.1, increasing the size of the dataset allows to learn more complex h which can perform better on more complicated task while reducing the risk of overfitting. However, in practice, learning a model h is associated with a computational cost (both in time and in memory space). In general, this cost rapidly increases as the size of the training dataset grows. So much so that for big datasets, a wide range of machine learning algorithms are not usable in practice. As such, deep learning's ability to scale to large (huge) training datasets sets them apart from a large portion of other machine learning algorithms.

The potential of deep networks to be universal approximators [11] makes regularization a quasi-necessity when using them. In practice, regularization for deep networks, as stated in Section 1.1.1, can go outside of the description provided by Equation 1.3. For example, dropout [24] aims at reducing *co-adaptation* (i.e. different units having highly correlated behavior) of neurons. Dropout achieves this by randomly "dropping" (disabling) a new set of neurons whenever an example/batch passes through the network during training. The percentage of neurons dropped is a hyperparameter. At test time, the complete network is used, but the weights are re-scaled to compensate for this increase in activation activity (as now all the neurons are active at the same time). Early stopping is another prevalent technique to reduce over-fitting. It consists of dividing the training dataset into a training and validation dataset (which are non-overlapping) before learning starts. Then, at each new epoch, the learned model is tested on the validation set. When no improvement is made over a certain amount of epochs on the validation set, the training procedure stops. The network which achieved the best performance on the validation set is then selected. Batch Normalization is another technique which also provides some form of regularization with the aim of maintaining a standard distribution of hidden layer activation values throughout training [13].

Convolutional Networks

Convolutional Networks (ConvNet) are a special type of deep neural networks that were originally conceived for image recognition [16]. Four key concepts describe ConvNet: local connections, shared weights, pooling and having a *deep* architecture [17].

As can be seen from Fig. 1.4, the typical architecture of a ConvNet is structured as two main blocks, each containing a series of stages. The first block is comprised of *convolutional* and *pooling* layers. Each rectangle in Fig 1.4 corresponds to a *Feature Map*. A convolutional layer is comprised of *Units*. Each *Unit* within a convolutional layer is connected to a local patch in the feature map through a set of weights called a *kernel* or *filter bank*. The result of this local weighted sum is then passed through a non-linearity function (most often ReLU). *Units* often overlap each other, but never share the exact same position. All *Units* within a feature map share the same filter banks, on the other hand different feature maps employ different filter banks. The name *convolutional layer* comes from the fact that mathematically, the filtering

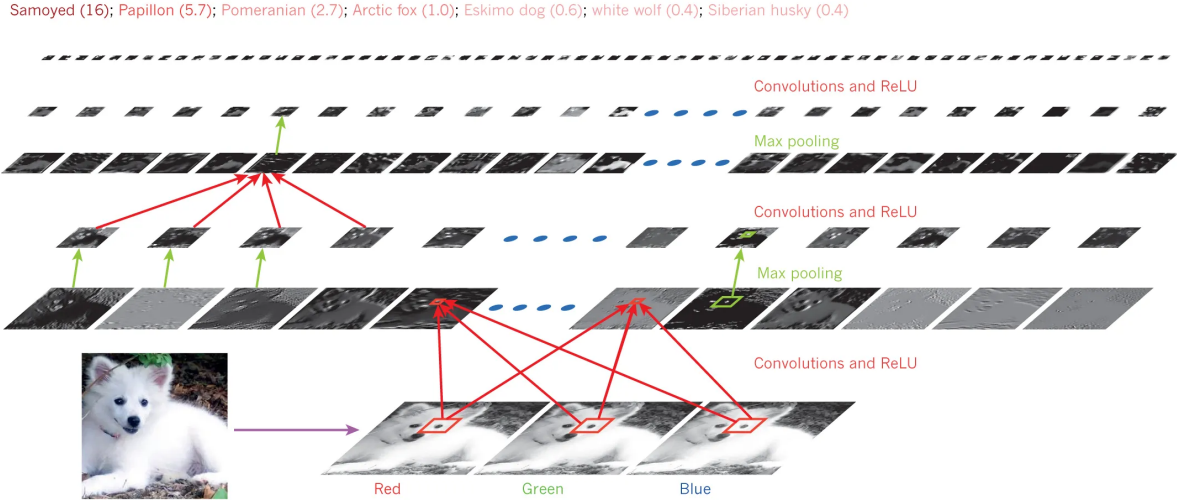


Figure 1.4 – Image taken from [17]. The outputs (not the filters) of the layers of a typical ConvNet applied to a Samoyed dog. Each rectangle is a feature map corresponding to output from one of the learned features.

operation performed by a feature map is a discrete convolution. The reasoning behind these layers revolve around two main points. First, local information in image-like structures tend to be highly correlated which form *motifs* that can be detected easily. Second, local statistics of a signal are often invariant to location. Meaning that if a motif can appear in one part of a signal, it can appear again anywhere and as such as the different *Units* share the same filter bank, they will be able to detect the same motif at different parts of the array.

While the role of convolutional layers is to detect meaningful motifs from the previous layer, the purpose of pooling layers is to merge similar features together. As such, pooling layers extract features that are invariant to small shifts and distortions. The reason for this is that many natural signals are hierarchical by nature. That is, higher-level features are obtained by a combination of lower-level ones. For example, images are a collection of edges, which form motifs, which can then form part of an object or an abstract concept like *hippopotamus*. A hierarchy can also be extracted from speech signals, where sounds aggregate into phonemes, syllables, words and sentences. Pooling layers typically work by computing the maximum of a local patch of *Units* in one feature map. Note that the function of the pooling layer can vary (e.g. mean, median, minimum) and they can also be applied to multiple features maps simultaneously. By having multiple layers that alternate between convolutional and pooling layers (typically, multiple convolutional layers are utilized between each pooling layer) the network is able to exploit this hierarchy with the first layer extracting low-level features while the later layers extract more abstract features. The pooling layers allow the representations of the features extracted to "vary very little when elements in the previous layer vary in position and appearance" [17]. Another important contribution of pooling layers is that they act as a form of dimensionality reduction, reducing the computational load of training and inference of

the network.

The second block of a ConvNet is a conventional deep learning network with fully connected layers. The number of fully connected layers is arbitrary and will vary depending on the task at hand. As such, the *convolutional* stage can be viewed as a form of feature extraction while the second stage is the actual classifier.

Nowadays, ConvNet are prevalent in many fields ranging from Image Recognition [19] to Board Game AI [22, 7] while also including Speech Recognition [1], Biological Signal Recognition [6, 2] and Natural Language Processing [20, 14] amongst many others.

1.2 Transfer Learning

In practice, the availability of a large labeled dataset on which to perform supervised learning is highly dependent on the task at hand. A situation that arises often however is that a large dataset exists for a task similar, but not identical, to the one that need to be solve which only has a small amount of data. Transfer learning algorithms try to leverage this abundance of labeled data for the first task, as to better perform on the second task.

For example, say that someone wants to detect if a picture is that of a lemur or a tarsier. Unfortunately, such a dataset does not exist and consequently, the person starts to painstakingly take pictures of both primates so as to train a model as described in Section 1.1. Fortuitously, a research group has already taken, and labeled, hundred of thousands of pictures of gibbons and orangutans. In supervised learning, this larger dataset would not be useful as it does not represent the desired task, which is where transfer learning comes in. The task of classifying the apes (gibbons vs orangutans) is referred to as the *source task*, while the task of classifying the lemurs vs tarsiers is known as the *target task*.

Transfer learning can formally be defined as follows, define $\mathcal{D} \stackrel{\text{def}}{=} (\mathcal{X}, P(\mathbf{x}))$, a domain, where $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^M$ is a feature space and $P(\mathbf{x})$ is the marginal probability distribution. A task is defined as $\mathcal{T} \stackrel{\text{def}}{=} (\mathcal{Y}, h)$, where \mathcal{Y} is the target space (as a reminder, for classification $\mathcal{Y} \subseteq \mathbb{N}$ and \mathcal{Y} finite). Define a source domain \mathcal{D}_S , its associated learning task \mathcal{T}_S and a target domain \mathcal{D}_T and its associated learning task \mathcal{T}_T . Transfer learning goal is then to improve h_T in \mathcal{D}_T by leveraging \mathcal{D}_S and \mathcal{T}_S when $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$.

Within deep learning, the technique known as *fine-tuning* [17] is arguably the most popular and used transfer learning algorithm due to its simplicity and effectiveness. To continue the previous example, fine-tuning consists in training a network to distinguish between the apes (source dataset). Then starting with the model learned on the source, continue training the network (generally allowing smaller changes in the weights between each iteration) but only on the target. The idea is that the initial training (*pre-training*) will allow the network to find a meaningful representation which simply need to be "fine-tuned" to perform well on the

target. In practice, the need for transfer learning arises often as labeling huge amounts of data is generally costly (both in term of money and time). As such, the few huge datasets already available (e.g. ImageNet [9]) are routinely used as a starting point (read pre-training) when working on a new problem or with new data.

1.3 Unsupervised Domain Adaptation

In domain adaptation, the task between the source and target is the same, but $\mathcal{D}_S \neq \mathcal{D}_T$. Additionally in the unsupervised case (the one considered in this thesis), the target dataset is devoid of labels. The goal is then to learn h_T having access to a labeled dataset from the source and unlabeled examples from the target.

In general, within deep learning, unsupervised domain adaptation algorithms tend to define auxiliary tasks [10, 25] or loss functions [21] to try to "guide" the network to learn a useful representation for classification on the new domain. These auxiliary tasks are designed so that they must be solved by including data from both domains and that true labels can be generated with the already available information. For example, DANN [10] proposes to create the auxiliary task of being "bad" at distinguishing between the source and target. The labels can thus be easily generated by assigning 1 to examples from the source and 2 otherwise. This idea is based on the hypothesis that for effective domain adaptation to be obtained, the data representation between the two domains should be indistinguishable. An example of an auxiliary loss function designed for unsupervised domain adaptation is VADA [21]. Building on DANN, VADA proposes to also add a loss which penalizes the network when its decision boundary goes through high-density clusters of the target data. The idea stems from the cluster assumption which states that examples within the same cluster probably belong to the same class.

1.4 Self-supervised Learning

In the previously described unsupervised domain adaptation algorithm, a classifier is trained on the source and leverages auxiliary task(s) to learn a discriminative parametrization on the target. These auxiliary tasks are designed to generate a feature representation that will, hopefully, be meaningful for data classification tasks, without having access to any actual targets' label. In self-supervised feature learning, the data itself generates labels to perform tasks that will, again hopefully, enhance the performance of the classifier on the "real" task.

Self-supervised learning does not necessarily consider a source and target domain and is often applied on unlabeled data from a single domain. In general, the main idea is to use different data modalities from which a relevant label can be automatically generated. For example, consider a robot which has a camera (long range sensor) and a proximity sensor (low range

sensor) and the task of learning to detect obstacle from the camera without any labeled dataset. Self-supervised learning would generate labels associated with the camera’s image by trying to link an image seen at time t with the output of the proximity sensor at time t' .

Within this thesis, self-learning is considered only within the unsupervised domain adaptation setting. More specifically, as this thesis focuses on myoelectric control, the delay between each prediction is one of the most important factor to consider. Hence, the amount of temporal data which can be used within any single example is limited during real-time control. In practice, to allow intuitive control from EMG-based real-time gesture recognition, 300ms of data per example seems to be the upper-limit [12]. This limitation negatively affects the ability of a model to generate accurate predictions. Additionally, as stated in the introduction, sEMG signals are non-stationary and multiple factors (e.g. limb position, gesture intensity) affect the signal in addition to the performed gestures. In other words, even given a labeled dataset to train on, the model’s performance will degrade over time. The idea is then to collect unlabeled data during gesture recognition tasks and use longer time intervals to re-label the predictions of the classifier, as this extra temporal context should allow to improve gesture recognition in an offline setting.

1.5 Bibliography

- [1] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, 22(10):1533–1545, 2014.
- [2] U Rajendra Acharya, Hamido Fujita, Shu Lih Oh, Yuki Hagiwara, Jen Hong Tan, and Muhammad Adam. Application of deep convolutional neural network for automated detection of myocardial infarction using ecg signals. *Information Sciences*, 415:190–198, 2017.
- [3] Shun-ichi Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4-5):185–196, 1993.
- [4] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.
- [5] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.
- [6] Hubert Cecotti and Axel Graser. Convolutional neural networks for p300 detection with application to brain-computer interfaces. *IEEE transactions on pattern analysis and machine intelligence*, 33(3):433–445, 2011.

- [7] Kumar Chellapilla and David B. Fogel. Evolving an expert checkers playing program without using human expertise. *IEEE Transactions on Evolutionary Computation*, 5(4):422–428, 2001.
- [8] A Cochocki and Rolf Unbehauen. *Neural networks for optimization and signal processing*. John Wiley & Sons, Inc., 1993.
- [9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [10] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.
- [11] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.
- [12] Bernard Hudgins, Philip Parker, and Robert N Scott. A new strategy for multifunction myoelectric control. *IEEE Transactions on Biomedical Engineering*, 40(1):82–94, 1993.
- [13] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
- [14] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.
- [15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [16] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.
- [17] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
- [18] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.
- [19] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

- [20] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. Learning semantic representations using convolutional neural networks for web search. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 373–374. ACM, 2014.
- [21] Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. *arXiv preprint arXiv:1802.08735*, 2018.
- [22] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.
- [23] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.
- [24] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.
- [25] Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A Efros. Unsupervised domain adaptation through self-supervision. *arXiv preprint arXiv:1909.11825*, 2019.
- [26] Steven R Young, Derek C Rose, Thomas P Karnowski, Seung-Hwan Lim, and Robert M Patton. Optimizing deep learning hyper-parameters through an evolutionary algorithm. In *Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments*, page 4. ACM, 2015.

Chapter 2

A convolutional neural network for robotic arm guidance using sEMG based frequency-features

2.1 Reference

Côté-Allard, Ulysse, François Nougrou, Cheikh Latyr Fall, Philippe Giguère, Clément Gosselin, François Laviolette, and Benoit Gosselin. **A convolutional neural network for robotic arm guidance using sEMG based frequency-features**. In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2464-2470. IEEE, 2016.

2.2 Context

This paper presents one of the first application of deep learning to the field of sEMG-based gesture recognition alongside [26, 2]. More importantly, and to the best of my knowledge, this article was the first proof-of-concept that a convolutional network could be employed to create a real-time myoelectric control system.

From a personal perspective, this article was the first one that I produced as a direct result of my master/doctorate research and my first experience writing a research article as a first author. Importantly this was a formative experience in regards to working with a multi-disciplinary team of researchers, something that I was fortunate enough to be able to do throughout my doctorate. I also had the opportunity to present this paper at the International Conference on Intelligent Robots and Systems, one of the two biggest robotic conference (alongside the International Conference on Robotics and Automation).

2.3 Résumé

Récemment, la robotique a été vue comme une solution clef afin d'améliorer la qualité de vie des personnes amputées. Pour créer des dispositifs prothétiques robotisés plus intelligents destinés à être utilisés dans un contexte quotidien, il faut que ceux-ci puissent interfacer de manière transparente et peu coûteuse avec l'utilisateur final.

Dans cet article, nous cherchons à guider un dispositif robotique en détectant les gestes de la main en mesurant l'activité électrique des muscles capturés par l'électromyographie de surface. De tels classificateurs de gestes sont cependant difficiles à concevoir, car ils doivent être extrêmement robustes à la dérive du signal, à la fatigue musculaire et au petit déplacement des électrodes sans nécessiter un réentraînement constant. Malgré des décennies de recherche dans ce domaine, les systèmes de classification multigestes basés sur l'électromyographie ne sont toujours pas utilisés, car ces systèmes ne parviennent souvent pas à résoudre tous ces problèmes simultanément.

Nous proposons de tenter de résoudre ces problèmes en utilisant des réseaux convolutionnels. Dans un premier temps, nous démontrons leur viabilité pour la classification de gestes de la main chez des personnes non-amputées à l'aide d'un bracelet myoélectrique commercial à 8 canaux utilisant des électrodes sèches et possédant un faible taux d'échantillonnage (200Hz). Dans ce travail, nous considérons la classification de sept gestes de la main. Nous montrons par un cas d'utilisation chez un participant non amputé, que notre approche utilisant l'apprentissage machine permet de prédire le bon geste parmi sept 97.9% du temps sur une période de six jours consécutifs sans réentraînement. De plus, en combinant le classificateur proposé dans cet article avec les données d'orientation contenues dans le bracelet, nous créons un système permettant de guider un bras robotique possédant 6 degrés de liberté avec la même vitesse et la même précision qu'en utilisant une manette (la façon normale de contrôler ce bras). Ce dernier résultat est obtenu en testant le système sur 18 participants non amputés.

2.4 Abstract

Recently, robotics has been seen as a key solution to improve the quality of life of amputees. In order to create smarter robotic prosthetic devices to be used in an everyday context, one must be able to interface them seamlessly with the end-user in an inexpensive, yet reliable way. In this paper, we are looking at guiding a robotic device by detecting gestures through measurement of the electrical activity of muscles captured by surface electromyography (sEMG). Reliable sEMG-based gesture classifiers for end-users are challenging to design, as they must be extremely robust to signal drift, muscle fatigue and small electrode displacement without the need for constant recalibration. In spite of extensive research, sophisticated sEMG classifiers for prostheses guidance are not yet widely used, as systems often fail to solve these issues simultaneously. We propose to address these problems by employing Convolutional Neural

Networks. Specifically as a first step, we demonstrate their viability to the problem of gesture recognition for a low-cost, low-sampling rate (200Hz) consumer-grade, 8-channel, dry electrodes sEMG device called *Myo armband* (Thalmic Labs) on able-bodied subjects. To this effect, we assessed the robustness of this machine learning oriented approach by classifying a combination of 7 hand/wrist gestures with an accuracy of $\sim 97.9\%$ in real-time, over a period of 6 consecutive days with no recalibration. In addition, we used the classifier (in conjunction with orientation data) to guide a 6DoF robotic arm, using the armband with the same speed and precision as with a joystick. We also show that the classifier is able to generalize to different users by testing it on 18 participants.

2.5 Introduction

The commoditization of robots and sensors creates new opportunities to integrate robotics into day-to-day life. In particular, some of these developments aim at easing or aiding in common everyday tasks. For those who depend on prostheses and assistive robots, such developments can significantly improve their quality of life [9], [6]. In order to leverage the full potential of robotic devices in this context, it is essential to develop novel and intuitive ways to control them. An ideal interface would also be as intuitive and inconspicuous as possible, to provide a seamless experience to non-expert users.

One possible way to achieve such a natural interface is through Surface electromyography (sEMG). It is a non-invasive technique, extensively adopted in clinical and research works related to muscular activities. sEMG signals are non-stationary, and represent the sum of subcutaneous motor unit action potentials generated during a muscular contraction [25]. The use of sEMG signals, combined with pattern recognition systems, has been proposed in the literature as an effective avenue to provide a more intuitive control of devices such as prosthesis or assistive robots [25], [12]. Studies in this topic mainly employ several sEMG electrodes placed on specified muscles to perform forearm pattern recognition. Furthermore, when using gel-based electrodes, the user’s skin has to be shaved and washed to obtain optimal contact between the electrodes and the skin. This severely limits the practicability of such systems by making the preparation step a long, delicate and complex process.

In order to be able to use sEMG signals for robotic guidance, pattern recognition must be performed to identify a user’s gesture. The two main components of pattern recognition are feature extraction and classification. For sEMG, features extracted from the time-domain have been extensively studied [10], [28] (*e.g.* Mean Absolute Value, Zero Crossing, Willison amplitude and Integrated EMG). However, as mentioned in [32], if temporal features are fast and easy to implement, they are sensitive to frequent amplitude fluctuations compared to features from the frequency-domain (*e.g.* Fourier Transform, Median Frequency, Mean Frequency [17], [28]). Features in the time-frequency domain (*e.g.* spectrograms, wavelet

transform, wavelet packet transform) provide a richer way to extract pattern information [11]. For classification, many methods (linear, non-linear, supervised or unsupervised) have been employed to estimate unknown patterns from a set of features [25], [12]. The most common methods are the linear discriminant analysis (LDA) and artificial neural networks (ANN). Even if each classifier presents its own advantages, they remain too sensitive to electrode displacement and positioning when used with sEMG electrodes placed on specific muscles [5].

In the case of a prosthesis, the guidance system should ideally be small, inexpensive, lightweight, require minimal preparation and be robust to a small displacement of the electrodes while still achieving excellent classification performance. Dry electrodes should also be preferred over gel-based ones, as they are inherently more convenient to use. However, they are less accurate and less robust to motion artifact, compared to gel-based ones [31].

The work presented in this paper addresses these severe limitations while still achieving state of the art results. Our approach is based on employing convolutional neural networks (CNN) to perform the classification of spectrograms of the sEMG signals, in order to identify a number of gestures. One of the contributions of this work is thus the use of a CNN to classify the very noisy sEMG data [13]. Indeed although they have been used before in speech recognition [30] and EEG classification [7], we believe this is the first time that they are used to classify sEMG data for gesture recognition. Importantly, the use of CNNs shifts the focus from *feature engineering* to *feature learning*. Indeed, because of the nature of CNNs, features are automatically learned via the convolutional layers. Those features are then transferred to fully connected layers that associate sEMG signals to specified gestures.

Although deep networks are often seen as computationally expensive, recent development in hardware for deep learning makes complex algorithms implementation in embedded systems a reality [29]. Additionally, dedicated deep learning materials such as Eyeriss [8] are able to run CNNs with up to 60 millions parameters at 35 fps using only 278 *mW*. Furthermore, those very-low power systems only need to handle the inference step since training can be done on a desktop and the weights of the parameters simply sent via bluetooth once the optimization is completed. Finally, using network pruning, one can achieve a compression rate over 10x [16] which significantly reduces both inference time and memory space requirement of the hardware.

Once the classifier is able to reliably identify these gestures, one can easily create a guiding system by associating a particular robot motion primitive to a gesture. To obtain the sEMG data in the least possible intrusive way, a Myo armband by Thalmic Labs¹ can be used. This consumer-grade device includes a 9-degree-of-freedom (DoF) Inertial Measurement Unit (IMU) and 8 dry surface electromyogram sensors. The proposed system use both the IMU and sEMG data for the guidance of the robotic arm.

The rest of the paper goes as follow. First, Section 2.6 provides an overview of our guidance

¹<https://www.myo.com/>

system. The architecture of our CNN classifier is then described in Section 2.7, along with the spectrogram features that are employed. The ability of the classifier to generalize well to different users is showcased in Section 2.8.1. We then demonstrate the precision and the robustness of this guidance system by performing precise and complex tasks in a speed test in Section 2.8.2. Finally, Section 2.8.3 establishes the long term stability of the classifier and its robustness to muscle fatigue by testing it over two periods of six consecutive days on a healthy subject. For each period, the emplacement of the electrodes on the forearm were not marked and no recalibration was performed after the initial training. This naturally led to small displacements of the electrodes between each recording session, which the classifier was not specifically trained to resist but was nonetheless robust to.

2.6 Proposed guidance system overview

The description of the robotic arm, the Myo and the gestures used in this work are given in this Section. The classifier itself is detailed in Section 2.7.

2.6.1 Myo Armband

In 2014, the Myo armband was released at a purchase cost of 200 \$. As stated previously, it contains a 9DoF IMU and 8 dry surface electromyogram sensors. One of the main advantages of the Myo is that it can simply be slipped on the arm to read sEMG signals with no preparation. The sEMG data from the Myo can be visualized in [33]. The Myo armband provides a sEMG sampling frequency (f_s) of 200 Hz per channel (a f_s of at least 1 kHz is normally preferred to address sEMG signals lying within 5-450 Hz [24]). Electrode placement was dependent on the size of the subject’s forearm due to the minimum circumference of the Myo (19.05 cm). Additionally, no shaving of hair or skin-cleaning were performed for this study as these were judged too constraining for a potential end-user. Note that this generates extra noise that can nevertheless be handled by our machine learning approach.

2.6.2 JACO guidance using a Myo armband

The JACO arm by Kinova ² is a 6DoF robotic arm that is usually manually operated using a 7-button joystick. The coordinate system used by the robot is Euler X-Y-Z. The user can navigate through 3 different modes to access the full motion of the robotic arm: 1) translate (Move the arm along X-Y-Z axes), 2) rotate (Rotation of the robotic hand around the X-Y-Z axes) and 3) grip (open-closing the hand).

The different gestures employed to generate sEMG patterns are explained in Section 2.6.3 and replace the rotation around the X-Y axis and the grip mode. The rotation around the Z axis

²<http://kinovarobotics.com/>

(corresponding to the roll motion on JACO's hand) is not considered in this work since it was deemed not useful in the experiments described in Section 2.8.

The translate mode is mapped to the orientation of the armband which is obtained using the IMU included inside the Myo. The yaw corresponding to the X-axis, the pitch to the Y-axis and the roll to the Z-axis. Furthermore, for the translate mode, the mapping is proportional to the angle between the orientation of the forearm and a horizontal reference orientation corresponding to a 90 degrees flexion of the elbow. Thus, the larger the angle between the neutral configuration of the forearm and the current orientation of the Myo, the faster JACO will move.

JACO also includes a spasm filter which is used solely for the third task in Section 2.8.1 at the lowest possible setting. The spasm filter limits the acceleration of JACO, while not affecting its deceleration speed. It is important to note that opening and closing the hand is not affected by this filter.

When guiding JACO with the Myo, a system of movement priority is established. Translation along the X and Y axes have the highest priority. If the arm of the user is within the resting position range with respect to the pitch and yaw, then the user can perform a translation of the robotic arm along the Z-axis by rotating his wrist (pronation and supination). Finally, if the roll is also within its neutral range, the classifier output is used to guide the robot. This priority list is necessary because inexperienced users, have a tendency to rotate their wrist involuntarily when moving their arm. Furthermore the classifier has a lower priority since the muscle activity generated by the rotation of the wrist, is not an activity the classifier is trained to recognize and therefore the output during this movement is not reliable.

2.6.3 Description of the gestures

Since the final purpose of this experiment is to guide the robotic arm with the Myo armband, seven different hand/wrist gestures are required. The classes are: neutral, hand open, hand close, wrist flexion, wrist extension, radial deviation and ulnar deviation. Fig. 2.1 shows the different gestures as well as the Myo and JACO. These gestures are chosen because they can intuitively be mapped to the rotate and grip mode of the robotic arm. The gestures and their corresponding action of the robotic arm, are as follows: opening and closing of the hand to represent the opening and closing of the robotic hand, the wrist flexion and extension correspond to moving the joystick to the left and right in the rotate mode. Finally, the radial and ulnar deviation emulate pushing the joystick to the front and the back respectively.

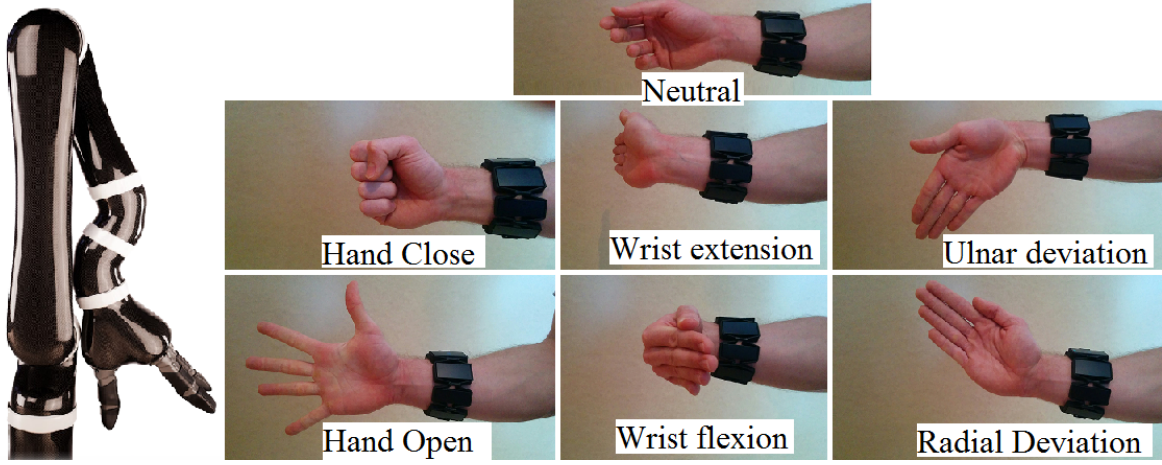


Figure 2.1 – The 7 gestures considered in this work. The Myo armband (right) is connected through bluetooth to the laptop. The computer, after data analysis, transfer the command to JACO (left) in real-time via a USB connection.

2.7 Classifier overview

Processing of the sEMG data signal was necessary to be able to recognize the different hand/wrist movements of the user. In this section, the different steps performed during an online classification are exposed. We first describe how the data is separated into time windows, then pre-processed using consecutive fast Fourier transform (FFT) for forming one spectrogram per channel, and finally fed to a CNN to predict the current hand/wrist movement.

2.7.1 Time-window

As stated previously, the Myo armband includes 8 sEMG channels, each sampled at 200 *Hz*. For closed loop and online operation, latency is an important parameter to consider. In [19], it was first recommended that the time-window between two predictions be equal to or less than 300 *ms*, while in [14] it was found that ideally, the latency should be between 100 and 125 *ms*. However, in [27] it was reported that the performance of the classifier should take priority over speed. In our system, we opted for a maximal latency of 300 *ms*, in order to accumulate a sufficient number of samples with the low f_s of the Myo, and thus increase classification performance. Considering that the time to process and classify one gesture's sEMG-pattern window of ~ 300 *ms* takes on average ~ 15 *ms* on our hardware (laptop with an NVIDIA GeForce GT 555M), we used windows of 285 *ms*. This corresponds to 57 data points per channel per example. Overall, this kept the data capture and processing time below our target latency of 300 *ms*.

2.7.2 Preprocessing

Spectrograms are calculated for each sEMG channel of 57 samples using windows of 30 points for FFT and an overlap of 21. Based on these parameters, 4 FFT will be contained in the spectrograms. This results in a spectrogram matrix of 16 by 4, with a frequency step of 6.67 *Hz*. Note that a Hamming window is used to avoid frequency leakage. The spectrograms are calculated using Scipy implementation in Python [21]. The first row of the spectrogram array is removed because it is out of the useful frequency range of the sEMG signal (Section 2.6.1). The final spectrograms have a frequency range of 6.67 to 100 *Hz*.

2.7.3 Classification Algorithm

We tried most of the state of the art machine learning algorithms (e.g. Support vector machine, Adaboost, Random Forest, Deep neural network). Considering that a CNN achieved by far the best results, it was selected as the classifier for our system. Its architecture is described below.

Description

The classification algorithm consists of a two-staged CNN, implemented using the python library Theano [4], [3]. This library allows the CNN to run on a GPU, thereby accelerating the training and prediction. The first stage is used to differentiate between the neutral class and the others. If the former is not detected, the algorithm proceeds to Stage 2, which differentiates between the remaining six gestures. Justification for this separation is at the bottom of this *Description*. The architecture of the CNN remains the same in both stages, except for the output layer which contains two and six neurons respectively. The architecture of stage 2 (containing ~ 3.6 millions parameters) was selected as usual in deep learning by trial and error using previously published architecture as inspiration (mainly [23]) and is presented in Fig. 2.2.

We use ADADELTA [35] for the optimization of the CNN weights. The hyperbolic tangent (tanh) is used as the non-linear activation function. The rectified linear Unit (ReLU) has been considered mainly for speed reasons [22]. However ReLU was not retained because even though each iteration experienced a slight speed boost as expected, the CNN tended to converge faster with tanh and the accuracy in validation was similar between the two. The sigmoid function was also considered, but performed poorly for this task compared to both tanh and ReLU. Additionally, the proposed system uses the dropout approach [18] to prevent overfitting. For the convolutions layers, the dropout is set at 25%. For the two fully connected layers before merging it is at 50%. Finally, the dropout of the last two layers is set at 75%.

In the implementation of Stage 2, we first go through a rescaling step. First, considering the eight spectrograms at time T as a 3D matrix of shape 8x15x4, we reshape it into a 480x1 vector x_T . Then rescaling is performed with $\frac{x_T^i}{\text{norm}(x_T)}$, where x_T^i is the value in position i of the vector x_T and norm is the L2 norm. Performing the rescaling on the concatenated

spectrograms ensures that their relative intensities are taken into account. After rescaling, the eight spectrograms are reshaped back into their 15x4 format, feeding the two-stage CNN with 2D images. This reshaping preserves important correlations between channels and within spectrograms. The L2 norm achieves a trade-off between a quasi-constant power spectrum (within a factor of L2) and putting more weight on frequencies with higher power spectrum, an approach similar to extracting peaks in a power spectrum for time-series classification [20]. This last effect is less noticeable when using the L1 norm instead of the L2 norm. Furthermore, we observed faster convergence rate of the CNN using the L2 norm over the L1 norm. Note that when using a single-stage approach with the rescaling, the performances of the classifier degrades significantly. We attribute this to the fact that the rescaling step tends to normalize the energy level of incoming spectrograms, before they are fed to the CNN. Since discriminating between a neutral gestures (homogeneously low energy) and the other gestures (high muscle activity) probably relies on this energy level, this would explain the poor performance of this rescaled, single-stage CNN approach. The two-staged CNN approach that we have adopted sidesteps this issue completely, by performing rescaling *only* after a gesture has been deemed as non-neutral.

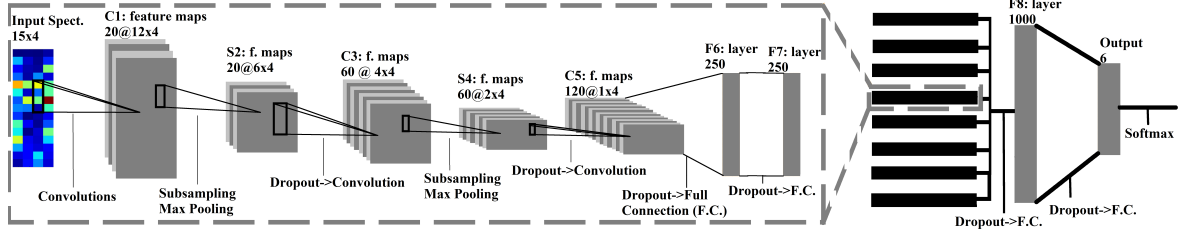


Figure 2.2 – Stage-2 architecture. Each channel is considered independently at first, going through parallel and identical convolutional networks for feature extraction. These networks are then merged together through a fully connected layer (F8). The output of F8 is then use to perform inference.

Training and validation

In the training phase, the process of collecting labeled data for the CNN required the user to hold each gestures for 5 s. These labeled intervals are then divided into time-window of 285 ms, as described in Section 2.7.1. To see more accurately variation within the same class, each window overlapped the previous one by 265 ms. This process is repeated three times, yielding 15 s of data per class. We take three trials of 5 s per gesture as opposed to one trial of 15 s to get more variation on the same gesture. Indeed, a user cannot perform exactly the same motion with the same strength when asked to do the same gesture twice. This also follows the recommendation made by [34] of varying strength recording for the same gesture. Validation data is created in a fourth independent run (5 s per gestures) in an identical manner.

2.8 Experimental Results

Experiments were conducted to evaluate the performance of the CNN classifier (described in Section 2.7) on three main aspects that correspond to Section 2.8.1, 2.8.2 and 2.8.3 respectively. The experiments of Section 2.8.1 assessed the ability of the classifier to generalize to different individuals. Then, experiments in Section 2.8.2 were used to compare the task completion time between an expert in guiding the robot with the joystick against one well-versed with our Myo interface. Finally, the classifier’s robustness to sEMG signal drift [1], small electrode displacement and short term muscle fatigue is evaluated in Section 2.8.3. All results reported here were based on the zero-one loss accuracy. Meaning that a classification is considered successful only if the predicted gesture is exactly the one being made.

2.8.1 Generalization Experiment

We tested our system on 18 (11 men and 7 women) healthy subjects aged between 23 and 29 years old. The Myo armband was placed at a single but different location on the forearm, depending on the user. Indeed, since the armband minimum circumference is 19.05 cm and the test subjects had a forearm circumference between 15.5 and 24.0 *cm* (measured 5 *cm* above the wrist), it would have been difficult to obtain the same forearm sensor placement for each subject. Therefore, we set the armband at the minimum circumference and simply slid it up until the forearm’s circumference matched the armband’s one. The placement of the armband was thus directly dependent on the subject’s forearm circumference. Consequently, it is important that the performance of our approach be as independent as possible from the placement of the electrodes.

Training of the CNN was realized as described in Section 2.7.3. The average accuracy in validation for the participants was 100% for the first stage and 97.71% for the second stage. The participants were then asked to perform three tasks: (1) gesture accuracy, (2) cube holding and (3) picking an object to place it in a specified zone. Details on these tasks are presented below.

Gesture accuracy test (Task 1)

To evaluate the accuracy of the classifier during short-term muscle fatigue, the participants were asked to hold one of seven gestures, chosen randomly, for 10 *s*. No rest was given between each gesture. The test length was 5 *min*, yielding 30 trial-gestures. The participants were noted on the amount of trial-gestures that they succeeded and given a score out of 30. A gesture was considered a success if no more than two false consecutive or no more than four non-consecutive miss-classifications occurred during a 10 *s* period. Transitioning between gestures was not considered in this task. The average success rate was 93.14% over all participants.

Cube holding (Task 2)

For the guidance of a robotic arm, the negative impact of miss-classification highly depends on the nature of the error. Indeed, if the user wants to close his hand and the classifier interpreted it as a neutral state, it is easy for the user to perform the gesture again. On the contrary, if the user wants to be in a resting position while the robotic arm is holding a glass of water and the resulting classification is hand open, serious consequences can be envisioned. The second task thus tested the capability of a user to hold an object in the robotic hand while making different gestures. The setup of this task was identical to Section 2.8.1 except that the gesture open hand was not requested and that the duration was 120 s. All participant successfully held the cube for 120 s during this task.

Picking and placing cube (Task 3)

In the final task, the participants were asked to pick a cube with the robotic arm and put it in a specified location. The participants were first asked to perform the task with the normal guidance system for the robotic arm (joystick). They were timed for both picking and dropping the cube at the specified place. They then performed the same task with the Myo as the guidance system. For both tasks, they had 10 *min* of training prior to performing their task. It is important to note that the results that follow only aim at providing an order of magnitude for the time required to complete the task. Indeed, since the participants always started with the joystick, this gives an unfair advantage to the Myo armband system. The task is thus not suited to truly compare them in terms of speed, but simply to show that the classifier is sufficiently accurate to perform similarly to the joystick. The average time to perform the task with the joystick was: 1 *min* 45 s and with the Myo armband: 1 *min* 33 s.

Results and discussions of the three tasks

None of the participants had experience with sEMG-based classifiers or guidance of a robotic arm and had no known physical disabilities. Table 2.1 presents the general information on the participants as well as the results for the first two tasks. The circumference of the forearm was measured 5 *cm* above the wrist. The accuracy reported in Table 2.1 is the validation accuracy (Section 2.7.3) for the two stages of the classifier. We can immediately see from the validation accuracy that the classifier is always able to learn to distinguish between the 7 gestures. Participants generally achieved high performance on Task 1 and all were perfect on Task 2.

Fig. 2.3 presents the time taken by the participant to complete Task 3. It is important to note that one of the participants did not complete the Myo portion of task 3. The reason is that the classifier performed too poorly for the precise manipulation required from Task 3. The participant data from the other tasks are included in the statistics (76.66% for task 1 and 120 s for Task 2).

Table 2.1 – Participants general information and first two tasks results

	Woman		Man	
	Average	STD	Average	STD
Age (Year)	24.29	1.67	24.55	2.27
Forearm circumference (<i>cm</i>)	17.0	1.36	19.73	2.85
stage 1 validation accuracy	100.00 %	0.00 %	100.00 %	0.00 %
stage 2 validation accuracy	97.61 %	1.14 %	97.76 %	2.93 %
Task 1	93.81 %	5.17 %	92.72 %	6.49 %
Task 2 (<i>s</i>)	120	0.0	120	0.0

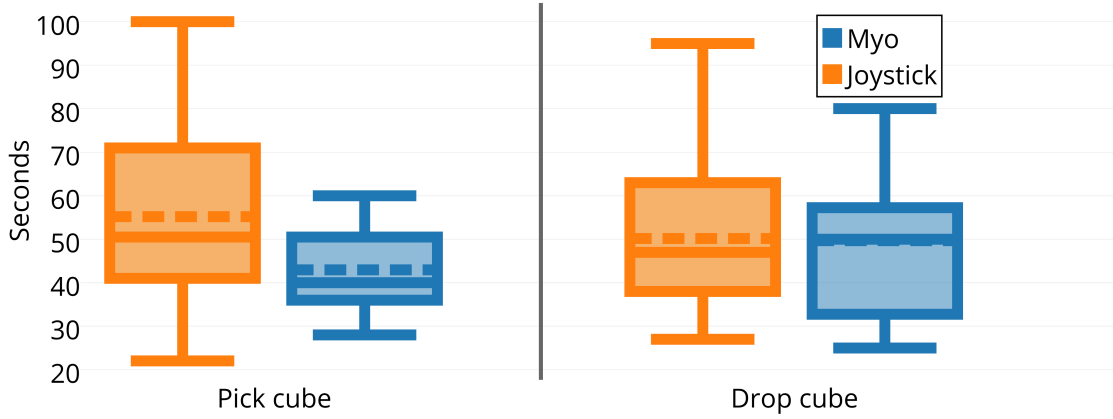


Figure 2.3 – Box plot of the time taken to complete task 3 (picking and placing the cube). The dotted line represent the mean of the distribution.

For inexperienced users, Fig. 2.3 shows that the time needed to complete task 3 with the joystick and the Myo are on the same order of magnitude. We cannot conclude that our guidance scheme is more intuitive because the time difference is not statistically significant (ANOVA $p\text{-value} > 0.05$). We can however conclude that our system is robust enough to reliably guide the robotic arm in precise tasks.

2.8.2 Speed test for a complex task

We specifically designed a speed challenge to evaluate the usefulness of the classifier in an online situation and to provide a time comparison between the joystick and the Myo. The task consisted in picking and placing three cubes consecutively, in a similar manner as described in Section 2.8.1. The joystick times were achieved by an expert in guidance of JACO. The expert

had no physical disability. Similarly, the Myo times were achieved by an experienced user with the guiding system with no physical disability. The challenge was performed three times for each guiding scheme. The reported results are the average over three runs. Fig. 2.4 compares each sub-task of the challenge (picking and dropping the three cubes). A video accompanying this article shows the task being performed with the Myo. It should be noted that the first cube grabbed and then dropped correspond exactly to task 3 as described in Section 2.8.1.

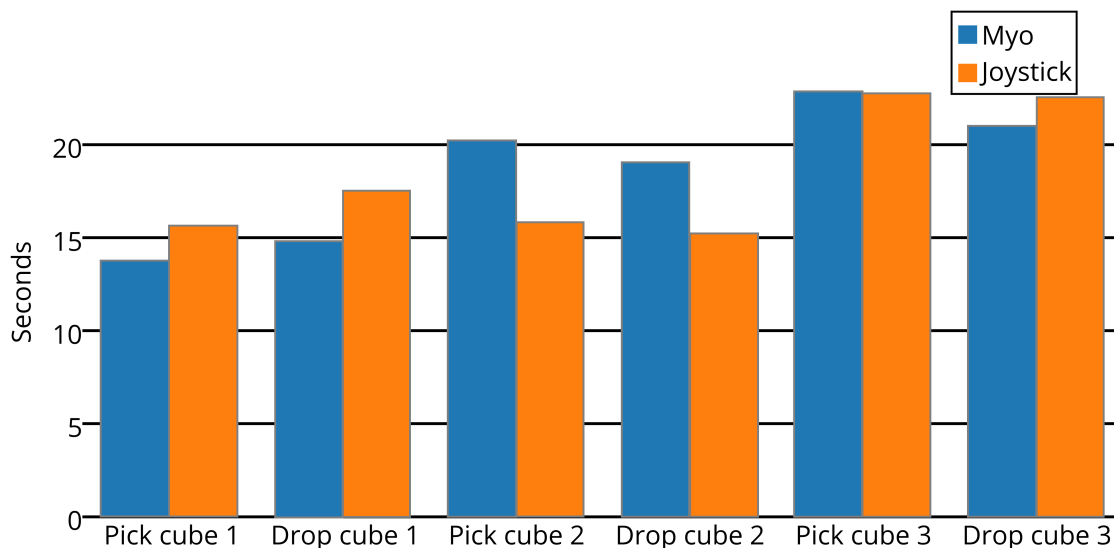


Figure 2.4 – Average time taken to complete each section of the speed challenge. Both the Myo and the joystick performed similarly, with the total average difference being less than 4 s in favor of the joystick.

2.8.3 Classifier stability

In order to assess the stability of the classifier, a set up similar to the one described in Section 2.8.1 was used. The classifier was first trained on a subject at time $T=0$. To assess the accuracy of the classifier, the test subject had to hold a random gesture for 10 s was repeated for a full 5 min, without rest. The test was conducted at least twice a day for six consecutive days. No re-training was done after $T=0$ and the armband was approximately at the same location on the forearm for each experiment (no marking to guide the user).

The accuracy was calculated by comparing the predicted gesture to the one requested by the computer. This however added errors well above the accuracy found when guiding the robot in real-time. Indeed, the time needed to read and start reacting to new instructions from the computer are considered errors in this setup. This is not a factor in a realistic guidance setting, where it is the user who decides when and which gesture to use. To mitigate this, we present two sets of results named : transition and no transition experiment. The first one does not try to alleviate the problems previously mentioned. The second simply does not consider the first

1.2 s after a change of gesture. The purpose of this is to remove the reaction-time, from the moment the subject receives the cue till the action is performed. Fig. 2.5 clearly shows that only a small performance degradation can be observed 6 days after training the classifier.

The two dotted lines of Fig 2.5 correspond to the linear regression lines. The first data point (at 1 hour) was not considered for the regression as it appears to be an outlier that would unjustifiably bias the results in favor of our proposed method. The poor performance of the first measurement, compared to the others, can be explained by the fact that the subject is still learning the decision function of the classifier. Another 6-day trial with a new classifier was run which yielded very similar results. Due to space consideration, they are not reported here. The experiment shows that our classifier is robust to small physical variations such as impedance of the skin changes from day to day and electrode placement inconsistency, muscle activity change, etc. In fact, the classifier shown in the video and the results for the Myo presented in Section 2.8.2 were achieved using a classifier that was trained 12 days prior to the speed challenge. Showing that the same classifier can be used extremely efficiently even several days after training.

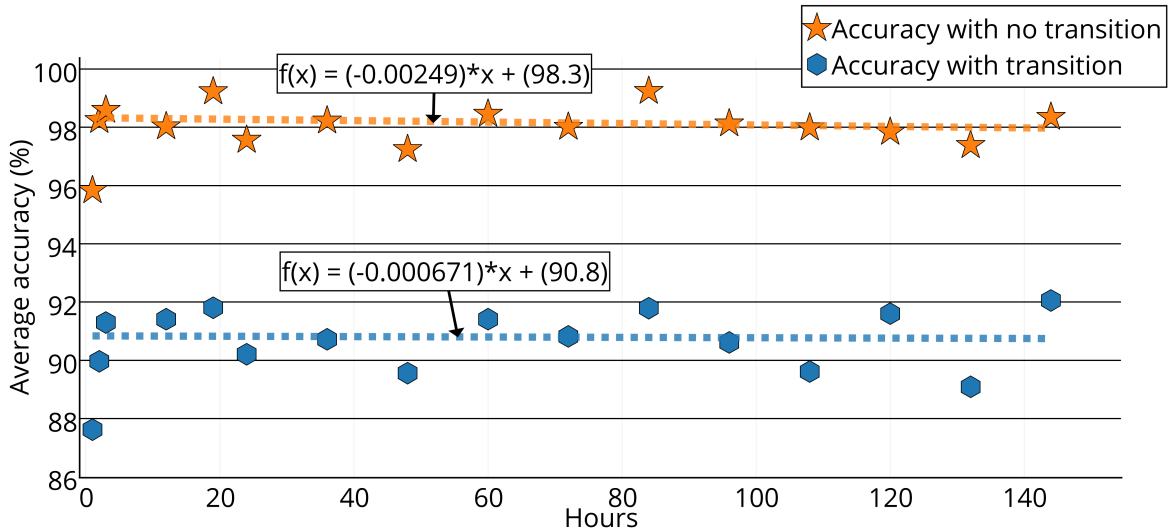


Figure 2.5 – Average accuracy of the first 6-day trials. The blue hexagons represent the accuracy over the complete 5- *min* period at different time after the training of the classifier. The orange stars is the accuracy over the 5- *min* period when omitting the first 1.2 s after each new gesture. The blue and orange dotted line come from the linear regression of the blue hexagons and orange stars data point respectively. The fact that the accuracy is almost constant through the 6-day period indicate that the classifier is robust to long-term use.

To assess if short-term muscle fatigue has a significant impact on the accuracy of the classifier, we examined its accuracy in 10 s intervals for a total of 5 *min*. We used the data obtained from the two 6-day trials, where a new gesture was requested every 10 s, yielding a total of 30 gestures over the 5 *min* period. Then the average accuracy for every 10 s gesture, over all the

two periods of six days, was combined. This enabled a clear view of any possible degradation in the accuracy, as time elapsed in the 5 *min* sequence. It is clear from Fig. 2.6 that muscle fatigue did not degrade the performance of the classifier in any noticeable way.

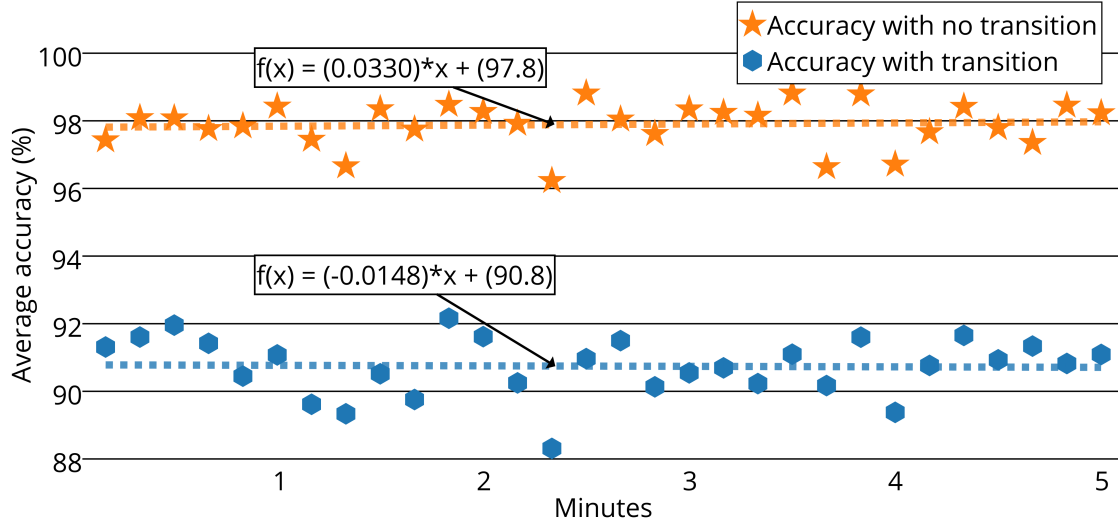


Figure 2.6 – Average accuracy over 5 *min* for the two 6-day trials. See Fig. 2.5 for the description of the dotted lines, blue hexagons and orange stars. The fact that the accuracy does not change over time shows that muscle fatigue is probably not adversely affecting our classifier.

The proposed classifier is thus not only accurate enough to perform complex and precise tasks, but is also robust to short term muscle fatigue, small displacement of electrodes and long term use without the need for recalibration. The average accuracy over the two 6-day trial is 97.9%.

2.9 Conclusion

In this work, a Myo armband was used to guide a robotic arm. The use of the armband offers several advantages for the intended users (inexpensive, no preparation time, easy to use). However the efficiency of the armband comes at the cost of quantity and quality of information. One of the major accomplishments in this paper has been to show that one can compensate for this lack of data quality with suitable machine learning approaches. Using the specified CNN architecture with spectrograms as input, our system was able to achieve state of the art results and obtain precise guidance of a 6DoF robotic arm using sEMG and orientation data that rivals the guidance with the joystick. The efficacy of the classifier was established when facing short-term muscle fatigue and long-term use achieving on average 97.9% during the two 6-day periods. The system was also shown to generalize effortlessly to different users.

Future work will focus on three main aspects. First, optimizing the CNN architecture in term of parameters and pruning it will allow both faster training and inference from the system. The presented approach will thus be more easily suitable for very low-power embedded hardware.

Secondly, the data collected from the participants of this study will be used to build a classifier that will require significantly less training data than presently required for a new user. To achieve this we intend to make use of training based on domain-adaptation techniques, which recover suitable information from one task and applies it to a similar one (new user). This can be used in a deep learning setting [15]. Finally, the classifier will be tested on upper limb amputees.

Acknowledgment

The authors would like to thank Alexandre Campeau-Lecours for sharing his expertise on JACO and his participation in the speed challenge.

2.10 Bibliography

- [1] S. Amsuss, L.P. Paredes, N. Rudigkeit, B. Graimann, M.J. Herrmann, and D. Farina. Long term stability of surface emg pattern classification for prosthetic control recognition. *Engineering in Medicine and Biology Society (EMBC)*, pages 3622–3625, 2013.
- [2] Manfredo Atzori, Matteo Cognolato, and Henning 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.
- [3] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, and Y. Bengio. Theano: new features and speed improvements. *Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop*, 2010.
- [4] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: a CPU and GPU math expression compiler. *Proceedings of the Python for Scientific Computing Conference (SciPy)*, 2012.
- [5] A. Boschmann and M. Platzner. Towards robust hd emg pattern recognition: Reducing electrode displacement effect using structural similarity. *IEEE EMBS conf.*, pages 4547–4550, 2014.
- [6] Steven W Brose, Douglas J Weber, Ben A Salatin, Garret G Grindle, Hongwu Wang, Juan J Vazquez, and Rory A Cooper. The role of assistive robotics in the lives of persons with disability. *American Journal of Physical Medicine & Rehabilitation*, 89(6):509–521, 2010.
- [7] H. Cecotti and A. Graeser. Convolutional neural network with embedded fourier transform for eeg classification. *19th International Conference on Pattern Recognition*, pages 1–4, 2008.

- [8] Y. Chen, T. Krishna, J. Emer, and V. Sze. Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. In *IEEE International Solid-State Circuits Conference, ISSCC 2016, Digest of Technical Papers*, pages 262–263, 2016.
- [9] Albert M Cook and Janice Miller Polgar. *Essentials of assistive technologies*. Elsevier Health Sciences, 2014.
- [10] K. Englehart and B. Hudgins. A robust, real-time control scheme for multifunction myoelectric control. *IEEE Transaction on Biomedical Engineering*, 50(7):848–854, 2003.
- [11] K. Englehart, B. Hudgins, and P.A. Parker. A wavelet-based continuous classification scheme for multifunction myoelectric control. *IEEE Transaction on Biomedical Engineering*, 48(3):302–311, 2001.
- [12] C. L. Fall, P. Turgeon, A. Campeau-Lecours, V. Maheu, M. Boukadoum, S. Roy, D. Massicotte, C. Gosselin, and B. Gosselin. Intuitive wireless control of a robotic arm for people living with an upper body disability. *IEEE EMBC conf*, pages 4399–4402, 2015.
- [13] D. Farina, N. Jiang, H. Rehbaum, A. Holobar, B. Graimann, H. Dietl, and O.C. Aszmann. The extraction of neural information from the surface emg for the control of upper-limb prostheses: Emerging avenues and challenges. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(4):797–809, 2014.
- [14] T.R. Farrell and R.F. Weir. The optimal controller delay for myoelectric prostheses. *IEEE Transaction on Neural systems and rehabilitation engineering*, 15(1):111–118, 2007.
- [15] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 17:1–35, 2016.
- [16] S. Han, J. Pool, J. Tran, and W. Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, pages 1135–1143, 2015.
- [17] L.J. Hargrove, K. Englehart, and B. Hudgins. A comparison of surface and intramuscular myoelectric signal classification. *IEEE Transaction on Biomedical Engineering*, 54(5):847–853, 2007.
- [18] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R.R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint*, arXiv:1207.0580, 2012.
- [19] B. Hudgins, P. Parker, and R.N. Scott. A new strategy for multifunction myoelectric control. *IEEE Transaction on Biomedical Engineering*, 40(1):82–94, 1993.

- [20] N. Jamalie and C. Sammut. Majority voting: Classification by tactile sensing using surface texture. *IEEE transactions on, Robotics*, 27(3):508–521, 2011.
- [21] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed 2016-02-28].
- [22] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436–444, 2015.
- [23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 11(86):2278–2324, 1998.
- [24] C.J. De Luca, L.D. Gilmore, M. Kuznetsov, and S.H. Roy. Filtering the surface emg signal: Movement artifact and baseline noise contamination. *Journal of Biomechanics*, 43(8):1573–1579, 2010.
- [25] M. Oskoei and H. Hu. Myoelectric control systems – a survey. *Biomedical Signal Processing and control*, 2:275–294, 2007.
- [26] Ki-Hee Park and Seong-Whan Lee. Movement intention decoding based on deep learning for multiuser myoelectric interfaces. In *2016 4th International Winter Conference on Brain-Computer Interface (BCI)*, pages 1–2. IEEE, 2016.
- [27] B. Peerdeman, D. Boere, H. Witteveen, R. Huis in ’t Veld, H. Hermens, S. Stramigioli, H. Rietman, P. Veltink, and S. Misra. Myoelectric forearm prostheses: State of the art from a user-centered perspective. *Journal of Rehabilitation Research & Development*, 48(6):719–738, 2011.
- [28] A. Phinyomark, S. Hirunviriya, C. Limsakul, and P. Phukpattaranont. Evaluation of emg feature extraction for hand movement recognition based on euclidean distance and standard deviation. *IEEE International Conference on Computer Telecommunications and Information Technology*, pages 856–860, 2010.
- [29] Jiantao Qiu, Jie Wang, Song Yao, Kaiyuan Guo, Boxun Li, Erjin Zhou, Jincheng Yu, Tianqi Tang, Ningyi Xu, Sen Song, et al. Going deeper with embedded fpga platform for convolutional neural network. In *ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pages 26–35. ACM, 2016.
- [30] T. N. Sainath, A. r. Mohamed, B. Kingsbury, and B. Ramabhadran. Deep convolutional neural networks for lvcsr. *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8614–8618, 2013.
- [31] D.F. Stegeman and B.G. Lapatki B.U. Kleine. High-density surface emg: Techniques and applications at a motor unit level. *Biocybernetics and Biomedical Engineering*, 32(3), 2012.

- [32] T. Xueyan, L. Yunhui, L. Congyi, and S. Dong. Hand motion classification using a multi-channel surface electromyography sensor. *Sensors*, pages 1130–1147, 2012.
- [33] C. Yang, S. Chang, P. Liang, Z. Li, and C.Y. Su. Teleoperated robot writing using emg signals. *IEEE International Conference on Information and Automation*, pages 2264–2269, 2015.
- [34] D. Yang, W. Yang, Q. Huang, and H. Liu. Classification of multiple finger motions during dynamic upper limb movements. *IEEE Journal of Biomedical and Health Informatics*, PP(99):1–1, 2015.
- [35] M.D. Zeiler. Adadelata: An adaptative learning rate method. *arXiv preprint*, arXiv:1212.5701, 2012.

Chapter 3

Deep learning for electromyographic hand gesture signal classification using transfer learning.

3.1 Reference

Ulysse Côté-Allard, Cheikh Latyr Fall, Alexandre Drouin, Alexandre Campeau-Lecours, Clément Gosselin, Kyrre Glette, François Laviolette[†], and Benoit Gosselin[†]. "Deep learning for electromyographic hand gesture signal classification using transfer learning." IEEE Transactions on Neural Systems and Rehabilitation Engineering 27, no. 4 (2019): 760-771.

[†]These authors share senior authorship

3.2 Context

This journal paper is the extended version of a conference paper [15] published at the IEEE Systems, Man (SMC) flagship conference. The conference article received the best paper award from the IEEE Systems, Man, and Cybernetics conference.

During the recording of the "pre-training" dataset used in this paper, and more specifically for the robotic arm control experiments of the SMC's conference paper, it became clear that a major factor affecting the system's usability came not from the sEMG interface, but from the orientation control algorithm of the robot. Essentially, the frame-of-reference of robotic arm is fixed within their end-effector. Meaning that inputting the command "up" can have drastically different effect depending on how much the robotic arm's wrist had rotated. Consequently, with Prof Alexandre Campeau-Lecours, Dinh-Son Vu, Dr François Routhier, Prof Benoit Gosselin and Prof Clément Gosselin, we proposed a new adaptive orientation control algorithm which was shown to be more intuitive (and resulting in significantly less errors and control time) than

the current standard. This work is presented in a conference paper [1] and expanded upon in a journal paper [2]. The algorithm was subsequently employed for any new robotic arm control within this thesis with great success. Note that while I share first co-authorship of the conference paper, this work was too tangential to my thesis to be included as a chapter.

[1] Vu, Dinh-Son[‡], Ulysse Côté Allard[‡], Clément Gosselin, François Routhier, Benoit Gosselin, and Alexandre Campeau-Lecours. "Intuitive adaptive orientation control of assistive robots for people living with upper limb disabilities." In 2017 International Conference on Rehabilitation Robotics (ICORR), pp. 795-800. IEEE, 2017.

[‡]These authors share first authorship

[2] Campeau-Lecours, Alexandre, Ulysse Côté-Allard, Dinh-Son Vu, François Routhier, Benoit Gosselin, and Clément Gosselin. "Intuitive Adaptive Orientation Control for Enhanced Human-Robot Interaction." IEEE Transactions on Robotics 35, no. 2 (2018): 509-520.

3.3 Résumé

Depuis plusieurs années, les algorithmes d'apprentissage profond sont devenus quasi-omniprésents dans le domaine de l'apprentissage machine de par leur capacité inégalée à apprendre automatiquement des caractéristiques discriminantes à partir de grandes quantités de données. Cependant, dans le domaine de la reconnaissance des gestes basée sur l'électromyographie, les algorithmes d'apprentissage profond sont rarement utilisés puisqu'ils nécessitent d'une seule personne de générer des quantités importantes de données pour chaque nouvelle utilisation.

L'hypothèse de cet article est que des caractéristiques générales et informatives peuvent être apprises à partir de grandes quantités de données générées en agrégeant les signaux de plusieurs utilisateurs. Réduisant ainsi le temps nécessaire pour chaque individu à enregistrer de nouvelles données, tout en améliorant la reconnaissance des mouvements du système. Cet article propose donc un nouvel algorithme d'apprentissage par transfert appliqué sur les données agrégées des utilisateurs, pour profiter de la capacité des algorithmes d'apprentissage profond d'apprendre des caractéristiques discriminantes à partir d'un grand ensemble de données.

Deux ensembles de données provenant respectivement de 19 et 17 participants (le premier est utilisé pour le préentraînement) ont été enregistrés pour ce travail grâce au *Myo Armband*. Un troisième jeu de données provenant de la base de données *NinaPro* contenant les signaux musculaires de dix participants enregistrés avec le *Myo Armband* est également utilisé.

Trois réseaux d'apprentissage profond utilisant trois modalités différentes (EMG brute, spectrogrammes et transformation par ondelettes continues (CWT)) sont testés sur le deuxième et troisième ensemble de données. L'algorithme d'apprentissage par transfert proposé améliore systématiquement et de manière significative les performances des trois réseaux sur les deux

ensembles de données. L’algorithme atteint une précision en temps différé de 98.31% pour 7 gestes sur 17 participants pour le réseau basé sur les CWT et de 68.98% pour 18 gestes sur dix participants pour le réseau recevant les données brutes.

Finalement, une étude de cas avec huit participants suggère que la rétroaction en temps réel permet aux utilisateurs d’adapter leur stratégie d’activation musculaire, ce qui réduit la dégradation de la performance du système dans le temps.

3.4 Abstract

In recent years, deep learning algorithms have become increasingly more prominent for their unparalleled ability to automatically learn discriminant features from large amounts of data. However, within the field of electromyography-based gesture recognition, deep learning algorithms are seldom employed as they require an unreasonable amount of effort from a single person, to generate tens of thousands of examples.

This work’s hypothesis is that general, informative features can be learned from the large amounts of data generated by aggregating the signals of multiple users, thus reducing the recording burden while enhancing gesture recognition. Consequently, this paper proposes applying transfer learning on aggregated data from multiple users, while leveraging the capacity of deep learning algorithms to learn discriminant features from large datasets. Two datasets comprised of 19 and 17 able-bodied participants respectively (the first one is employed for pre-training) were recorded for this work, using the Myo Armband. A third Myo Armband dataset was taken from the NinaPro database and is comprised of 10 able-bodied participants. Three different deep learning networks employing three different modalities as input (raw EMG, Spectrograms and Continuous Wavelet Transform (CWT)) are tested on the second and third dataset. The proposed transfer learning scheme is shown to systematically and significantly enhance the performance for all three networks on the two datasets, achieving an offline accuracy of 98.31% for 7 gestures over 17 participants for the CWT-based ConvNet and 68.98% for 18 gestures over 10 participants for the raw EMG-based ConvNet. Finally, a use-case study employing eight able-bodied participants suggests that real-time feedback allows users to adapt their muscle activation strategy which reduces the degradation in accuracy normally experienced over time.

3.5 Introduction

Robotics and artificial intelligence can be leveraged to increase the autonomy of people living with disabilities. This is accomplished, in part, by enabling users to seamlessly interact with robots to complete their daily tasks with increased independence. In the context of hand prosthetic control, muscle activity provides an intuitive interface on which to perform hand

gesture recognition [54]. This activity can be recorded by surface electromyography (sEMG), a non-invasive technique widely adopted both in research and clinical settings. The sEMG signals, which are non-stationary, represent the sum of subcutaneous motor action potentials generated through muscular contraction [54]. Artificial intelligence can then be leveraged as the bridge between sEMG signals and the prosthetic behavior.

The literature on sEMG-based gesture recognition primarily focuses on feature engineering, with the goal of characterizing sEMG signals in a discriminative way [54, 58, 60]. Recently, researchers have proposed deep learning approaches [4, 5, 19], shifting the paradigm from feature engineering to feature learning. Regardless of the method employed, the end-goal remains the improvement of the classifier’s robustness. One of the main factors for accurate predictions, especially when working with deep learning algorithms, is the amount of training data available. Hand gesture recognition creates a peculiar context where a single user cannot realistically be expected to generate tens of thousands of examples in a single sitting. Large amounts of data can however be obtained by aggregating the recordings of multiple participants, thus fostering the conditions necessary to learn a general mapping of users’ sEMG signal. This mapping might then facilitate the hand gestures’ discrimination task with new subjects. Consequently, deep learning offers a particularly attractive context from which to develop a Transfer Learning (TL) algorithm to leverage inter-user data by pre-training a model on multiple subjects before training it on a new participant.

As such, the main contribution of this work is to present a new TL scheme employing a convolutional network (ConvNet) to leverage inter-user data within the context of sEMG-based gesture recognition. A previous work [15] has already shown that learning simultaneously from multiple subjects significantly enhances the ConvNet’s performance whilst reducing the size of the required training dataset typically seen with deep learning algorithms. This paper expands upon the aforementioned conference paper’s work, improving the TL algorithm to reduce its computational load and improving its performance. Additionally, three new ConvNet architectures, employing three different input modalities, specifically designed for the robust and efficient classification of sEMG signals are presented. The raw signal, short-time Fourier transform-based spectrogram and Continuous Wavelet Transform (CWT) are considered for the characterization of the sEMG signals to be fed to these ConvNets. To the best of the authors’ knowledge, this is the first time that CWTs are employed as features for the classification of sEMG-based hand gesture recognition (although they have been proposed for the analysis of myoelectric signals [41]). Another major contribution of this article is the publication of a new sEMG-based gesture classification dataset comprised of 36 able-bodied participants. This dataset and the implementation of the ConvNets along with their TL augmented version are made readily available¹. Finally, this paper further expands the aforementioned conference paper by proposing a use-case experiment on the effect of real-time feedback on the online

¹<https://github.com/Giguelingueling/MyoArmbandDataset>

performance of a classifier without recalibration over a period of fourteen days. Note that, due to the stochastic nature of the algorithms presented in this paper, unless stated otherwise, all experiments are reported as an average of 20 runs.

This paper is organized as follows. An overview of the related work in hand gesture recognition through deep learning and transfer learning/domain adaptation is given in Sec. 3.6. Sec. 3.7 presents the proposed new hand gesture recognition dataset, with data acquisition and processing details alongside an overview of the NinaPro DB5 dataset. A presentation of the different state-of-the-art feature sets employed in this work is given in Sec. 3.8. Sec. 3.9 thoroughly describes the proposed networks’ architectures, while Sec. 6.8 presents the TL algorithm used to augment said architecture. Moreover, comparisons with the state-of-the-art in gesture recognition are given in Sec. 3.11. A real-time use-case experiment on the ability of users to counteract signal drift from sEMG signals is presented in Sec. 3.12. Finally, results are discussed in Sec. 3.13.

3.6 Related Work

sEMG signals can vary significantly between subjects, even when precisely controlling for electrode placement [10]. Regardless, classifiers trained from a user can be applied to new participants achieving slightly better than random performances [10] and high accuracy (85% over 6 gestures) when augmented with TL on never before seen subjects [68]. As such, sophisticated techniques have been proposed to leverage inter-user information. For example, research has been done to find a projection of the feature space that bridges the gap between an original subject and a new user [43, 12]. Several works have also proposed leveraging a pre-trained model removing the need to simultaneously work with data from multiple users [73, 55, 53]. These non-deep learning TL approaches showed important performance gains compared to their non-augmented versions. Although, some of these gains might be due to the baseline’s poorly optimized hyperparameters [30].

Short-Time Fourier Transform (STFT) have been sparsely employed in the last decades for the classification of sEMG data [22, 75]. A possible reason for this limited interest in STFT is that much of the research on sEMG-based gesture recognition focuses on designing feature ensembles [58]. Because STFT on its own generates large amounts of features and are relatively computationally expensive, they can be challenging to integrate with other feature types. Additionally, STFTs have also been shown to be less accurate than Wavelet Transforms [22] on their own for the classification of sEMG data. Recently however, STFT features, in the form of spectrograms, have been applied as input feature space for the classification of sEMG data by leveraging ConvNets [4, 19].

CWT features have been employed for electrocardiogram analysis [1], electroencephalography [24] and EMG signal analysis, but mainly for lower limbs [40, 38]. Wavelet-based features

have been used in the past for sEMG-based hand gesture recognition [20]. The features employed however, are based on the Discrete Wavelet Transform [72] and the Wavelet Packet Transform (WPT) [22] instead of the CWT. This preference might be due to the fact that both DWT and WPT are less computationally expensive than the CWT and are thus better suited to be integrated into an ensemble of features. Similarly to spectrograms however, CWT offers an attractive image-like representation to leverage ConvNets for sEMG signal classification and can now be efficiently implemented on embedded systems (see Appendix 3.B). To the best of the authors’ knowledge, this is the first time that CWT is utilized for sEMG-based hand gesture recognition.

Recently, ConvNets have started to be employed for hand gesture recognition using single array [4, 5] and matrix [27] of electrodes. Additionally, other authors applied deep learning in conjunction with domain adaptation techniques [19] but for inter-session classification as opposed to the inter-subject context of this paper. A thorough overview of deep learning techniques applied to EMG classification is given in [62]. To the best of our knowledge, this paper, which is an extension of [15], is the first time inter-user data is leveraged through TL for training deep learning algorithms on sEMG data.

3.7 sEMG datasets

3.7.1 Myo Dataset

One of the major contributions of this article is to provide a new, publicly available, sEMG-based hand gesture recognition dataset, referred to as the *Myo Dataset*. This dataset contains two distinct sub-datasets with the first one serving as the *pre-training dataset* and the second as the *evaluation dataset*. The former, which is comprised of 19 able-bodied participants, should be employed to build, validate and optimize classification techniques. The latter, comprised of 17 able-bodied participants, is utilized only for the final testing. To the best of our knowledge, this is the largest dataset published utilizing the commercially available Myo Armband (Thalmic Labs) and it is our hope that it will become a useful tool for the sEMG-based hand gesture classification community.

The data acquisition protocol was approved by the Comités d’Éthique de la Recherche avec des êtres humains de l’Université Laval (approbation number: 2017-026/21-02-2016) and informed consent was obtained from all participants.

sEMG Recording Hardware

The electromyographic activity of each subject’s forearm was recorded with the Myo Armband; an 8-channel, dry-electrode, low-sampling rate (200Hz), low-cost consumer-grade sEMG armband.

The Myo is non-intrusive, as the dry-electrodes allow users to simply slip the bracelet on without any preparation. Comparatively, gel-based electrodes require the shaving and washing of the skin to obtain optimal contact between the subject’s skin and electrodes. Unfortunately, the convenience of the Myo Armband comes with limitations regarding the quality and quantity of the sEMG signals that are collected. Indeed, dry electrodes, such as the ones employed in the Myo, are less accurate and robust to motion artifact than gel-based ones [70]. Additionally, while the recommended frequency range of sEMG signals is 5-500Hz [50] requiring a sampling frequency greater or equal to 1000Hz, the Myo Armband is limited to 200Hz. This information loss was shown to significantly impact the ability of various classifiers to differentiate between hand gestures [59]. As such, robust and adequate classification techniques are needed to process the collected signals accurately.

Time-Window Length

For real-time control in a closed loop, input latency is an important factor to consider. A maximum latency of 300ms was first recommended in [36]. Even though more recent studies suggest that the latency should optimally be kept between 100-250ms [23, 69], the performance of the classifier should take priority over speed [23, 57]. As is the case in [15], a window size of 260ms was selected to achieve a reasonable number of samples between each prediction due to the low frequency of the Myo.

Labeled Data Acquisition Protocol

The seven hand/wrist gestures considered in this work are depicted in Fig. 3.1. For both sub-datasets, the labeled data was created by requiring the user to hold each gesture for five seconds. The data recording was manually started by a researcher only once the participant correctly held the requested gesture. Generally, five seconds was given to the user between each gesture. This rest period was not recorded and as a result, the final dataset is balanced for all classes. The recording of the full seven gestures for five seconds is referred to as a *cycle*, with four cycles forming a *round*. In the case of the *pre-training dataset*, a single *round* is available per subject. For the *evaluation dataset* three *rounds* are available with the first *round* utilized for training (i.e. 140s per participant) and the last two for testing (i.e. 240s per participant).

During recording, participants were instructed to stand up and have their forearm parallel to the floor and supported by themselves. For each of them, the armband was systematically tightened to its maximum and slid up the user’s forearm, until the circumference of the armband matched that of the forearm. This was done in an effort to reduce bias from the researchers, and to emulate the wide variety of armband positions that end-users without prior knowledge of optimal electrode placement might use (see Fig. 3.2). While the electrode placement was not controlled for, the orientation of the armband was always such that the blue light bar on the Myo was facing towards the hand of the subject. Note that this is the case for both left

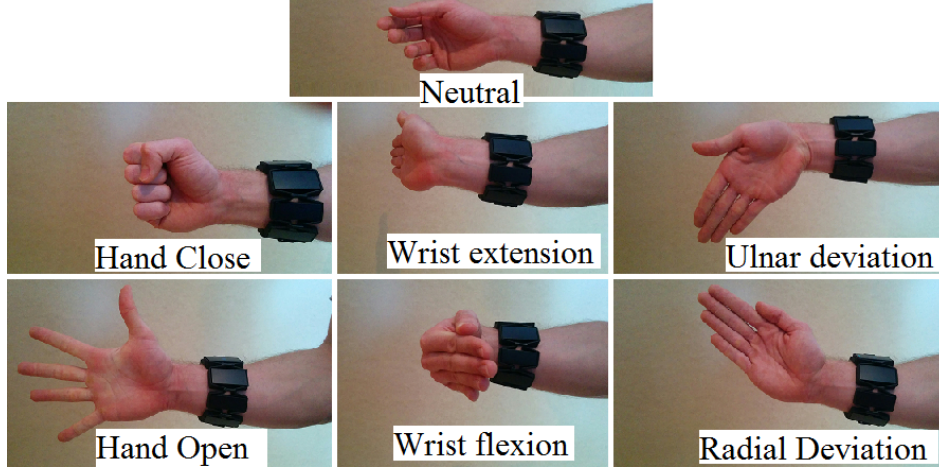


Figure 3.1 – The 7 hand/wrist gestures considered in the *Myo Dataset*.

and right handed subjects. The raw sEMG data of the Myo is what is made available with this dataset.



Figure 3.2 – Examples of the range of armband placements on the subjects' forearm

Signal processing must be applied to efficiently train a classifier on the data recorded by the Myo armband. The data is first separated by applying sliding windows of 52 samples ($260ms$) with an overlap of $235ms$ (i.e. 7×190 samples for one cycle (5s of data)). Employing windows of $260ms$ allows $40ms$ for the pre-processing and classification process, while still staying within the $300ms$ target [36]. Note that utilizing sliding windows is viewed as a form of data augmentation in the present context (see Appendix 3.A). This is done for each gesture in each cycle on each of the eight channels. As such, in the dataset, an *example* corresponds to the eight windows associated with their respective eight channels. From there, the processing depends on the classification techniques employed which will be detailed in Sec. 3.8 and 3.9.

3.7.2 NinaPro DB5

The *NinaPro DB5* is a dataset built to benchmark sEMG-based gesture recognition algorithms [63]. This dataset, which was recorded with the Myo Armband, contains data from 10 able-bodied participants performing a total of 53 different movements (including neutral) divided into three exercise sets. The second exercise set, which contains 17 gestures + neutral

gesture, is of particular interest, as it includes all the gestures considered so far in this work. The 11 additional gestures which are presented in [6] include wrist pronation, wrist supination and diverse finger extension amongst others. While this particular dataset was recorded with two Myo Armband, only the lower armband is considered as to allow direct comparison to the preceding dataset.

Data Acquisition and Processing

Each participant was asked to hold a gesture for five seconds followed by three seconds of neutral gesture and to repeat this action five more times (total of six repetitions). This procedure was repeated for all the movements contained within the dataset. The first four repetitions serve as the training set (20s per gesture) and the last two (10s per gesture) as the test set for each gesture. Note that the *rest* movement (i.e. neutral gesture) was treated identically as the other gestures (i.e. first four repetitions for training (12s) and the next two for testing (6s)).

All data processing (e.g. window size, window overlap) are exactly as described in the previous sections.

3.8 Classic sEMG Classification

Traditionally, one of the most researched aspects of sEMG-based gesture recognition comes from feature engineering (i.e. manually finding a representation for sEMG signals that allows easy differentiation between gestures). Over the years, several efficient combinations of features both in the time and frequency domain have been proposed [61, 21, 7, 44]. This section presents the feature sets used in this work. See Appendix 3.C for a description of each feature.

3.8.1 Feature Sets

As this paper’s main purpose is to present a deep learning-based TL approach to the problem of sEMG hand gesture recognition, contextualizing the performance of the proposed algorithms within the current state-of-the-art is essential. As such, four different feature sets were taken from the literature to serve as a comparison basis. The four feature sets will be tested on five of the most common classifiers employed for sEMG pattern recognition: Support Vector Machine (SVM) [7], Artificial Neural Networks (ANN) [2], Random Forest (RF) [7], K-Nearest Neighbors (KNN) [7] and Linear Discriminant Analysis (LDA) [44]. Hyperparameters for each classifier were selected by employing three fold cross-validation alongside random search, testing 50 different combinations of hyperparameters for each participant’s dataset for each classifier. The hyperparameters considered for each classifier are presented in Appendix 3.D.

As is often the case, dimensionality reduction is applied [54, 60, 45]. LDA was chosen to perform feature projection as it is computationally inexpensive, devoid of hyperparameters and was

shown to allow for robust classification accuracy for sEMG-based gesture recognition [44, 81]. A comparison of the accuracy obtained with and without dimensionality reduction on the *Myo Dataset* is given in Appendix 3.E. This comparison shows that in the vast majority of cases, the dimensionality reduction both reduced the computational load and enhanced the average performances of the feature sets.

The implementation employed for all the classifiers comes from the scikit-learn (v.1.13.1) Python package [56]. The four feature sets employed for comparison purposes are:

Time Domain Features (TD) [21]

This set of features, which is probably the most commonly employed in the literature [59], often serves as the basis for bigger feature sets [54, 44, 63]. As such, TD is particularly well suited to serve as a baseline comparison for new classification techniques. The four features are: Mean Absolute Value (MAV), Zero Crossing (ZC), Slope Sign Changes (SSC) and Waveform Length (WL).

Enhanced TD [44]

This set of features includes the TD features in combination with Skewness, Root Mean Square (RMS), Integrated EMG (IEMG), Autoregression Coefficients (AR) (P=11) and the Hjorth Parameters. It was shown to achieve excellent performances on a setup similar to the one employed in this article.

Nina Pro Features [7, 63]

This set of features was selected as it was found to perform the best in the article introducing the NinaPro dataset. The set consists of the the following features: RMS, Marginal Discrete Wavelet Transform (mDWT) (wavelet=db7, S=3), EMG Histogram (HIST) (bins=20, threshold=3 σ) and the TD features.

SampEn Pipeline [61]

This last feature combination was selected among fifty features that were evaluated and ranked to find the most discriminating ones. The SampEn feature was ranked first amongst all the others. The best multi-features set found was composed of: SampEn(m=2, r=0.2 σ), Cepstral Coefficient (order=4), RMS and WL.

3.9 Deep Learning Classifiers Overview

ConvNets tend to be computationally expensive and thus ill-suited for embedded systems, such as those required when guiding a prosthetic. However, in recent years, algorithmic

improvements and new hardware architectures have allowed for complex networks to run on very low power systems (see Appendix 3.B). As previously mentioned, the inherent limitations of sEMG-based gesture recognition force the proposed ConvNets to contend with a limited amount of data from any single individual. To address the over-fitting issue, Monte Carlo Dropout (MC Dropout) [26], Batch Normalization (BN) [37], and early stopping are employed.

3.9.1 Batch Normalization

BN is a technique that accelerates training and provides some form of regularization with the aims of maintaining a standard distribution of hidden layer activation values throughout training [37]. BN accomplishes this by normalizing the mean and variance of each dimension of a batch of examples. To achieve this, a linear transformation based on two learned parameters is applied to each dimension. This process is done independently for each layer of the network. Once training is completed, the whole dataset is fed through the network one last time to compute the final normalization parameters in a layer-wise fashion. At test time, these parameters are applied to normalize the layer activations. BN was shown to yield faster training times whilst allowing better generalization.

3.9.2 Proposed Convolutional Network Architectures

Videos are a representation of how spatial information (images) change through time. Previous works have combined this representation with ConvNets to address classification tasks [8, 42]. One such successful algorithm is the slow-fusion model [42] (see Fig. 3.3).

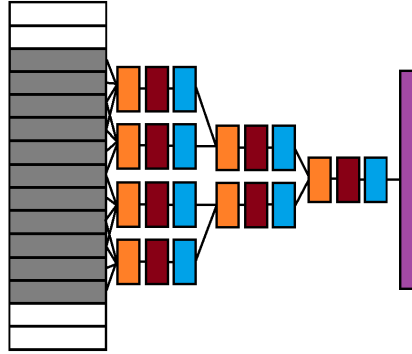


Figure 3.3 – Typical slow-fusion ConvNet architecture [42]. In this graph, the input (represented by grey rectangles) is a video (i.e. a sequence of images). The model separates the temporal part of the examples into disconnected parallel layers, which are then slowly fused together throughout the network.

When calculating the spectrogram of a signal, the information is structured in a Time x Frequency fashion (Time x Scale for CWT). When the signal comes from an array of electrodes, these examples can naturally be structured as Time x Spatial x Frequency (Time x Spatial x Scale for CWT). As such, the motivation for using a slow-fusion architecture based

ConvNet in this work is due to the similarities between videos data and the proposed characterization of sEMG signals, as both representations have analogous structures (i.e. Time x Spatial x Spatial for videos) and can describe non-stationary information. Additionally, the proposed architectures inspired by the slow-fusion model were by far the most successful of the ones tried on the pre-training dataset.

ConvNet for Spectrograms

The spectrograms, which are fed to the ConvNet, were calculated with Hann windows of length 28 and an overlap of 20 yielding a matrix of 4×15 . The first frequency band was removed in an effort to reduce baseline drift and motion artifact. As the armband features eight channels, eight such spectrograms were calculated, yielding a final matrix of $4 \times 8 \times 14$ (Time x Channel x Frequency).

The implementation of the spectrogram ConvNet architecture (see Fig. 3.4) was created with Theano [3] and Lasagne [17]. As usual in deep learning, the architecture was created in a trial and error process taking inspiration from previous architectures (primarily [4, 19, 42, 15]). The non-linear activation functions employed are the parametric exponential linear unit (PELU) [74] and PReLU [32]. ADAM [46] is utilized for the optimization of the ConvNet (learning rate=0.00681292). The deactivation rate for MC Dropout is set at 0.5 and the batch size at 128. Finally, to further reduce overfitting, early stopping is employed by randomly removing 10% of the data from the training and using it as a validation set at the beginning of the optimization process. Note that learning rate annealing is applied with a factor of 5 when the validation loss stops improving. The training stops when two consecutive decays occurs with no network performance amelioration on the validation set. All hyperparameter values were found by a random search on the *pre-training dataset*.

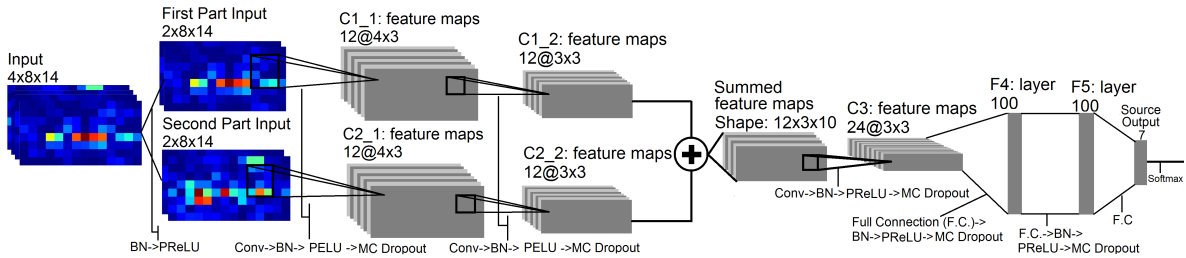


Figure 3.4 – The proposed spectrogram ConvNet architecture to leverage spectrogram examples employing 67 179 learnable parameters. To allow the slow fusion process, the input is first separated equally into two parts with respect to the time axis. The two branches are then fused together by element-wise summing the feature maps together. In this figure, *Conv* refer to *Convolution* and *F.C.* to *Fully Connected* layers.

ConvNet for Continuous Wavelet Transforms

The architecture for the CWT ConvNet, (Fig. 3.5), was built in a similar fashion as the spectrogram ConvNet one. Both the *Morlet* and *Mexican Hat* wavelet were considered for this work due to their previous application in EMG-related work [65, 66]. In the end, the Mexican Hat wavelet was selected, as it was the best performing during cross-validation on the *pre-training dataset*. The CWTs were calculated with 32 scales yielding a 32x52 matrix. Downsampling is then applied at a factor of 0.25 employing spline interpolation of order 0 to reduce the computational load of the ConvNet during training and inference. Following downsampling, similarly to the spectrogram, the last row of the calculated CWT was removed as to reduce baseline drift and motion artifact. Additionally, the last column of the calculated CWT was also removed as to provide an even number of time-columns from which to perform the slow-fusion process. The final matrix shape is thus 12x8x7 (i.e. Time x Channel x Scale). The MC Dropout deactivation rate, batch size, optimization algorithm, and activation functions remained unchanged. The learning rate was set at 0.0879923 (found by cross-validation).

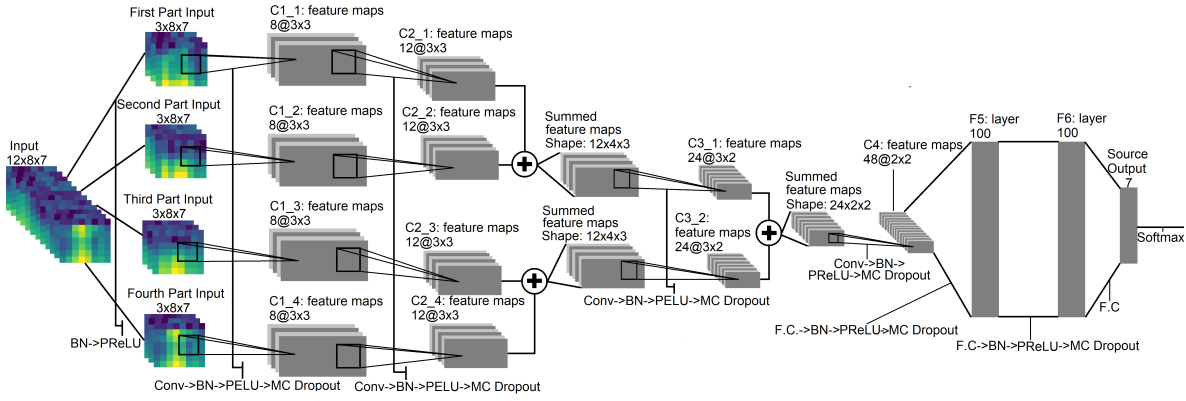


Figure 3.5 – The proposed CWT ConvNet architecture to leverage CWT examples using 30 219 learnable parameters. To allow the slow fusion process, the input is first separated equally into four parts with respect to the time axis. The four branches are then slowly fused together by element-wise summing the feature maps together. In this figure, *Conv* refers to *Convolution* and *F.C.* to *Fully Connected* layers.

ConvNet for raw EMG

A third ConvNet architecture taking the raw EMG signal as input is also considered. This network will help assess if employing time-frequency features lead to sufficient gains in accuracy performance to justify the increase in computational cost. As the raw EMG represents a completely different modality, a new type of architecture must be employed. To reduce bias from the authors as much as possible, the architecture considered is the one presented in [83]. The *raw ConvNet* architecture can be seen in Fig. 3.6. This architecture was selected as it was also designed to classify a hand gesture dataset employing the Myo Armband. The architecture implementation (in PyTorch v.0.4.1) is exactly as described in [83] except for

the learning rate ($=1.1288378916846883e - 5$) which was found by cross-validation (tested 20 uniformly distributed values between $1e - 6$ to $1e - 1$ on a logarithm scale) and extending the length of the window size as to match with the rest of this manuscript. The *raw ConvNet* is

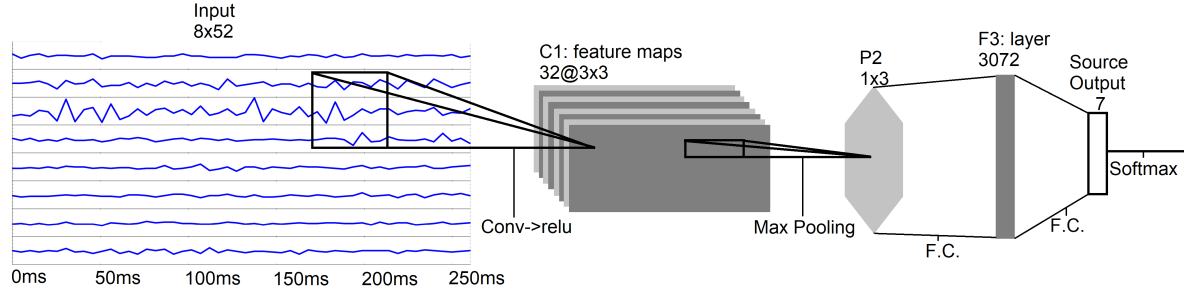


Figure 3.6 – The raw ConvNet architecture to leverage raw EMG signals. In this figure, *Conv* refers to *Convolution* and *F.C.* to *Fully Connected* layers.

further enhanced by introducing a second convolutional and pooling layer as well as adding dropout, BN, replacing RELU activation function with PReLU and using ADAM (learning rate= 0.002335721469090121) as the optimizer. The *enhanced raw ConvNet*'s architecture, which is shown in Fig. 3.7, achieves an average accuracy of 97.88% compared to 94.85% for the *raw ConvNet*. Consequently, all experiments using raw emg as input will employ the *raw enhanced ConvNet*.

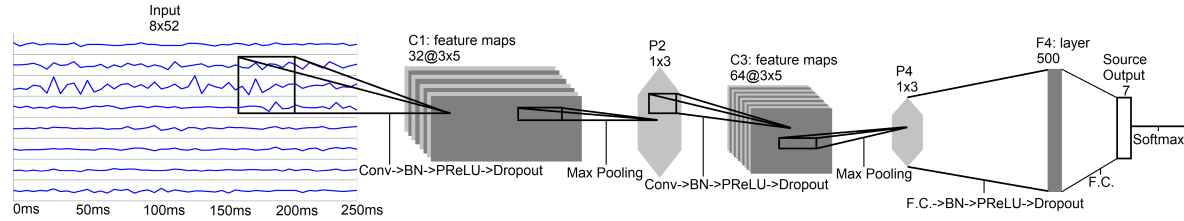


Figure 3.7 – The enhanced raw ConvNet architecture using 549 091 learnable parameters. In this figure, *Conv* refers to *Convolution* and *F.C.* to *Fully Connected* layers.

3.10 Transfer Learning

One of the main advantages of deep learning comes from its ability to leverage large amounts of data for learning. As it would be too time-consuming for a single individual to record tens of thousands of examples, this work proposes to aggregate the data of multiple individuals. The main challenge thus becomes to find a way to leverage data from multiple users, with the objective of achieving higher accuracy with less data. TL techniques are well suited for such a task, allowing the ConvNets to generate more general and robust features that can be applied to a new subject's sEMG activity.

As the data recording was purposefully as unconstrained as possible, the armband's orientation from one subject to another can vary widely. As such, to allow for the use of TL, automatic

alignment is a necessary first step. The alignment for each subject was made by identifying the most active channel (calculated using the IEMG feature) for each gesture on the first subject. On subsequent subjects, the channels were then circularly shifted until their activation for each gesture matched those of the first subject as closely as possible.

3.10.1 Progressive Neural Networks

Fine-tuning is the most prevalent TL technique in deep learning [9, 78]. It consists of training a model on a *source domain* (abundance of labeled data) and using the trained weights as a starting point when presented with a new task. However, fine-tuning can suffer from *catastrophic forgetting* [67], where relevant and important features learned during pre-training are lost on the *target domain* (i.e. new task). Moreover, by design, fine-tuning is ill-suited when significant differences exist between the source and the target, as it can bias the network into poorly adapted features for the task at hand. Progressive Neural Networks (PNN) [67] attempt to address these issues by pre-training a model on the source domain and freezing its weights. When a new task appears, a new network, with random initialization, is created and connected in a layer-wise fashion to the original network. This connection is done via non-linear lateral connections (See [67] for details).

3.10.2 Adaptive Batch Normalization

In opposition to the PNN architecture, which uses a different network for the source and the target, AdaBatch employs the same network for both tasks. The TL occurs by freezing all the network’s weights (learned during pre-training) when training on the target, except for the parameters associated with BN. The hypothesis behind this technique is that the label-related information (i.e. gestures) rests in the network model weights whereas the domain-related information (i.e. subjects) is stored in their BN statistic. In the present context, this idea can be generalized by applying a multi-stream AdaBatch scheme [19]. Instead of employing one *Source Network* per subject during pre-training, a single network is shared across all participants. However, the BN statistics from each subject are calculated independently from one another, allowing the ConvNet to extract more general and robust features across all participants. As such, when training the source network, the data from all subjects are aggregated and fed to the network together. It is important to note that each training batch is comprised solely of examples that belong to a single participant. This allows the update of the participant’s corresponding BN statistic.

3.10.3 Proposed Transfer Learning Architecture

The main tenet behind TL is that similar tasks can be completed in similar ways. The difficulty in this paper’s context is then to learn a mapping between the source and target task as to leverage information learned during pre-training. Training one network per source-task (i.e.

per participant) for the PNN is not scalable in the present context. However, by training a *Source Network* (presented in Sec. 3.9) shared across all participants of the *pre-training dataset* with the multi-stream AdaBatch and adding only a second network for the target task using the PNN architecture, the scaling problem in the current context vanishes. This second network will hereafter be referred to as the *Second Network*. The architecture of the *Second Network* is almost identical to the *Source Network*. The difference being in the activation functions employed. The *Source Network* leveraged a combination of PReLU and PELU, whereas the *Second Network* only employed PELU. This architecture choice was made through trial and error and cross-validation on the *pre-training dataset*. Additionally, the weights of both networks are trained and initialized independently. During pre-training, only the *Source Network* is trained to represent the information of all the participants in the *pre-training dataset*. The parameters of the *Source Network* are then frozen once pre-training is completed, except for the BN parameters as they represent the domain-related information and thus must retain the ability to adapt to new users.

Due to the application of the multi-stream AdaBatch scheme, the source task in the present context is to learn the *general* mapping between muscle activity and gestures. One can see the problem of learning such mapping between the target and the source task as learning a residual of the source task. For this reason, the *Source Network* shares information with the *Second Network* through an element-wise summation in a layer-by-layer fashion (see Fig. 3.8). The idea behind the merging of information through element-wise summation is two-fold. First, compared to concatenating the features maps (as in [15]) or employing non-linear lateral connections (like in [67]), element-wise summation minimizes the computational impact of connecting the *Source Network* and the *Second Network* together. Second, this provides a mechanism that fosters residual learning as inspired by Residual Networks [33]. Thus, the *Second Network* only needs to learn weights that express the difference between the new target and source task. All outputs from the *Source Network* layers to the *Second Network* are multiplied by learnable coefficients before the sum-connection. This scalar layer provides an easy mechanism to neuter the *Source Network's* influence on a layer-wise level. This is particularly useful if the new target task is so different that for some layers the information from the *Source Network* actually hinders learning. Note that a single-stream scheme (i.e. all subjects share statistics and BN parameters are also frozen on the *Source Network*) was also tried. As expected, this scheme's performances started to rapidly worsen as the number of source participants augmented, lending more credence to the initial AdaBatch hypothesis.

The combination of the *Source Network* and *Second Network* will hereafter be referred to as the *Target Network*. An overview of the final proposed architecture is presented in Fig. 3.8. During training of the *Source Network* (i.e. pre-training), MC Dropout rate is set at 35% and when training the *Target Network* the rate is set at 50%. Note that different architecture choices for the *Source Network* and *Second Network* were required to augment the performance

of the system as a whole. This seems to indicate that the two tasks (i.e. learning a general mapping of hand gestures and learning a specific mapping), might be different enough that even greater differentiation through specialization of the two networks might increase the performance further.

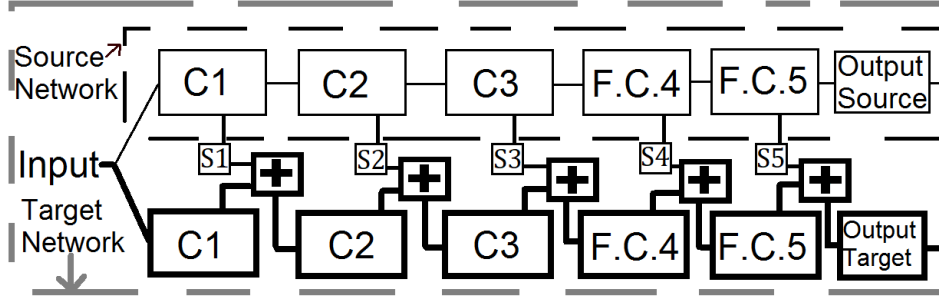


Figure 3.8 – The PNN-inspired architecture. This figure represents the case with the spectrogram ConvNet. Note that the TL behavior is the same for the Raw-based or CWT-based ConvNet. C1,2,3 and F.C.4,5 correspond to the three stages of convolutions and two stages of fully connected layers respectively. The S_i ($i=1..5$) boxes represent a layer that scales its inputs by learned coefficients. The number of learned coefficients in one layer is the number of channels or the number of neurons for the convolutional and fully connected layers respectively. For clarity's sake, the slow fusion aspect is omitted from the representation although they are present for both the spectrogram and CWT-based ConvNet). The $+$ boxes represent the merging through an element-wise summation of the ConvNets' corresponding layers.

3.11 Classifier Comparison

3.11.1 Myo Dataset

All pre-trainings in this section were done on the *pre-training dataset* and all training (including for the traditional machine learning algorithms) were done on the first *round* of the *evaluation dataset*.

Comparison with Transfer Learning

Considering each participant as a separate dataset allows for the application of the one-tail Wilcoxon signed-rank test [76] ($n = 17$). Table 3.1 shows a comparison of each ConvNet with their TL augmented version. Accuracies are given for one, two, three and four cycles of training.

Comparison with State of the art

A comparison between the proposed CWT-based ConvNet and a variety of classifiers trained on the features sets presented in Sec. 3.8.1 is given in Table 3.2.

Table 3.1 – Classification accuracy of the ConvNets on the *Evaluation Dataset* with respect to the number of training cycles performed.

	Raw	Raw + TL	Spectrogram	Spectrogram + TL	CWT	CWT + TL
4 Cycles	97.08%	97.39%	97.14%	97.85%	97.95%	98.31%
STD	4.94%	4.07%	2.85%	2.45%	2.49%	2.16%
H0 (p-value)	0 (0.02187)	-	0 (0.00030)	-	0 (0.00647)	-
3 Cycles	96.22%	96.95%	96.33%	97.40%	97.22%	97.82%
STD	6.49%	4.88%	3.49%	2.91%	3.46%	2.41%
H0 (p-value)	0 (0.00155)	-	0 (0.00018)	-	0 (0.00113)	-
2 Cycles	94.53%	95.49%	94.19%	96.05%	95.17%	96.63%
STD	9.63%	7.26%	5.95%	6.00%	5.77%	4.54%
H0 (p-value)	0 (0.00430)	-	0 (0.00015)	-	0 (0.00030)	-
1 Cycle	89.04%	92.46%	88.51%	93.93%	89.02%	94.69%
STD	10.63%	7.79%	8.37%	6.56%	10.24%	5.58%
H0 (p-value)	0 (0.00018)	-	0 (0.00015)	-	0 (0.00015)	-

* The *one-tail Wilcoxon signed rank test* is applied to compare the ConvNet enhanced with the proposed TL algorithm to their non-augmented counterpart. Null hypothesis is rejected when $H_0 = 0$ ($p < 0.05$).

**The STD represents the pooled standard variation in accuracy for the 20 runs over the 17 participants.

Table 3.2 – Classifiers comparison on the *Evaluation Dataset* with respect to the number of training cycles performed.

	TD	Enhanced TD	Nina Pro	SampEn Pipeline	CWT	CWT + TL
4 Cycles	97.61% (LDA)	98.14% (LDA)	97.59% (LDA)	97.72% (LDA)	97.95%	98.31%
STD	2.63%	2.21%	2.74%	1.98%	2.49%	2.16%
Friedman Rank	3.94	2.71	4.29	3.47	3.94	2.65
H0	1	1	1	1	1	-
3 Cycles	96.33% (KNN)	97.33% (LDA)	96.76% (KNN)	96.87% (KNN)	97.22%	97.82%
STD	6.11%	3.24%	3.85%	5.06%	3.46%	2.41%
Friedman Rank	4.41	2.77	4.05	3.53	3.94	2.29
H0	0 (0.00483)	1	0 (0.02383)	1	0 (0.03080)	-
2 Cycles	94.12% (KNN)	94.79% (LDA)	94.23% (KNN)	94.68% (KNN)	95.17%	96.63%
STD	9.08%	7.82%	7.49%	8.31%	5.77%	4.54%
Friedman Rank	4.41	3.24	4.41	3.29	3.65	2.00
H0 (adjusted p-value)	0 (0.00085)	1	0 (0.00085)	1	0 (0.03080)	-
1 Cycle	90.77% (KNN)	91.25% (LDA)	90.21% (LDA)	91.66% (KNN)	89.02%	94.69%
STD	9.04%	9.44%	7.73%	8.74%	10.24%	5.58%
Friedman Rank	3.71	3.41	4.41	3.05	4.88	1.53
H0 (adjusted p-value)	0 (0.00208)	0 (0.00670)	0 (0.00003)	0 (0.01715)	0 (<0.00001)	-

*For brevity's sake, only the best performing classifier for each feature set in each cycle is reported (indicated in parenthesis).

**The STD represents the pooled standard variation in accuracy for the 20 runs over the 17 participants.

***The Friedman Ranking Test followed by the Holm's post-hoc test is performed.

As suggested in [16], a two-step procedure is employed to compare the deep learning algorithms with the current state-of-the-art. First, Friedman’s test ranks the algorithms amongst each other. Then, Holm’s post-hoc test is applied ($n = 17$) using the best ranked method as a comparison basis.

3.11.2 NinaPro Dataset

Comparison with Transfer Learning

Performance of the proposed ConvNet architecture alongside their TL augmented versions are investigated on the *NinaPro DB5*. As no specific pre-training dataset is available for the *NinaPro DB5*, the pre-training for each participant is done employing the training sets of the remaining nine participants. Table 3.3 shows the average accuracy over the 10 participants of the *NinaPro DB5* for one to four cycles. Similarly to Sec. 3.11.1, the one-tail Wilcoxon Signed rank test is performed for each cycle between each ConvNet and their TL augmented version.

Table 3.3 – Classification accuracy of the ConvNets on the *NinaPro DB5* with respect to the number of training cycles performed.

	Raw	Raw + TL	Spectrogram	Spectrogram + TL	CWT	CWT + TL
4 Repetitions	66.32%	68.98%	63.60%	65.10%	61.89%	65.57%
STD	3.94%	4.46%	3.94%	3.99%	4.12%	3.68%
H0 (p-value)	0 (0.00253)	-	0 (0.00253)	-	0 (0.00253)	-
3 Repetitions	61.91%	65.16%	60.09%	61.70%	58.37%	62.21%
STD	3.94%	4.46%	4.03%	4.29%	4.19%	3.93%
H0 (p-value)	0 (0.00253)	-	0 (0.00253)	-	0 (0.00253)	-
2 Repetitions	55.67%	60.12%	55.35%	57.19%	53.32%	57.53%
STD	4.38%	4.79%	4.50%	4.71%	3.72%	3.69%
H0 (p-value)	0 (0.00253)	-	0 (0.00253)	-	0 (0.00253)	-
1 Repetitions	46.06%	49.41%	45.59%	47.39%	42.47%	48.33%
STD	6.09%	5.82%	5.58%	5.30%	7.04%	5.07%
H0 (p-value)	0 (0.00467)	-	0 (0.00467)	-	0 (0.00253)	-

* The *Wilcoxon signed rank test* is applied to compare the ConvNet enhanced with the proposed TL algorithm to their non-augmented counterpart. Null hypothesis is rejected when $H_0 = 0$ ($p < 0.05$).

**The STD represents the pooled standard variation in accuracy for the 20 runs over the 17 participants.

Comparison with State of the art

Similarly to Sec. 3.11.1, a comparison between the TL-augmented ConvNet and the traditional classifier trained on the state-of-the-art feature set is given in Table 3.4. The accuracies are given for one, two, three and four cycles of training. A two-step statistical test with the Friedman test as the first step and Holm post-hoc as the second step is again employed.

Table 3.4 – Classifiers Comparison on the *NinaPro DB5* with respect to the number of repetitions used during training.

	TD	Enhanced TD	Nina Pro	SampEn Pipeline	Raw	Raw + TL
4 Repetitions	59.91% (RF)	59.57% (RF)	56.72% (RF)	62.30% (RF)	66.32%	68.98%
STD	3.50%	4.43%	4.01%	3.94%	3.77%	4.09%
Friedman Rank	4.30	4.60	6.00	3.00	2.10	1.00
H0 (Adjusted p-value)	0 (0.00024)	0 (0.00007)	0 (<0.00001)	0 (0.03365)	1	-
3 Repetitions	55.73% (RF)	55.32% (RF)	52.33% (RF)	58.24% (RF)	61.91%	65.16%
STD	3.75%	4.48%	4.63%	4.22%	3.94%	4.46%
Friedman Rank	4.40	4.60	6.00	3.00	2.00	1.00
H0 (Adjusted p-value)	0 (0.00014)	0 (0.00007)	0 (<0.00001)	0 (0.03365)	1	-
2 Repetitions	50.85% (RF)	50.08% (LDA)	46.85% (LDA)	53.00% (RF)	55.65%	60.12%
STD	4.29%	4.63%	4.81%	3.85%	4.38%	4.79%
Friedman Rank	4.20	4.60	6.00	3.10	2.10	1.00
H0 (Adjusted p-value)	0 (0.00039)	0 (0.00007)	0 (<0.00001)	0 (0.02415)	1	-
1 Repetitions	40.70% (RF)	40.86% (LDA)	37.60% (LDA)	42.26% (LDA)	46.06%	49.41%
STD	5.84%	6.91%	6.67%	5.78%	6.09%	5.82%
Friedman Rank	4.30	4.30	5.80	3.50	2.00	1.10
H0 (Adjusted p-value)	0 (0.00052)	0 (0.00052)	0 (<0.00001)	0 (0.00825)	1	-

*For brevity's sake, only the best performing classifier for each feature set is reported (indicated in parenthesis).

**The STD represents the pooled standard variation in accuracy for the 20 runs over the 17 participants.

***The Friedman Ranking Test followed by the Holm's post-hoc test is performed.

Out-of-Sample Gestures

A final test involving the *NinaPro DB5* was conducted to evaluate the impact on the proposed TL algorithm when the target is comprised solely of out-of-sample gestures (i.e. never-seen-before gestures). To do so, the proposed CWT ConvNet was trained and evaluated on the training and test set of the *NinaPro DB5* as described before, but considering only the gestures that were absent from the *pre-training dataset* (11 total). The CWT ConvNet was then compared to its TL augmented version which was pre-trained on the *pre-training dataset*. Fig. 3.9 presents the accuracies obtained for the classifiers with different number of repetitions employed for training. The difference in accuracy is considered statistically significant by the one-tail Wilcoxon Signed rank test for all cycles of training. Note that, similar, statistically significant results were obtained for the raw-based and spectrogram-based ConvNets.

3.12 Real-Time Classification and Medium Term Performances (case study)

This last experiment section proposes a use-case study of the online (i.e. real-time) performance of the classifier over a period of 14 days for eight able-bodied participants. In previous literature, it has been shown that, when no re-calibration occur, the performance of a classifier degrades over time due to the non-stationary property of sEMG signals [48]. The main goal of this use-case experiment is to evaluate if users are able to self-adapt and improve the way they



Figure 3.9 – Classification accuracy of the CWT-based ConvNets on the *NinaPro DB5* with respect to the number of repetitions employed during training. The pre-training was done using the *pre-training dataset*. Training and testing only considered the 11 gestures from the *NinaPro DB5* not included in the pre-training. The error bars correspond to the STD across all ten participants.

perform gestures based on visual feedback from complex classifiers (e.g. *CWT+TL*), thus reducing the expected classification degradation.

To achieve this, each participant recorded a training set as described in Sec. 3.7. Then, over the next fourteen days, a daily *session* was recorded based on the participant’s availability. A *session* consisted of holding a set of 30 randomly selected gestures (among the seven shown in Fig. 3.1) for ten seconds each, resulting in five minutes of continuous sEMG data. Note that to be more realistic, the participants began by placing the armband themselves, leading to slight armband position variations between sessions.

The eight participants were randomly separated into two equal groups. The first group, referred to as the *Feedback* group, received real-time feedback on the gesture predicted by the classifier in the form of text displayed on a computer screen. The second group, referred to as the *Without Feedback* group, did not receive classifier feedback. The classifier employed in this experiment is the *CWT+TL*, as it was the best performing classifier tested on the *Evaluation Dataset*. Because the transitions are computer-specified, there is a latency between a new requested gesture and the participant’s reaction. To reduce the impact of this phenomenon, the data from the first second after a new requested gesture is ignored from this section results. The number of data points generated by a single participant varies between 10 and 16 depending on the participant’s availability during the experiment period.

As it can be observed in Fig. 3.10, while the *Without Feedback* group did experience accuracy degradation over the 14 days, the *Feedback* group was seemingly able to counteract this degradation. Note that, the average accuracy across all participants for the first recording

session was 95.42%.

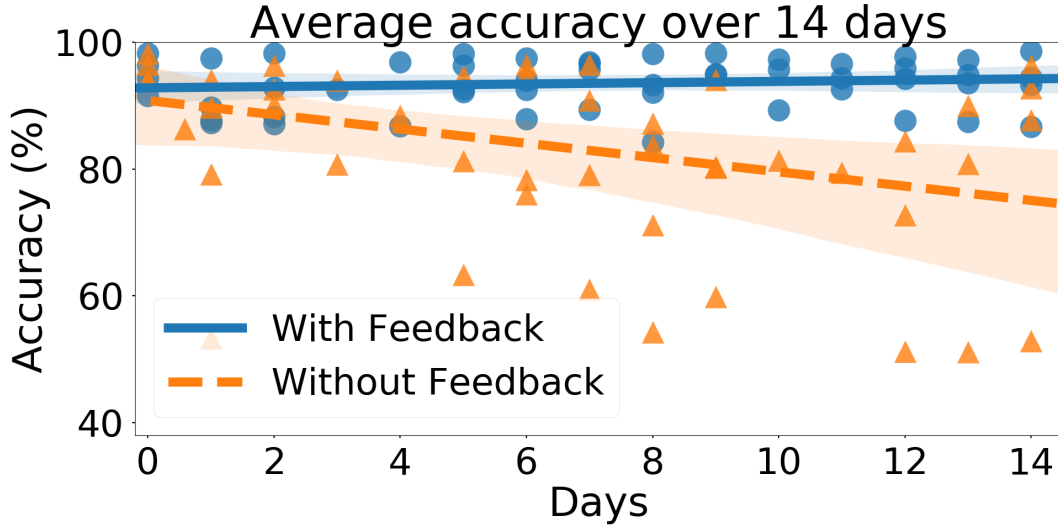


Figure 3.10 – Average accuracy over 14 days without recalibration of the CWT+TL ConvNet. The blue circles represent data from the *Feedback* group whereas the orange triangles represent data from the *Without Feedback* group. The translucent bands around the linear regressions represent the confidence interval (95%) estimated by bootstrap.

Many participants reported experiencing muscular fatigue during the recording of both this experiment and the *evaluation dataset*. As such, in an effort to quantify the impact of muscle fatigue on the classifier’s performance, the average accuracy of the eight participants over the five minute session is computed as a function of time. As can be observed from the positive slope of the linear regression presented in Fig. 3.11, muscle fatigue, does not seem to negatively affect the proposed ConvNet’s accuracy.

3.13 Discussion

Table 3.1 and Table 3.3 show that, in all cases the TL augmented ConvNets significantly outperformed their non-augmented versions, regardless of the number of training cycles. As expected, reducing the amount of training cycles systematically degraded the performances of all tested methods (see Table 3.1, 3.2, 3.3, 3.4 and Fig. 3.9), with the non-TL ConvNets being the most affected on the *Myo Dataset*. This is likely due to overfitting that stems from the small size of the dataset. However, it is worth noting that, when using a single cycle of training, augmenting the ConvNets with the proposed TL scheme significantly improves their accuracies. In fact, with this addition, the accuracies of the ConvNets become the highest of all methods on both tested datasets. Overall, the proposed TL-augmented ConvNets were competitive with the current state-of-the-art, with the *TL augmented CWT-based ConvNet* achieving a higher average accuracy than the traditional sEMG classification technique on both datasets for all training cycles. It is also noteworthy that while the *raw+TL* ConvNet was

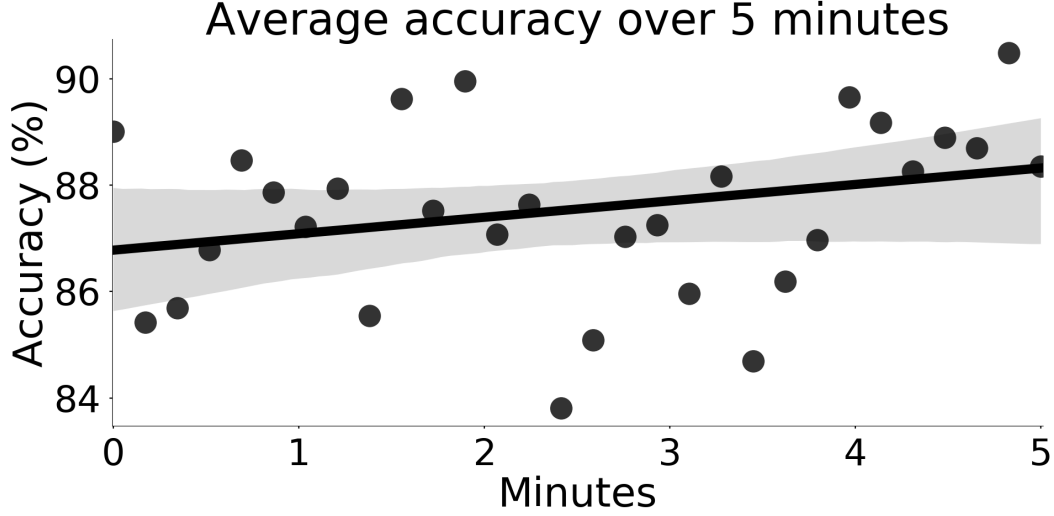


Figure 3.11 – The average accuracy of the eight participants over all the five minute sessions recorded to evaluate the effect of muscle fatigue on the classifier performance. During each session of the experiment, participants were asked to hold a total of 30 random gestures for ten seconds each. As such, a dot represents the average accuracy across all participants over one of the ten second periods. The translucent bands around the linear regression represent the confidence intervals (95%) estimated by bootstrap.

the worst amongst the TL augmented ConvNet on the *Myo Dataset*, it achieved the highest accuracy on the *NinaPro DB5*. Furthermore, the TL method outperformed the non-augmented ConvNets on the out-of-sample experiment. The difference in accuracy of the two methods was deemed significant by the Wilcoxon Signed Rank Test ($p < 0.05$) for all training repetitions. This suggests that the proposed TL algorithm enables the network to learn features that can generalize not only across participants but also for never-seen-before gestures. As such, the weights learned from the *pre-training dataset* can easily be re-used for other work that employs the Myo Armband with different gestures.

While in this paper, the proposed *source* and *second network* were almost identical they are performing different tasks (see Sec. 3.10.3). As such further differentiation of both networks might lead to increased performance. At first glance, the element-wise summation between the *source* and *second network* might seem to impose a strong constraint on the architecture of the two networks. However, one could replace the learned scalar layers in the *target network* by convolutions or fully connected layers to bridge the dimensionality gap between potentially vastly different *source* and *second* networks.

Additionally, a difference in the average accuracy between the real-time experiment (Sec. 3.12) and the *Evaluation Dataset* (Sec. 3.11.1) was observed (95.42% vs 98.31% respectively). This is likely due to the reaction delay of the participants, but more importantly to the transition between gestures. These transitions are not part of the training dataset, because they are too time consuming to record as the number of possible transitions equals $n^2 - n$ where n is

the number of gestures. Consequently, it is expected that the classifiers predictive power on transition data is poor in these circumstances. As such, being able to accurately detect such transitions in an unsupervised way might have a greater impact on the system’s responsiveness than simply reducing the window size. This and the aforementioned point will be investigated in future works.

The main limitation of this study is the absence of tests with amputees. Additionally, the issue of electrode shifts has not been explicitly studied and the variability introduced by various limb positions was not considered when recording the dataset. A limitation of the proposed TL scheme is its difficulty to adapt when the new user cannot wear the same amount of electrodes as the group used for pre-training. This is because changing the number of channels changes the representation of the phenomena (i.e. muscle contraction) being fed to the algorithm. The most straightforward way of addressing this would be to numerically remove the relevant channels from the dataset used for pre-training. Then re-running the proposed TL algorithm on an architecture adapted to the new representation fed as input. Another solution is to consider the EMG channels in a similar way as color channels in image. This type of architecture seems, however, to perform worse than the ones presented in this paper (see Appendix 3.F).

3.14 Conclusion

This paper presents three novel ConvNet architectures that were shown to be competitive with current sEMG-based classifiers. Moreover, this work presents a new TL scheme that systematically and significantly enhances the performances of the tested ConvNets. On the newly proposed *evaluation* dataset, the TL augmented ConvNet achieves an average accuracy of 98.31% over 17 participants. Furthermore, on the *NinaPro DB5* dataset (18 hand/wrist gestures), the proposed classifier achieved an average accuracy of 68.98% over 10 participants on a single Myo Armband. This dataset showed that the proposed TL algorithm learns sufficiently general features to significantly enhance the performance of ConvNets on out-of-sample gestures. Showing that deep learning algorithms can be efficiently trained, within the inherent constraints of sEMG-based hand gesture recognition, offers exciting new research avenues for this field.

Future works will focus on adapting and testing the proposed TL algorithm on upper-extremity amputees. This will provide additional challenges due to the greater muscle variability across amputees and the decrease in classification accuracy compared to able-bodied participants [6]. Additionally, tests for the application of the proposed TL algorithm for inter-session classification will be conducted as to be able to leverage labeled information for long-term classification.

3.15 Acknowledgements

This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), [401220434], the Institut de recherche Robert-Sauvé en santé et en sécurité du travail (IRSST), the Fondation Famille Choquette, and the Research Council of Norway through its Centres of Excellence scheme, project number 262762.

3.16 Bibliography

- [1] Paul S Addison. Wavelet transforms and the ecg: a review. *Physiological measurement*, 26(5):R155, 2005.
- [2] Md Rezwanul Ahsan, Muhammad I Ibrahimy, Othman O Khalifa, et al. Emg signal classification for human computer interaction: a review. *European Journal of Scientific Research*, 33(3):480–501, 2009.
- [3] Rami Al-Rfou, Guillaume Alain, Amjad Almahairi, Christof Angermueller, Dzmitry Bahdanau, et al. Theano: A python framework for fast computation of mathematical expressions. *arXiv preprint arXiv:1605.02688*, 2016.
- [4] Ulysse Côté Allard, François Nougrou, Cheikh Latyr Fall, Philippe Giguère, Clément Gosselin, François Laviolette, and Benoit Gosselin. A convolutional neural network for robotic arm guidance using semg based frequency-features. In *Intelligent Robots and Systems (IROS)*, pages 2464–2470. IEEE, 2016.
- [5] Manfredo Atzori, Matteo Cognolato, and Henning 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, 2016.
- [6] Manfredo Atzori, Arjan Gijsberts, Claudio Castellini, Barbara Caputo, Anne-Gabrielle Mit-taz Hager, Simone Elsig, Giorgio Giatsidis, Franco Bassetto, and Henning Müller. Elec-tromyography data for non-invasive naturally-controlled robotic hand prostheses. *Scientific data*, 1:140053, 2014.
- [7] Manfredo Atzori, Arjan Gijsberts, Claudio Castellini, Barbara Caputo, Anne-Gabrielle Mit-taz Hager, Simone Elsig, Giorgio Giatsidis, Franco Bassetto, and Henning Müller. Elec-tromyography data for non-invasive naturally-controlled robotic hand prostheses. *Scientific data*, 1:140053, 2014.
- [8] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt. Sequential deep learning for human action recognition. In *International Workshop on Human Behavior Understanding*, pages 29–39. Springer, 2011.

- [9] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. *ICML Unsupervised and Transfer Learning*, 27:17–36, 2012.
- [10] Claudio Castellini, Angelo Emanuele Fiorilla, and Giulio Sandini. Multi-subject/daily-life activity emg-based control of mechanical hands. *Journal of neuroengineering and rehabilitation*, 6(1):41, 2009.
- [11] Lukas Cavigelli, Michele Magno, and Luca Benini. Accelerating real-time embedded scene labeling with convolutional networks. In *Design Automation Conference (DAC), 2015 52nd ACM/EDAC/IEEE*, pages 1–6. IEEE, 2015.
- [12] Rita Chattopadhyay, Narayanan Chatapuram Krishnan, and Sethuraman Panchanathan. Topology preserving domain adaptation for addressing subject based variability in semg signal. In *AAAI Spring Symposium: Computational Physiology*, pages 4–9, 2011.
- [13] Yiran Chen, Hai Helen Li, Chunpeng Wu, Chang Song, Sicheng Li, Chuhan Min, Hsin-Pai Cheng, Wei Wen, and Xiaoxiao Liu. Neuromorphic computing’s yesterday, today, and tomorrow—an evolutonal view. *Integration, the VLSI Journal*, 2017.
- [14] Yu-Hsin Chen, Tushar Krishna, Joel S Emer, and Vivienne Sze. Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. *IEEE Journal of Solid-State Circuits*, 52(1):127–138, 2017.
- [15] Ulysse Côté-Allard, Cheikh Latyr Fall, Alexandre Campeau-Lecours, Clément Gosselin, François Laviolette, and Benoit Gosselin. Transfer learning for semg hand gestures recognition using convolutional neural networks. In *Systems, Man, and Cybernetics, 2017 IEEE International Conference on (in press)*. IEEE, 2017.
- [16] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006.
- [17] Sander Dieleman, Jan Schlüter, Colin Raffel, Eben Olson, Søren Kaae Sønderby, et al. Lasagne: First release., August 2015.
- [18] Sander Dieleman, Aron van den Oord, Ira Korshunova, Jeroen Burms, Jonas Degraeve, Lionel Pigou, and Pieter Buteneers. Classifying plankton with deep neural networks. *URL <http://benanne.github.io/2015/03/17/plankton.html>*, 2015.
- [19] Yu Du, Wenguang Jin, Wentao Wei, Yu Hu, and Weidong Geng. Surface emg-based inter-session gesture recognition enhanced by deep domain adaptation. *Sensors*, 17(3):458, 2017.
- [20] Kevin Englehart, B Hudgin, and Philip A Parker. A wavelet-based continuous classification scheme for multifunction myoelectric control. *IEEE Transactions on Biomedical Engineering*, 48(3):302–311, 2001.

- [21] Kevin Englehart and Bernard Hudgins. A robust, real-time control scheme for multifunction myoelectric control. *IEEE transactions on biomedical engineering*, 50(7):848–854, 2003.
- [22] Kevin Englehart, Bernard Hudgins, Philip A Parker, and Maryhelen Stevenson. Classification of the myoelectric signal using time-frequency based representations. *Medical engineering & physics*, 21(6):431–438, 1999.
- [23] Todd R Farrell and Richard F Weir. The optimal controller delay for myoelectric prostheses. *IEEE Transactions on neural systems and rehabilitation engineering*, 15(1):111–118, 2007.
- [24] Oliver Faust, U Rajendra Acharya, Hojjat Adeli, and Amir Adeli. Wavelet-based eeg processing for computer-aided seizure detection and epilepsy diagnosis. *Seizure*, 26:56–64, 2015.
- [25] Dennis Gabor. Theory of communication. part 1: The analysis of information. *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, 93(26):429–441, 1946.
- [26] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.
- [27] Weidong Geng, Yu Du, Wenguang Jin, Wentao Wei, Yu Hu, and Jiajun Li. Gesture recognition by instantaneous surface emg images. *Scientific reports*, 6:36571, 2016.
- [28] Rafael C Gonzalez. *Digital image processing*. 1977.
- [29] Amara Graps. An introduction to wavelets. *IEEE computational science and engineering*, 2(2):50–61, 1995.
- [30] Valentina Gregori, Arjan Gijsberts, and Barbara Caputo. Adaptive learning to speed-up control of prosthetic hands: A few things everybody should know. In *Rehabilitation Robotics (ICORR), 2017 International Conference on*, pages 1130–1135. IEEE, 2017.
- [31] Song Han, Xingyu Liu, Huizi Mao, Jing Pu, Ardavan Pedram, Mark A Horowitz, and William J Dally. Eie: efficient inference engine on compressed deep neural network. In *Proceedings of the 43rd International Symposium on Computer Architecture*, pages 243–254. IEEE Press, 2016.
- [32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.
- [33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

- [34] Bo Hjorth. Eeg analysis based on time domain properties. *Electroencephalography and clinical neurophysiology*, 29(3):306–310, 1970.
- [35] John H Hollman, Jeffrey M Hohl, Jordan L Kraft, Jeffrey D Strauss, and Katie J Traver. Does the fast fourier transformation window length affect the slope of an electromyogram’s median frequency plot during a fatiguing isometric contraction? *Gait & posture*, 38(1):161–164, 2013.
- [36] Bernard Hudgins, Philip Parker, and Robert N Scott. A new strategy for multifunction myoelectric control. *IEEE Transactions on Biomedical Engineering*, 40(1):82–94, 1993.
- [37] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.
- [38] Adham R Ismail and Shihab S Asfour. Continuous wavelet transform application to emg signals during human gait. In *Signals, Systems & Computers, 1998. Conference Record of the Thirty-Second Asilomar Conference on*, volume 1, pages 325–329. IEEE, 1998.
- [39] Wen-Juh Kang, Jiue-Rou Shiu, Cheng-Kung Cheng, Jin-Shin Lai, Hen-Wai Tsao, and Te-Son Kuo. The application of cepstral coefficients and maximum likelihood method in emg pattern recognition [movements classification]. *IEEE Transactions on Biomedical Engineering*, 42(8):777–785, 1995.
- [40] Stefan Karlsson and Björn Gerdle. Mean frequency and signal amplitude of the surface emg of the quadriceps muscles increase with increasing torque—a study using the continuous wavelet transform. *Journal of electromyography and kinesiology*, 11(2):131–140, 2001.
- [41] Stefan Karlsson, Jun Yu, and Metin Akay. Time-frequency analysis of myoelectric signals during dynamic contractions: a comparative study. *IEEE transactions on Biomedical Engineering*, 47(2):228–238, 2000.
- [42] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.
- [43] Rami N Khushaba. Correlation analysis of electromyogram signals for multiuser myoelectric interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(4):745–755, 2014.
- [44] Rami N Khushaba and Sarath Kodagoda. Electromyogram (emg) feature reduction using mutual components analysis for multifunction prosthetic fingers control. In *Control Automation Robotics & Vision (ICARCV), 2012 12th International Conference on*, pages 1534–1539. IEEE, 2012.

- [45] Rami N Khushaba, Sarath Kodagoda, Maen Takruri, and Gamini Dissanayake. Toward improved control of prosthetic fingers using surface electromyogram (emg) signals. *Expert Systems with Applications*, 39(12):10731–10738, 2012.
- [46] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [47] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [48] Jianwei Liu, Xinjun Sheng, Dingguo Zhang, Jiayuan He, and Xiangyang Zhu. Reduced daily recalibration of myoelectric prosthesis classifiers based on domain adaptation. *IEEE journal of biomedical and health informatics*, 20(1):166–176, 2016.
- [49] Marie-Françoise Lucas, Adrien Gaufriaux, Sylvain Pascual, Christian Doncarli, and Dario Farina. Multi-channel surface emg classification using support vector machines and signal-based wavelet optimization. *Biomedical Signal Processing and Control*, 3(2):169–174, 2008.
- [50] Roberto Merletti and P Di Torino. Standards for reporting emg data. *J Electromyogr Kinesiol*, 9(1):3–4, 1999.
- [51] M Mouzé-Amady and F Horwat. Evaluation of hjorth parameters in forearm surface emg analysis during an occupational repetitive task. *Electroencephalography and Clinical Neurophysiology/Electromyography and Motor Control*, 101(2):181–183, 1996.
- [52] Eriko Nurvitadhi, Ganesh Venkatesh, Jaewoong Sim, Debbie Marr, Randy Huang, Jason Ong Gee Hock, Yeong Tat Liew, Krishnan Srivatsan, Duncan Moss, Suchit Subhaschandra, et al. Can fpgas beat gpus in accelerating next-generation deep neural networks? In *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pages 5–14. ACM, 2017.
- [53] Francesco Orabona, Claudio Castellini, Barbara Caputo, Angelo Emanuele Fiorilla, and Giulio Sandini. Model adaptation with least-squares svm for adaptive hand prosthetics. In *Robotics and Automation, 2009. ICRA’09. IEEE International Conference on*, pages 2897–2903. IEEE, 2009.
- [54] Mohammadreza Asghari Oskoei and Huosheng Hu. Myoelectric control systems— a survey. *Biomedical Signal Processing and Control*, 2(4):275–294, 2007.
- [55] Novi Patricia, Tatiana Tommasit, and Barbara Caputo. Multi-source adaptive learning for fast control of prosthetics hand. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 2769–2774. IEEE, 2014.

- [56] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [57] Bart Peerdeman, Daphne Boere, Heidi Witteveen, Rianne Huis Veld, Hermie Hermens, Stefano Stramigioli, et al. Myoelectric forearm prostheses: State of the art from a user-centered perspective. *Journal of Rehabilitation Research & Development*, 48(6):719, 2011.
- [58] Angkoon Phinyomark, S Hirunviriya, C Limsakul, and P Phukpattaranont. Evaluation of emg feature extraction for hand movement recognition based on euclidean distance and standard deviation. In *Electrical Engineering/Electronics Computer Telecommunications and Information Technology (ECTI-CON), 2010 International Conference on*, pages 856–860. IEEE, 2010.
- [59] Angkoon Phinyomark, Rami N Khushaba, and Erik Scheme. Feature extraction and selection for myoelectric control based on wearable emg sensors. *Sensors*, 18(5):1615, 2018.
- [60] Angkoon Phinyomark, Pornchai Phukpattaranont, and Chusak Limsakul. Feature reduction and selection for emg signal classification. *Expert Systems with Applications*, 39(8):7420–7431, 2012.
- [61] Angkoon Phinyomark, Franck Quaine, Sylvie Charbonnier, Christine Serviere, Franck Tarpin-Bernard, and Yann Laurillau. Emg feature evaluation for improving myoelectric pattern recognition robustness. *Expert Systems with Applications*, 40(12):4832–4840, 2013.
- [62] Angkoon Phinyomark and Erik Scheme. Emg pattern recognition in the era of big data and deep learning. *Big Data and Cognitive Computing*, 2(3):21, 2018.
- [63] Stefano Pizzolato, Luca Tagliapietra, Matteo Cognolato, Monica Reggiani, Henning Müller, and Manfredo Atzori. Comparison of six electromyography acquisition setups on hand movement classification tasks. *PloS one*, 12(10):e0186132, 2017.
- [64] Yahya T Qassim, Tim RH Cutmore, and David D Rowlands. Optimized fpga based continuous wavelet transform. *Computers & Electrical Engineering*, 49:84–94, 2016.
- [65] M BI Reaz, MS Hussain, and Faisal Mohd-Yasin. Techniques of emg signal analysis: detection, processing, classification and applications. *Biological procedures online*, 8(1):11–35, 2006.
- [66] Raymond Reynolds and Martin Lakie. Postmovement changes in the frequency and amplitude of physiological tremor despite unchanged neural output. *Journal of neurophysiology*, 104(4):2020–2023, 2010.

- [67] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.
- [68] Ali-Akbar Samadani and Dana Kulic. Hand gesture recognition based on surface electromyography. In *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, pages 4196–4199. IEEE, 2014.
- [69] Lauren H Smith, Levi J Hargrove, Blair A Lock, and Todd A Kuiken. Determining the optimal window length for pattern recognition-based myoelectric control: balancing the competing effects of classification error and controller delay. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 19(2):186–192, 2011.
- [70] D.F. Stegeman and B.G. Lapatki B.U. Kleine. High-density surface emg: Techniques and applications at a motor unit level. *Biocybernetics and Biomedical Engineering*, 32(3), 2012.
- [71] Michal Teplan et al. Fundamentals of eeg measurement. *Measurement science review*, 2(2):1–11, 2002.
- [72] C Toledo, R Muñoz, and L Leija. semg signal detector using discrete wavelet transform. In *Health Care Exchanges (PAHCE), 2012 Pan American*, pages 62–65. IEEE, 2012.
- [73] Tatiana Tommasi, Francesco Orabona, Claudio Castellini, and Barbara Caputo. Improving control of dexterous hand prostheses using adaptive learning. *IEEE Transactions on Robotics*, 29(1):207–219, 2013.
- [74] Ludovic Trottier, Philippe Giguère, and Brahim Chaib-draa. Parametric exponential linear unit for deep convolutional neural networks. *arXiv preprint arXiv:1605.09332*, 2016.
- [75] G Tsenov, AH Zeghib, F Palis, N Shoylev, and V Mladenov. Neural networks for online classification of hand and finger movements using surface emg signals. In *Neural Network Applications in Electrical Engineering, 2006. NEUREL 2006. 8th Seminar on*, pages 167–171. IEEE, 2006.
- [76] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80–83, 1945.
- [77] Jiaxiang Wu, Cong Leng, Yuhang Wang, Qinghao Hu, and Jian Cheng. Quantized convolutional neural networks for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4820–4828, 2016.
- [78] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

- [79] Mahyar Zardoshti-Kermani, Bruce C Wheeler, Kambiz Badie, and Reza M Hashemi. Emg feature evaluation for movement control of upper extremity prostheses. *IEEE Transactions on Rehabilitation Engineering*, 3(4):324–333, 1995.
- [80] Nikola Žarić, Srdjan Stanković, and Zdravko Uskoković. Hardware realization of the robust time–frequency distributions. *annals of telecommunications-Annales des télécommunications*, 69(5-6):309–320, 2014.
- [81] Daohui Zhang, Xingang Zhao, Jianda Han, and Yiwen Zhao. A comparative study on pca and lda based emg pattern recognition for anthropomorphic robotic hand. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 4850–4855. IEEE, 2014.
- [82] Xu Zhang and Ping Zhou. Sample entropy analysis of surface emg for improved muscle activity onset detection against spurious background spikes. *Journal of Electromyography and Kinesiology*, 22(6):901–907, 2012.
- [83] Muhammad Zia ur Rehman, Asim Waris, Syed Gilani, Mads Jochumsen, Imran Niazi, Mohsin Jamil, Dario Farina, and Ernest Kamavuako. Multiday emg-based classification of hand motions with deep learning techniques. *Sensors*, 18(8):2497, 2018.

3.A Data Augmentation

The idea behind data augmentation is to augment the size of the training set, with the objective of achieving better generalization. This is generally accomplished by adding realistic noise to the training data, which tends to induce a robustness to noise into the learned model. In many cases, this has been shown to lead to better generalization [47, 18]. In this paper’s context, data augmentation techniques can thus be viewed as part of the solution to reduce the overfitting from training a ConvNet on a small dataset. When adding noise to the data, it is important to ensure that the noise does not change the label of the examples. Hence, for image datasets, the most common and often successful techniques have relied on affine transformations [18].

Unfortunately, for sEMG signals, most of these techniques are unsuitable and cannot be applied directly. As such, specific data augmentation techniques must be employed. In this work, five data augmentation techniques are tested on the *pre-training dataset* as they are part of the architecture building process. Note that this comparison was made with the ConvNet architecture presented in [15], which takes as input a set of eight spectrograms (one for each channel of the Myo Armband).

Examples are constructed by applying non-overlapping windows of 260ms. This non-augmented dataset is referred to as the *Baseline*. Consequently, an intuitive way of augmenting sEMG data is to apply overlapping windows (i.e. temporal translation) when building the examples. A major advantage of this technique within the context of sEMG signals - and time signals in general - is that it does not create any synthetic examples in the dataset compared to the affine transformation employed with images. Furthermore, with careful construction of the dataset, no new mislabeling occurs. In this work, this technique will be referred to as *Sliding Window* augmentation.

Second, the effect of muscle fatigue on the frequency response of muscles fibers [35] can be emulated, by altering the calculated spectrogram. The idea is to reduce the median frequency of a channel with a certain probability, by systematically redistributing part of the power of a frequency bin to an adjacent lower frequency one and so on. This was done in order to approximate the effect of muscle fatigue on the frequency response of muscle fibers [35]. In this work, this technique will be referred to as *Muscle Fatigue* augmentation.

The third data augmentation technique employed aims at emulating electrode displacement on the skin. This is of particular interest, as the dataset was recorded with a dry electrode armband, for which this kind of noise is to be expected. The data augmentation technique consists of shifting part of the power spectrum magnitude from one channel to the next. In other words, part of the signal energy from each channel is sent to an adjacent channel emulating electrode displacement on the skin. In this work, this approach will be referred to as *Electrode Displacement* augmentation.

For completeness, a fourth data augmentation technique which was proposed in a paper [5] employing a ConvNet for sEMG gestures classification is also considered. The approach consists of adding a white Gaussian noise to the signal, with a signal-to-noise ratio of 25. This technique will be referred to as *Gaussian Noise* augmentation.

Finally, the application of all these data augmentation methods simultaneously is referred to as the *Aggregated Augmentation* technique.

Data from these augmentation techniques will be generated from the *pre-training dataset*. The data will be generated on the first two cycles, which will serve as the training set. The third cycle will be the validation set and the test set will be the fourth cycle. All augmentation techniques will generate double the amount of training examples compared to the baseline dataset.

Table 3.5 reports the average test set accuracy for the 19 participants over 20 runs. In this appendix, the one-tail Wilcoxon signed rank test with Bonferroni correction is applied to compare the data augmentation methods with the baseline. The results of the statistical test are summarized in Table 3.5. The only techniques that produce significantly different results from the *Baseline* is the *Sliding Window* (improves accuracy). As such, as described in Sec. 3.7.1 the only data augmentation technique employed in this work is the sliding windows.

Table 3.5 – Comparison of the five data augmentation techniques proposed.

	Baseline	Gaussian Noise	Muscle Fatigue	Electrode Displacement	Sliding Window	Aggregated Augmentation
Accuracy	95.62%	93.33%	95.75%	95.80%	96.14%	95.37%
STD	5.18%	7.12%	5.07%	4.91%	4.93%	5.27%
Rank	4	6	3	2	1	5
H_0 (p-value)	-	1	1	1	0 (0.00542)	1

The values reported are the average accuracies for the 19 participants over 20 runs. The Wilcoxon signed rank test is applied to compare the training of the ConvNet with and without one of the five data augmentation techniques. The null hypothesis is accepted when $H_0 = 1$ and rejected when $H_0 = 0$ (with $p = 0.05$). As the *Baseline* is employed to perform multiple comparison, Bonferroni correction is applied. As such, to obtain a global p-value of 0.05, a per-comparison p-value of 0.00833 is employed.

3.B Deep Learning on Embedded Systems and real-time classification

Within the context of sEMG-based gesture recognition, an important consideration is the feasibility of implementing the proposed ConvNets on embedded systems. As such, important efforts were deployed when designing the ConvNets architecture to ensure attainable implementation on currently available embedded systems. With the recent advent of deep learning, hardware systems particularly well suited for neural networks training/inference have been made commercially available. Graphics processing units (GPUs) such as the Nvidia Volta GV100 from *Nvidia* (50 GFLOPs/s/W) [13], field programmable gate arrays (FPGAs) such as

the Stratix 10 from *Altera* (80 GFLOPs/s/W) [52] and mobile system-on-chips (SoCs) such as the Nvidia Tegra from *Nvidia* (100 GFLOPs/s/W) [11], are commercially available platforms that target the need for portable, computationally efficient and low-power systems for deep learning inference. Additionally, dedicated Application-Specific Integrated Circuits (ASICs) have arisen from research projects capable of processing ConvNet orders of magnitudes bigger than the ones proposed in this paper at a throughput of 35 frames/s at 278mW [14]. Pruning and quantizing network architectures are further ways to reduce the computational cost when performing inference with minimal impact on accuracy [31, 77].

Efficient CWT implementation employing the Mexican Hat wavelet has already been explored for embedded platforms [64]. These implementations are able to compute the CWT of larger input sizes than those required in this work in less than 1ms. Similarly, in [80], a robust time-frequency distribution estimation suitable for fast and accurate spectrogram computation is proposed. To generate a classification, the proposed CNN-Spectrogram and CNN-CWT architectures (including the TL scheme proposed in Sec. 6.8) require approximately 14 728 000 and 2 274 000 floating point operations (FLOPs) respectively. Considering a 40ms inference processing delay, hardware platforms of 3.5 and 0.5 GFLOPs/s/W will be suitable to implement a 100mW embedded system for sEMG classification. As such, adopting hardware-implementation approaches, along with state-of-the-art network compression techniques will lead to a power-consumption lower than 100mW for the proposed architectures, suitable for wearable applications.

Note that currently, without optimization, it takes 21.42ms to calculate the CWT and classify one example with the CWT-based ConvNet compared to 2.94ms and 3.70ms for the spectrogram and raw EMG Convnet respectively. Applying the proposed TL algorithm add an additional 0.57ms, 0.90ms and 0.14ms to the computation for the CWT, spectrogram and raw EMG-based ConvNet respectively. These timing results were obtained by averaging the pre-processing and classifying time of the same 5309 examples across all methods. The gpu employed was a GeForce GTX 980M.

3.C Feature Engineering

This section presents the features employed in this work. Features can be regrouped into different types, mainly: time, frequency and time-frequency domains. Unless specified otherwise, features are calculated by dividing the signal x into overlapping windows of length L . The k th element of the i th window then corresponds to $x_{i,k}$.

3.C.1 Time Domain Features

Mean Absolute Value (MAV)

[21]: A feature returning the mean of a fully-rectified signal.

$$\text{MAV}(x_i) = \frac{1}{L} \sum_{k=1}^L |x_{i,k}| \quad (3.1)$$

Slope Sign Changes (SSC) [21]

A feature that measures the frequency at which the sign of the signal slope changes. Given three consecutive samples $x_{i,k-1}$, $x_{i,k}$, $x_{i,k+1}$, the value of SSC is incremented by one if:

$$(x_{i,k} - x_{i,k-1}) * (x_{i,k} - x_{i,k+1}) \geq \epsilon \quad (3.2)$$

Where $\epsilon \geq 0$, is employed as a threshold to reduce the impact of noise on this feature.

Zero Crossing (ZC) [21]

A feature that counts the frequency at which the signal passes through zero. A threshold $\epsilon \geq 0$ is utilized to lessen the impact of noise. The value of this feature is incremented by one whenever the following condition is satisfied:

$$(|x_{i,k} - x_{i,k+1}| \geq \epsilon) \wedge (\text{sgn}(x_{i,k}, x_{i,k+1}) \Leftrightarrow \text{False}) \quad (3.3)$$

Where $\text{sgn}(a, b)$ returns true if a and b (two real numbers) have the same sign and false otherwise. Note that depending on the slope of the signal and the selected ϵ , the zero crossing point might not be detected.

Waveform Length (WL) [21]

A feature that offers a simple characterization of the signal's waveform. It is calculated as follows:

$$\text{WL}(x_i) = \sum_{k=1}^L |x_{i,k} - x_{i,k-1}| \quad (3.4)$$

Skewness

The Skewness is the third central moment of a distribution which measures the overall asymmetry of a distribution. It is calculated as follows:

$$\text{Skewness}(x_i) = \frac{1}{L} \sum_{k=1}^L \left(\frac{x_{i,k} - \bar{x}_i}{\sigma} \right)^3 \quad (3.5)$$

Where σ is the standard deviation:

Root Mean Square (RMS) [58]

This feature, also known as the quadratic mean, is closely related to the standard deviation as both are equal when the mean of the signal is zero. RMS is calculated as follows:

$$\text{RMS}(x_i) = \sqrt{\frac{1}{L} \sum_{k=1}^L x_{i,k}^2} \quad (3.6)$$

Hjorth Parameters [34]

Hjorth parameters are a set of three features originally developed for characterizing electroencephalography signals and then successfully applied to sEMG signal recognition [51, 44]. *Hjorth Activity Parameter* can be thought of as the surface of the power spectrum in the frequency domain and corresponds to the variance of the signal calculated as follows:

$$\text{Activity}(x_i) = \frac{1}{L} \sum_{k=1}^L (x_{i,k} - \bar{x}_i)^2 \quad (3.7)$$

Where \bar{x}_i is the mean of the signal for the i th window. *Hjorth Mobility Parameter* is a representation of the mean frequency of the signal and is calculated as follows:

$$\text{Mobility}(x_i) = \sqrt{\frac{\text{Activity}(x'_i)}{\text{Activity}(x_i)}} \quad (3.8)$$

Where x'_i is the first derivative in respect to time of the signal for the i th window. Similarly, the *Hjorth Complexity Parameter*, which represents the change in frequency, is calculated as follows:

$$\text{Complexity}(x_i) = \frac{\text{Mobility}(x'_i)}{\text{Mobility}(x_i)} \quad (3.9)$$

Integrated EMG (IEMG)

[58]: A feature returning the sum of the fully-rectified signal.

$$\text{IEMG}(x_i) = \sum_{k=1}^L |x_{i,k}| \quad (3.10)$$

Autoregression Coefficient (AR)

[60] An autoregressive model tries to predict future data, based on a weighted average of the previous data. This model characterizes each sample of the signal as a linear combination of the previous sample with an added white noise. The number of coefficients calculated is a trade-off between computational complexity and predictive power. The model is defined as follows:

$$x_{i,k} = \sum_{j=1}^P \rho_j x_{i,k-j} + \epsilon_t \quad (3.11)$$

Where P is the model order, ρ_j is the j th coefficient of the model and ϵ_t is the residual white noise.

Sample Entropy (SampEn)

[82] Entropy measures the complexity and randomness of a system. Sample Entropy is a method which allows entropy estimation.

$$\text{SampEn}(x_i, m, r) = -\ln \left(\frac{A^m(r)}{B^m(r)} \right) \quad (3.12)$$

EMG Histogram (HIST) [79]

When a muscle is in contraction, the EMG signal deviates from its baseline. The idea behind HIST is to quantify the frequency at which this deviation occurs for different amplitude levels. HIST is calculated by determining a symmetric amplitude range centered around the baseline. This range is then separated into n bins of equal length (n is a hyperparameter). The HIST is obtained by counting how often the amplitude of the signal falls within each bin's boundaries.

3.C.2 Frequency Domain Features

Cepstral Coefficient [39, 60]

The cepstrum of a signal is the inverse Fourier transform of the log power spectrum magnitude of the signal. Like the AR, the coefficients of the cepstral coefficients are employed as features. They can be directly derived from AR as follows:

$$c_1 = -a_1 \quad (3.13)$$

$$c_i = -a_i - \sum_{n=1}^{i-1} \left(1 - \frac{n}{i}\right) a_n c_{i-n}, \text{ with } 1 < i \leq P \quad (3.14)$$

Marginal Discrete Wavelet Transform (mDWT) [49]

The mDWT is a feature that removes the time-information from the discrete wavelet transform to be insensitive to wavelet time instants. The feature instead calculates the cumulative energy of each level of the decomposition. The computation of the mDWT for each channel is implemented as follow in [63] (See Algorithm 1).

Algorithm 1 mDWT pseudo-code

```

1: procedure mDWT
2:    $wav \leftarrow db7$ 
3:    $level \leftarrow 3$ 
4:    $coefficients \leftarrow wavDec(x, level, wav)$ 
5:    $N \leftarrow length(coefficients)$ 
6:    $SMax \leftarrow \log_2(N)$ 
7:    $Mxk \leftarrow []$ 
8:   for  $s=1, \dots, SMax$  do
9:      $CMax \leftarrow \frac{N}{2^s} - 1$ 
10:     $val \leftarrow \sum_{u=0}^{CMax} |coefficients[u]|$ 
11:     $Mxk.append(val)$ 
return  $Mxk$ 

```

Where x is the 1-d signal from which to calculate the mDWT and $wavDec$ is a function that calculates the wavelet decomposition of a vector at level n using the wavelet wav . The coefficients are returned in a 1-d vector with the *Approximation Coefficients*(AC) placed first followed by the *Detail Coefficients*(DC) (i.e. $coefficients = [CA, CD3, CD2, CD1]$, where 3, 2, 1 are the level of decomposition of the DC).

Note that due to the choice of the level (3) of the wavelet decomposition in conjunction with the length of x (52) in this paper, the mDWT will be affected by boundaries effects. This choice was made to be as close as possible to the mDWT features calculated in [63] which employed the same wavelet and level on a smaller x length (40).

3.C.3 Time-Frequency Domain Features

Short Term Fourier Transform based Spectrogram (Spectrogram)

The Fourier transform allows for a frequency-based analysis of the signal as opposed to a time-based analysis. However, by its nature, this technique cannot detect if a signal is non-stationary. As sEMG are non-stationary [45], an analysis of these signals employing the Fourier

transform is of limited use. An intuitive technique to address this problem is the STFT, which consists of separating the signal into smaller segments by applying a sliding window where the Fourier transform is computed for each segment. In this context, a window is a function utilized to reduce frequency leakage and delimits the segment's width (i.e. zero-valued outside of the specified segment). The spectrogram is calculated by computing the squared magnitude of the STFT of the signal. In other words, given a signal $s(t)$ and a window of width w , the spectrogram is then:

$$spectrogram(s(t), w) = |STFT(s(t), w)|^2 \quad (3.15)$$

Continuous Wavelet Transform (CWT)

The Gabor limit states that a high resolution both in the frequency and time-domain cannot be achieved [25]. Thus, for the STFT, choosing a wider window yields better frequency resolution to the detriment of time resolution for all frequencies and vice versa.

Depending on the frequency, the relevance of the different signal's attributes change. Low-frequency signals have to be well resolved in the frequency band, as signals a few Hz apart can have dramatically different origins (e.g. Theta brain waves (4 to 8 Hz) and Alpha brain waves (8 to 13 Hz) [71]). On the other hand, for high-frequency signals, the relative difference between a few or hundreds Hz is often irrelevant compared to its resolution in time for the characterization of a phenomenon.

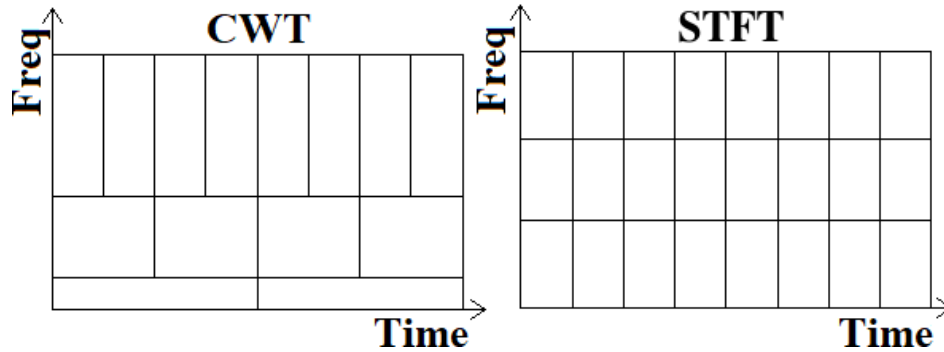


Figure 3.12 – A visual comparison between the CWT and the STFT. Note that due to its nature, the *frequency* of the CWT is, in fact, a pseudo-frequency.

As illustrated in Fig. 3.12, this behavior can be obtained by employing *wavelets*. A wavelet is a signal with a limited duration, varying frequency and a mean of zero [28]. The *mother wavelet* is an arbitrarily defined wavelet that is utilized to generate different wavelets. The idea behind the wavelets transform is to analyze a signal at different scales of the mother wavelet [29]. For this, a set of wavelet functions are generated from the mother wavelet (by applying different scaling and shifting on the time-axis). The CWT is then computed by calculating the convolution between the input signal and the generated wavelets.

3.D Hyperparameters selection for state of the art feature sets.

The hyperparameters considered for each classifiers were as follow:

- SVM: Both the RBF and Linear kernel were considered. The soft margin tolerance (C) was chosen between 10^{-3} to 10^3 on a logarithm scale with 20 values equally distributed. Similarly the γ hyperparameter for the RBF kernel was selected between 10^{-5} to 10^2 on a logarithm scale with 20 values equally distributed.
- ANN: The size of the hidden layers was selected between 20 to 1500 on a logarithm scale with 20 values equally distributed. The activation functions considered were sigmoid, tanh and relu. The learning rate was initialized between 10^{-4} to 10^0 . The L2 penalty was selected between 10^{-6} to 10^{-2} with 20 values. Finally, the solver employed is Adam and early stopping is applied using 10% of the training data as validation.
- KNN: The number of possible neighbors considered were 1, 2, 3, 4, 5, 10, 15 and 20. The metric distance considered was the Manhattan distance, the euclidean distance and the Minkowski distance of the third and fourth degree.
- RF: The range of estimators considered were between 5 to 1000 using a logarithm scale with 100 values equally distributed. The maximum number of features considered (expressed as a ratio of the total number of features fed to the RF) were: .1, .2, .3, .4, .5, .6, .7, .8, .9, 1. Additionally, both the square root and the \log_2 of the total number of features fed to the RF were also considered.

Note that the hyperparameter ranges for each classifier were chosen using 3 fold cross-validation on the *pre-training dataset*.

3.E Dimensionality Reduction on the Myo Armband Dataset for State of the Art Feature Set

Table 3.6 shows the average accuracies obtained on the *Evaluation dataset* for the state-of-the-art feature sets with and without dimensionality reduction. Note that all the results with dimensionality reduction were obtained in a week of computation. In contrast, removing the dimensionality reduction significantly augmented the required time to complete the experiments to more than two and a half months of continuous run time on an AMD-Threadripper 1900X 3.8Hz 8-core CPU.

Table 3.6 – Classification accuracy on the *Evaluation dataset* for the feature sets with and without dimensionality reduction.

	TD		Enhanced TD		Nina Pro		SampEn Pipeline	
	With Dimensionality Reduction	Without Dimensionality Reduction	With Dimensionality Reduction	Without Dimensionality Reduction	With Dimensionality Reduction	Without Dimensionality Reduction	With Dimensionality Reduction	Without Dimensionality Reduction
4 Cycles	97.76% (LDA)	96.74% (KNN)	98.14% (LDA)	96.85% (RF)	97.58% (LDA)	97.14% (RF)	97.72% (LDA)	96.72% (KNN)
3 Cycles	96.26% (KNN)	96.07% (RF)	97.33% (LDA)	95.78% (RF)	96.54% (KNN)	96.53% (RF)	96.51% (KNN)	95.90% (KNN)
2 Cycles	94.12% (KNN)	93.45% (RF)	94.79% (LDA)	93.06% (RF)	93.82% (KNN)	94.25% (SVM)	94.64% (KNN)	93.23% (KNN)
1 Cycle	90.62% (KNN)	89.28% (KNN)	91.25% (LDA)	88.63% (SVM)	90.13% (LDA)	90.32% (SVM)	91.08% (KNN)	89.27% (KNN)

3.F Reducing the number of EMG channels on the target dataset

If the new user cannot wear the same amount of electrodes as what was worn during pre-training the proposed transfer learning technique cannot be employed out of the box. A possible solution is to consider that the EMG channels are akin to the channel of an image, giving different view of the same phenomenon. In this section, the enhanced raw ConvNet is modified to accommodate this new representation. The 2D image (8 x 52) that was fed to the network is now a 1D image (of length 52) with 8 channels. The architecture now only employs 1D convolutions (with the same parameters). Furthermore, the amount of neurons in the fully connected layer was reduced from 500 to 256. The *second network* is identical to the *source network*.

Pre-training is done on the *pre-training dataset*, training on the first *round* of the *evaluation dataset* with 4 cycles of training and the test is done on the last two *rounds* of the *evaluation dataset*. The first, third, fifth and eighth channels are removed from every participant on the *evaluation dataset*. The *pre-training dataset* remains unchanged.

The non-augmented ConvNet achieves an average accuracy of 61.47% over the 17 participants. In comparison, the same network enhanced by the proposed transfer learning algorithm achieves an average accuracy of 67.65% accuracy. This difference is judged significant by the one-tail Wilcoxon Signed Rank Test (p-value=0.00494). While the performance of this modified ConvNet is noticeably lower than the other classification methods viewed so far it does show that the proposed TL algorithm can be adapted to different numbers of electrodes between the source and the target.

Chapter 4

A Low-Cost, Wireless, 3-D-Printed Custom Armband for sEMG Hand Gesture Recognition

4.1 Reference

Ulysse Côté-Allard[†], Gabriel Gagnon-Turcotte[†], François Laviolette, and Benoit Gosselin. "A Low-Cost, Wireless, 3-D-Printed Custom Armband for sEMG Hand Gesture Recognition." *Sensors* 19, no. 12 (2019): 2811.

[†]These authors contributed equally to this work.

4.2 Context

So far in this thesis, sEMG signal were obtained using the Myo Armband, the first (and only) consumer grade myoelectric armband, which was both affordable and allowed to easily access the recorded EMG signal. However, the company stopped selling it in 2018 (to instead sell "smart glasses"). Meanwhile, Gabriel Gagnon-Turcotte was finishing his PhD, during which he created a state-of-the-art biosignal acquisition system which was use on mouses' brain. The idea was then to collaborate to create a fully functional sEMG armband using the previously designed biosignal acquisition system. The two main advantage of this work as it relate to this thesis is that. 1) I was able to keep recording EMG data without relying on a piece of hardware no one could get anymore. 2) the armband provided higher sampling rate per channel and more channels, which improves potential classification performances.

4.3 Résumé

Les technologies portatives peuvent être utilisées pour permettre aux humains d'effectuer des tâches plus complexes et exigeantes de manière plus efficace. Les bracelets capables d'électromyographie de surface (sEMG) sont des dispositifs attrayants et non intrusifs, permettant de déduire l'intention humaine à l'aide d'algorithmes d'apprentissage automatique. Cependant, les systèmes d'acquisition sEMG actuellement disponibles ont tendance à coûter trop chers pour être utilisés par un individu, ne sont pas suffisamment portables ou encore sacrifient la qualité du signal enregistré pour être abordables. Ce travail présente le *3DC Armband*, un bracelet myoélectrique conçu par le Laboratoire de Microsystèmes Biomédicaux de l'Université Laval. Ce bracelet à électrode sèche sans fil de 10 canaux a une cadence d'enregistrement de 1000 points par seconde par canaux et son coût de production est estimé à 150\$USD. Il comporte également un système de mesure inertielle à 9 axes permettant d'obtenir l'orientation de l'avant-bras de l'utilisateur. Le système proposé est comparé au *Myo Armband* de Thalmic Labs, l'un des systèmes d'acquisition sEMG les plus populaire au monde. La comparaison est effectuée en utilisant un nouvel ensemble de données hors-ligne comprenant 22 participants effectuant onze gestes de la main en portant les deux bracelets simultanément. Le *3DC Armband* surpasse systématiquement et de façons significatives ($p < 0.05$) le *Myo Armband* avec trois classificateurs différents utilisant trois modalités d'entrée différentes lorsque ceux-ci sont entraînés avec dix secondes ou plus de données d'entraînement par geste. Le code source, le nouvel ensemble de données, le projet Altium et les modèles 3D pour construire le *3DC Armband* sont facilement accessibles en téléchargement grâce à un lien GitHub.

4.4 Abstract

Wearable technology can be employed to elevate the abilities of humans to perform demanding and complex tasks more efficiently. Armbands capable of surface electromyography (sEMG) are attractive and noninvasive devices from which human intent can be derived by leveraging machine learning. However, the sEMG acquisition systems currently available tend to be prohibitively costly for personal use or sacrifice wearability or signal quality to be more affordable. This work introduces the 3DC Armband designed by the *Biomedical Microsystems Laboratory* in Laval University; a wireless, 10-channel, 1000 sps, dry-electrode, low-cost (~ 150 USD) myoelectric armband that also includes a 9-axis inertial measurement unit. The proposed system is compared with the Myo Armband by Thalmic Labs, one of the most popular sEMG acquisition systems. The comparison is made by employing a new offline dataset featuring 22 able-bodied participants performing eleven hand/wrist gestures while wearing the two armbands simultaneously. The 3DC Armband systematically and significantly ($p < 0.05$) outperforms the Myo Armband, with three different classifiers employing three different input modalities when using ten seconds or more of training data per gesture. This new dataset, alongside the source code, Altium project and 3-D models are made readily available for download within a

Github repository.

4.5 Introduction

The way people interface with machines is constantly evolving with the aim of bridging the gap between human intention and machine action. Improved interfaces can profoundly alter the way entertainment is consumed or even change lives by elevating the autonomy of people living with disabilities. In certain situations, physical interfaces (e.g., touch screen and keyboard) can be replaced with the conscious modulation of biological signals by the user.

In the context of upper-limb amputees, the signals provided by muscular activity offer an attractive modality from which a user’s intention can be derived. Surface electromyography (sEMG) can also be leveraged to achieve intuitive interfaces in a vast array of domains for able-bodied participants [23, 7, 24]. sEMG signals are non-stationary and represent the sum of subcutaneous motor action potentials generated through muscular contraction [29]. In contrast with intramuscular EMG signals, which are recorded using needles that penetrate the muscle, sEMG signals are recorded directly on the participant’s skin surface [23]. While the latter has the advantage of being noninvasive, important noise is introduced when going further away from the muscle fibers, especially when nonintrusive dry electrodes are employed instead of their more intrusive and robust gel-based counterpart [38]. The quality of the sEMG acquisition system is thus crucial in obtaining as clear of a signal as possible. However, current systems tend to be expensive, often costing several thousands of dollars per channel or making noticeable compromises on the quality of the recorded signal (see Section 4.6 for details).

As such, the main contribution of this work is to present a new 3-D-printed wireless sEMG acquisition system based on an application-specific integrated circuit (ASIC). This armband aims at providing state-of-the-art recording while being low cost to produce and small enough to be easily wearable. The proposed device is referred to in this work as the 3DC Armband and was designed by the *Biomedical Microsystems Laboratory* in Laval University (BML-UL). Additionally, while comparisons of different sEMG acquisition system have been done in the past [34], these comparisons are made across different datasets and are thus generated from different conditions. Other works have explored the impact of sampling rate on gesture classification accuracy [32], but they did so by downsampling the signal of an acquisition system. While this allows for a direct comparison of the impact of the signal bandwidth on classification performance, such comparisons do not take into account the technical limitations associated with a higher bandwidth when building sEMG acquisition systems. To the best of the authors’ knowledge, this is the first time that a direct comparison between two different armbands cadenced at different sampling rates (200 vs. 1000 sps) is made on the same dataset. Finally, an additional contribution of this work is the publication of a new dataset recorded with both the Myo Armband and the proposed acquisition system on 22 able-bodied

participants for eleven hand/wrist gestures. The dataset, alongside the 3-D models, Altium project, and the source code used in this article, are made readily available to the community (<https://github.com/UlysseCoteAllard/3DC-Armband>).

This paper is organized as follows. An overview of different sEMG acquisition systems is given in Section 4.6. Section 4.7 presents the proposed 3-D-printed sEMG armband in detail. The new dataset is detailed in Section 4.8. The methods employed for the comparison between the Myo armband and the proposed system are detailed in Section 4.9, and the results of this comparison are presented in Section 6.9. Finally, this work’s outcomes and possible improvements are discussed in Section 4.11.

4.6 Overview of Surface EMG Acquisition Systems

By their nature, sEMG signals are recorded with multiple layers of material between the electrode site and the muscle fibers generating the signal. As such, particularly robust acquisition systems have been developed over the years to contend with the different types of contaminants associated with such signals (e.g., power line interference, motion artifact, and biosignal crosstalk) [17].

One such system employed for clinical research is the Ultium EMG by Noraxon systems [3] which can record up to 32 channels simultaneously at a rate of ~ 2 ksp/s and a baseline noise of less than $1\text{ }\mu\text{V}$. Each channel is fully self-contained and wireless, allowing researchers to target multiple recording sites. Each module also integrates a 9-axis Inertial Measurement Unit (IMU) sensor. Similar systems such as the Trigno Avanti by Delsys Systems [2] and DataLITE sEMG by Biometrics [1] are also available. While these systems are highly accurate, they necessitate preparation of the recording site (i.e., washing and sometimes shaving the subject’s skin) before fixing each module to the skin, often using medical tape. This, coupled with their high cost ranging between $\sim \$17,000$ – $20,000$ USD, often renders such systems impractical for consumer-grade applications.

In 2015, the Myo Armband by Thalmic Labs [5] was released as a new consumer-grade sEMG acquisition system. This wireless armband offering eight channels was retailed for several orders of magnitude less than medical-grade acquisition systems ($\$200$ USD). The armband is also nonintrusive, requiring no preparation of the recording site of any sort. However, to attain this, concessions were made both in terms of data quality and signal bandwidth. Most notably, the armband is limited to a sampling rate of 200 sps with 8-bit precision and comprised of only 8 channels. Regardless of these limitations, the Myo Armband has been widely utilized in a wide array of research topics (e.g., robotic arm control [7], video game control [39], motor imagery [30], and sign language recognition [6]).

More recently, the gForce-Pro from Oymotion [4] was released. The armband is sampled

at 1000 sps, enabling it to leverage the full spectra of the sEMG signal [32]. However, this sampling rate increase made the gForce-Pro six times more expensive than the Myo armband for the same amount of channels and recording resolution.

Several sEMG acquisition systems have been presented in the literature such as in Refs. [8, 9, 14, 42, 16]. However, these systems tend to not offer a fully developed wearable form factor [14] or are simply too bulky to be embedded within an armband [9, 16, 8, 42].

A technical comparison of the main sEMG acquisition systems previously mentioned alongside the proposed 3DC Armband is given in Table 4.1. Note that, for the rest of this work, the Myo Armband will be used for comparison with the proposed sEMG acquisition system. The system by Thalmic Labs was selected as it is “arguably the most widely known EMG armband in research” [32] (mentioned in more than 1250 articles on Google scholar). Additionally, its price range is in the same order of magnitude as the one estimated for the proposed 3DC armband.

Table 4.1 – Characterization of different surface electromyography (sEMG) acquisition systems. Values of N.A. mean that the information is not available. Note that, while the Hercules system [14] is included for completeness, it does not possess a wearable form factor.

	Delsys systems Trigno Avanti	Biometrics DataLITE sEMG	Noraxon Ultium EMG	Oymotion gForce-Pro	Thalmic Lab Myo Armband	Hercules	3DC Armband
sEMG channels	up to 16	up to 16	up to 32 (at 2000 sps) or 16 (at 4000 sps)	8	8	8	10
sEMG ADC *	16 bits	13 bits	16 bits	8 bits	8 bits	12 bits	10 bits (ENOB *) (data sent on 16 bits)
sEMG Sampling rate	1960 sps	2000 sps	4000 sps	1000 sps	200 sps	1000 sps	1000 sps
Bandwidth or Built-in Filters	20-450 Hz or 10-850 Hz	10-490 Hz	5/10/20- 500/1000/1500 Hz	20-500 Hz	~5-100 Hz	20-500 Hz	20-500 Hz
Contact Dimensions	5 mm ²	78 mm ²	N.A.	~66 mm ²	100 mm ²	78 mm ²	50 mm ²
Contact Material	Silver	Stainless Steel	N.A.	Stainless steel silver coated	Stainless Steel	Gold plated Copper	Electroless nickel immersion gold (ENIG)
Full Scale (Peak to Peak)	+/-11 sps or +/-22 sps	+/-6 sps	+/-24 sps	N.A.	~+/-1 sps (measured)	+/-6 sps	+/-3 sps
Input referred-noise (On system bandwidth)	N.A.	<5μV	<1 μV	N.A.	N.A.	N.A.	2.2 μV
IMU * sensors	9-axis Acc, Gyro, Mag	No	9-axis Acc, Gyro, Mag (if EMG set at 2000 sps or below)	9-axis Acc, Gyro, Mag	9-axis Acc, Gyro, Mag	No	9-axis Acc, Gyro, Mag
IMU Sampling rate	24-470 Hz (Acc), 24-360 Hz (Gyro), 50 Hz (Mag)	-	200 Hz	50 Hz	50 Hz	-	50 Hz
Transmitter	BLE 4.2	WiFi	2.4 GHz	BLE 4.1	BLE 4.0	Wi-Fi	Enhanced Shockburst **
Autonomy	4 to 8 h	8 h	8 h	N.A.	16 h	N.A.	6 h
Weight	14 g (per channel)	17 g (per channel)	14 g (per channel)	80 g	93 g	N.A.	62 g
Price	~\$20,000 USD (for 16 channels)	~\$17,000 USD (for 16 channels)	~\$20,000 USD (for 16 channels and free battery replacement)	\$1250 USD	\$200 USD	N.A.	~\$150 USD ***

* ADC: Analog-to-digital converter; ENOB: effective number of bits; IMU: inertial measurement unit; BLE: Bluetooth low energy. ** 2.4 GHz low-power custom protocol (similar to BLE*) from Nordic Semiconductor, Norway. *** The cost of the System-on-Chip was replaced by the cost of a comparable product: the ADS1298 from Texas Instruments, USA.

4.7 The 3DC Armband (Prototype)

The 3DC Armband, which is depicted in Figure 4.1, features ten sEMG recording channels cadenced at 1000 sps alongside a 9-axis Inertial Measurement Unit (IMU). The proposed armband weighs 63 g and is assembled with a custom System-on-Chip (SoC), featuring competitive performance for sEMG recording: input referred noise of $2.2 \mu V_{\text{rms}}$, resolution of 10 bits, dynamic range of 6 mV_{pp} , and a bandwidth of 20–500 Hz. The 3DC Armband consists of two interconnected parts. The first is the sensor printed circuit board (PCB) that includes all the electronic components for multichannel sEMG signal conditioning and multichannel sEMG data acquisition through a custom ASIC, IMU data acquisition, and wireless data transmission. The second part is the armband receptacles holding the sEMG electrodes. Both parts are interconnected using a detachable Molex connector, enabling easy electronic and software updates outside the armband. The following subsections detail each component of the proposed sEMG acquisition system.

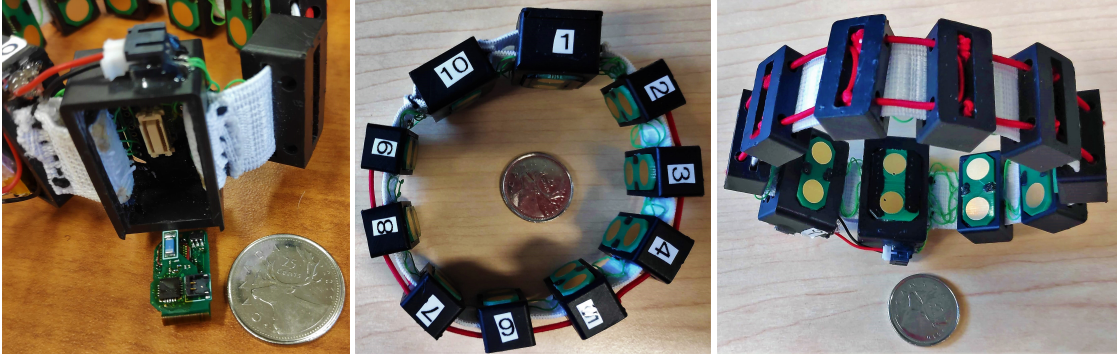


Figure 4.1 – The proposed 3DC Armband. The system and the battery are held in the receptacles identified by 1 and 10 respectively. The label on each part of the armband corresponds to the channels’ order that are recorded for the dataset described in Section 4.8.

4.7.1 System Overview

The proposed sensor, of which the system-level concept is shown in Figure 4.2, consists of six main building blocks:

1. A custom $0.13\text{-}\mu\text{m}$ complementary metal oxide semi-conductor (CMOS) mixed signal (i.e., analog and numeric circuits on the same die) SoC that can record 10 sEMG channels in parallel [19, 20].
2. An ICM-20948 low-power 9-axis IMU from InvenSense, USA. This component has a 3-axis gyroscope, a 3-axis accelerometer, and a 3-axis magnetometer.
3. An nRF24L01+ low-power wireless transceiver from Nordic Semiconductors, Norway, which sends the sEMG and IMU data to a base station at a 1 Mbps datarate.

4. An MSP430F5328 low-power microcontroller unit (MCU) from Texas Instruments, USA. This MCU is mainly used for interfacing the SoC, the IMU, and the wireless transceiver.
5. The power management unit (PMU), which includes a 1.9-V low-dropout regulator (LDO) for powering the MCU, the wireless transceiver, and the IMU. The highest voltage in the sEMG sensor is 1.9-V, which is optimized for low-power consumption since it is the smallest viable voltage for powering the MCU, the IMU, and the wireless transceiver, yielding around half the power consumption compared with a typical 3.3-V power supply. The PMU also includes a 1.2-V LDO for powering the SoC, which is the recommend supply voltage for the 0.13- μ m technology used in the SoC. The system is powered with a 100-mAh LiPo battery.
6. The Molex connector (# 0529910308) used for connecting with the Armband and for programming the MCU.

Note that the SoC was originally developed for neural electrophysiological signal acquisition and is detailed in previous publications [19, 20]. Additionally, this SoC was shown to be able to successfully acquisition sEMG data [18].

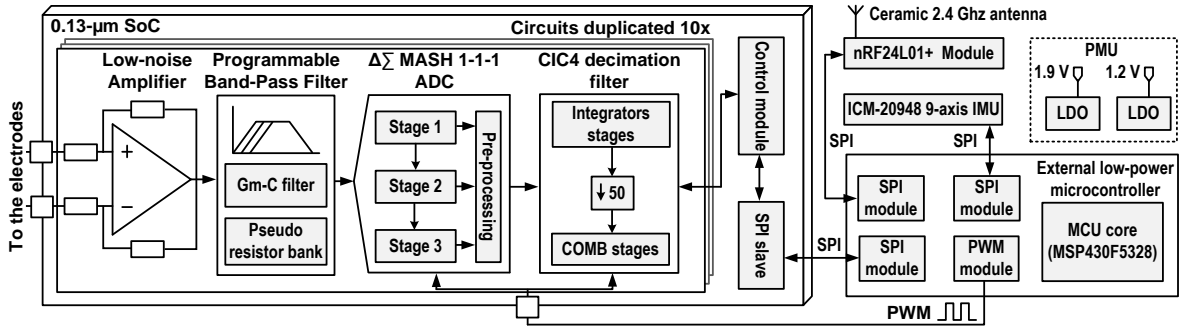


Figure 4.2 – System-level concept of the multichannel wireless sEMG sensor: The sensor is built around a custom 0.13- μ m SoC that includes 10 \times sEMG channels, each of which encompasses a bioamplifier, a $\Delta\Sigma$ analog-to-digital converter (ADC), and a 4th order decimation filter. The SoC, the nRF24L01+ low-power wireless transceiver, and the ICM-20948 9-axis IMU are interfaced with an MSP430F5328 low-power MCU.

The complete sensor is shown in Figure 4.3a with the main building blocks identified. The sensor has a flexible part, allowing it to fold the rigid parts on top of each other to save space (See Figure 4.3c). When folded, the PCB occupies 1.25 cm³. Finally, the 3DC sensor communicates with a custom-based base station consisting of (i) an nRF24L01+ low-power wireless transceiver from Nordic Semiconductors, (ii) an ARM cortex M4 MCU from Texas Instruments for managing the data, and (iii) an FT232RL UART-to-USB chip from FTDI, United Kingdom, for sending the sEMG data to the computer. The following Sections 4.7.2

and 4.7.3 give more details about the SoC, MCU firmware, and the IMU, while Section 4.7.5 presents the 3-D models that contain the armband’s electronics.

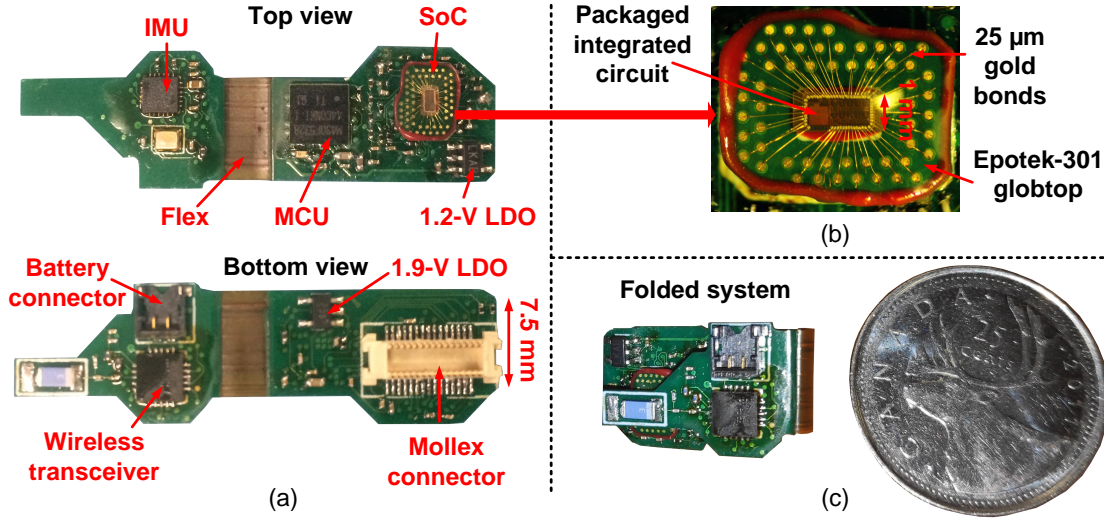


Figure 4.3 – (a) Two-sided view of the sEMG sensor with each part identified: The printed circuit board (PCB) has a flexible region to fold the two rigid parts on top of each other to save space. (b) The packaged SoC which is wirebonded directly on a PCB substrate. (c) The system folded in its final position beside a Canadian quarter coin (diameter of 23.88 mm).

4.7.2 sEMG Acquisition Interface

Each recording channel of the SoC encompasses a low-noise and low-power fully differential bioamplifier, followed by a fully differential third-order Delta-Sigma ($\Delta\Sigma$) multi-stage noise shaping (MASH) analog-to-digital converter (ADC) and an on-chip fourth-order cascaded integrator-comb (CIC) decimation filter [19, 20]. The use of fully differential topologies (amplifier and ADC) doubles the dynamic range of the SoC (6-dB increase) while being more robust to external noise sources compared to a single-ended solution [15]. In this design, the bioamplifier is a single stage AC-coupled operational transconductance amplifier (OTA) which rejects any DC offsets generated at the electrode–skin interface. This topology also features pseudo-resistors in the feedback path to produce a high-pass analog cutoff frequency of ~ 1 Hz.

Conventional ADCs require strict analog anti-aliasing filtering. This analog filtering is commonly performed with 2–3rd or higher order filtering, which increases the bioamplifier’s complexity, size, and power consumption. One advantage of using a $\Delta\Sigma$ ADC is to relax the constraints on the analog anti-aliasing filter. The proposed system performs an implicit fourth-order CIC anti-aliasing digital filter before decimation. Indeed, the $\Delta\Sigma$ pushes the *Nyquist* frequency at $F_{decim} \times OSR/2$, where F_{decim} is the sampling frequency after decimation and OSR is the oversampling ratio; thus, less restrictive filtering between $F_{decim}/2$ and $F_{decim} \times OSR/2$ is required to avoid aliasing with this type of ADC.

As it can be seen in Figure 4.4a, the oversampling of the $\Delta\Sigma$ pushes the *Nyquist* frequency far from the bandwidth after decimation. For this application, an oversampling ratio (OSR) of 50 is employed to achieve an effective number of bits (ENOB) of 10 bits, pushing the *Nyquist* frequency to 25 kHz. The analog low-pass filtering is performed implicitly by the internal analog G-mC filter of the OTA [19, 20] inside the bioamplifier, which cuts at -3 dB at ~ 7 kHz (black curve in Figure 4.4a), leading to almost no aliasing, as there is a -12 -dB attenuation at 25 kHz. The final low-pass filtering is performed by the fourth-order CIC decimation filter, which has a -3 -dB low-pass cutoff frequency of 460 Hz, which is close to the ideal cutoff frequency of 500 Hz, and with a -80 -dB attenuation per decade before the signal is downsampled to 1 kHz (blue curve in Figure 4.4a). Figure 4.4a also illustrates the Myo bandwidth for comparison. As can be seen, only a small portion (<100 Hz) of the proposed sensor bandwidth is covered by the Myo. The bioamplifier noise spectrum over a 500 Hz bandwidth is shown in Figure 4.4b. The total input referred noise is of $2.2 \mu\text{V}_{\text{rms}}$ (20–500 Hz), which is smaller than the quantifying step of the ADC (resolution of $7 \mu\text{V}$). The SoC communicates with an external MCU using a dedicated serial peripheral interface (SPI) bus and using a custom protocol to extract the data from all the channels.

The custom SoC employed for sEMG acquisition is wire-bonded onto a PCB substrate using $25 \mu\text{m}$ gold bonds and protected by an EPO-TEK 301 Glob-Top that was held in place during the curing phase using an AD1-10S dam from ChipQuick. An enlargement of the packaged SoC is depicted in Figure 4.3b.

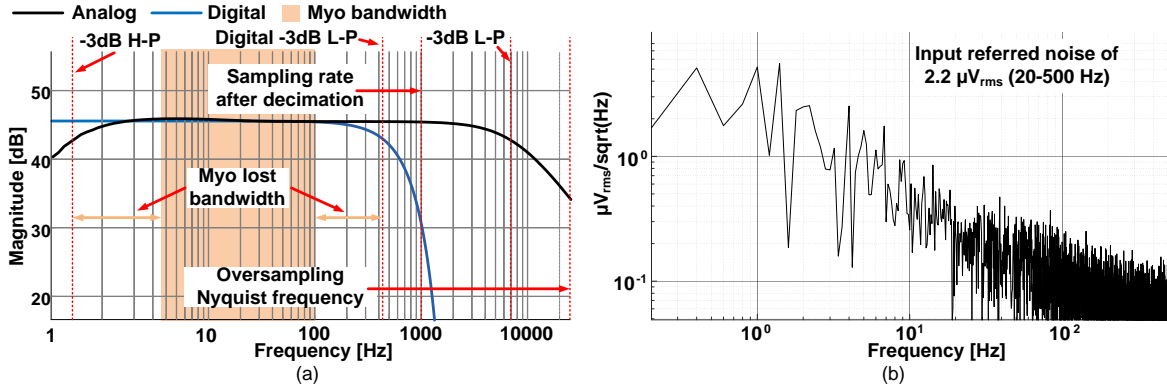


Figure 4.4 – (a) Analog bandwidth of the bioamplifier (in black), digital bandwidth of the decimation filter (in blue), Myo bandwidth comparison (in orange), and (b) noise spectrum of the bioamplifier. The input referred noise is of $2.5 \mu\text{V}_{\text{rms}}$ over a 500-Hz bandwidth.

4.7.3 MCU Firmware

The MSP430F5328 MCU controls the SoC, the IMU, and the wireless transceiver together by using three dedicated SPI busses. To use the least amount of hardware components as possible, the oversampling clock signal driving the SoC $\Delta\Sigma$ s is provided by a pulse-width modulation (PWM) module within the MCU (set at 50% duty-cycle). The sEMG sampling is triggered by

the PWM timer interruption when 50 clock cycles (OSR of 50, interruptions at 1 kHz) have been issued. Then, the MCU triggers one of the direct memory access (DMA) module channels to send commands to the SoC as to read all the 10 SoC channels one after the other. The acquired sEMG data is pushed automatically by the DMA within a First-in, First-out (FIFO) structure and sent to the wireless transceiver by another DMA channel when 20 or more bytes are available (10×2 bytes packets). Since the DMA module performs all the work, the MCU is idle most of the time. It is woken up at a 50-Hz frequency to get and forward the IMU data to the transceiver.

4.7.4 Inertial Measurement Unit

An IMU is a device consisting of accelerometers and gyroscopes from which the tracking of the device's orientation can be derived. A tri-axis magnetometer can be added to form a hybrid IMU, sometimes referred to as a *Magnetic, Angular Rate, and Gravity* (MARG) sensor [27], to reduce the orientation accumulated error. Information from an IMU system is widely employed in the domain of rehabilitation [22]. Additionally, for dynamic sEMG-based gesture recognition, orientation information from IMU devices can be leveraged to obtain higher performances than with EMG alone [21, 37, 42]. Furthermore, IMUs have been employed to increase the number of gestures that can be detected by combining the orientation of the forearm with static hand gestures [41, 44].

As the inclusion of an IMU device alongside sEMG channels allows a wider range of dynamic gestures to be detected, an armband featuring both modalities can be employed for a broader range of applications. Consequently, the ICM-20948, consisting of a low-power IMU featuring a 3-axis gyroscope, a 3-axis accelerometer, and a 3-axis magnetometer, was incorporated within the 3DC Armband. This IMU was selected for its small footprint ($3 \times 3 \text{ mm}^2$ 24-QFN package), its low power supply capability (1.9 V supported), and its high resolution of 16 bits. This chip communicates with the MCU using a 4-wire serial peripheral interface (SPI) bus cadenced at 5 MHz. The MCU extracts three $\langle x, y, z \rangle$ vectors from the IMU for the accelerometer, gyroscope, and magnetometer by reading the first user bank of the IMU. This allows the MCU to read the data all at once (18 bytes block read). The vectors are then stored in a dedicated packet and sent to the wireless transceiver for further ex situ processing. The orientation data, in the form of a quaternion, is computed from the 9-axis IMU with the Madgwick's algorithm [27] using the x-IMU implementation (<https://github.com/xioTechnologies/Open-Source-AHRS-With-x-IMU>) on the receiving computer.

4.7.5 3-D Printing Models

The armband's microelectronics are held by three different receptacles (shown in Figure 4.5) each fulfilling diverse functions. The system's holder, depicted in Figure 4.5A, shows the

system's receptacle, which also houses the main electrode. The battery and a standard size electrode are stored in the battery holder, which can be seen in Figure 4.5B.

Finally, the eight remaining standard electrodes are housed in eight small electrode receptacles. The 3-D model of these receptacles is shown in Figure 4.5C. The hole in the top is there to facilitate the assembly of the system.

The circular holes on all three receptacles serve to pass elastic cords through, that link the different modules together and ensures that the armband can be worn by a wide variety of persons. The rectangular holes serve to pass small elastic bands through, on which the different electrical cables can be attached on to link the different microelectronic components together.

The overall armband price is valued at ~\$150 USD. The price was estimated using the ADS1298 from Texas Instrument as an estimation for the custom SoC and assuming the fabrication of 20 PCBs.

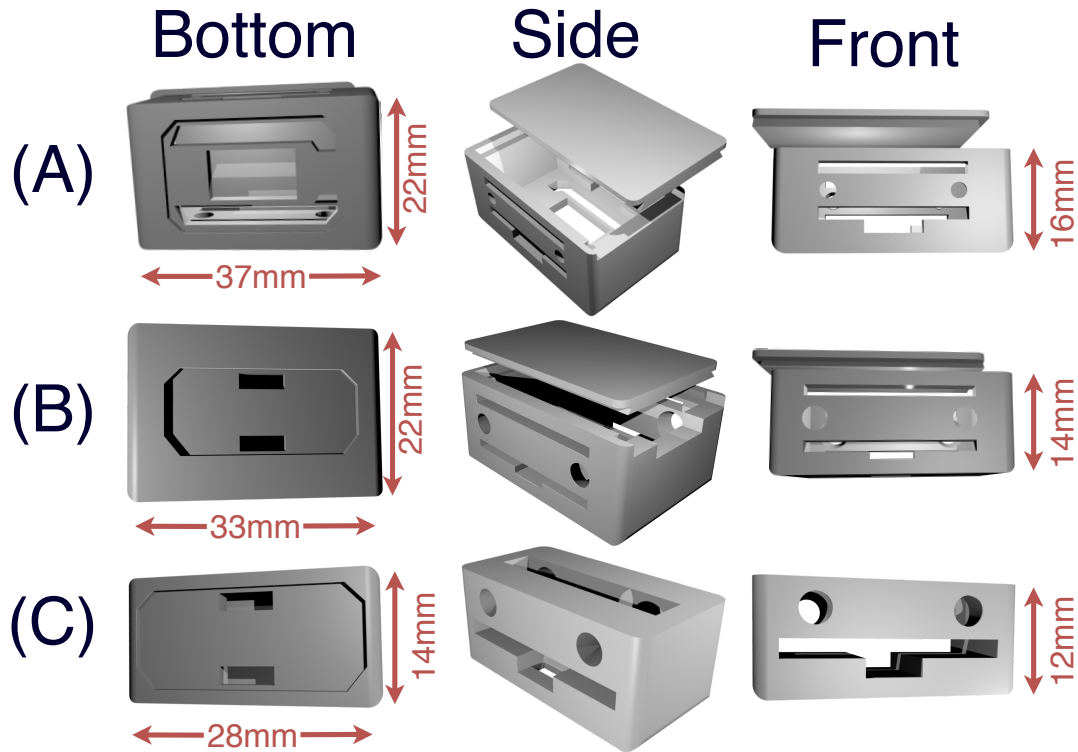


Figure 4.5 – (A) The system's receptacle: The bottom of the unit is used to receive the main electrode, while the system is stored inside. A cover slides on to enclose the system. (B) The battery holder: This receptacle is used to house the power source of the armband and, as such, should be placed next to the system's holder. Once the battery is placed, the cover can then slide on to protect the system. A standard electrode is placed on the bottom of this holder. (C) This holder houses a standard electrode. For the proposed 3DC Armband, eight such receptacles are required.

4.8 Comparison Dataset

Beyond the technical description presented in the previous section, the usefulness of the proposed armband must be assessed with real-life data. As such, a new dataset was recorded as to allow as close of a direct comparison as possible between the Myo and the proposed 3DC Armband. The dataset is comprised of 22 able-bodied participants (7F/15M, 17/5 right/left handed) aged between 23 and 69 years old (average 34 ± 14 years old).

The data acquisition protocol was approved by the Comités d'Éthique de la Recherche avec des êtres humains de l'Université Laval (approbation number: 2017-0256 A-1/10-09-2018), and informed consent was obtained from all participants.

4.8.1 Data Acquisition Protocol

Before the recording started, both the Myo and the 3DC Armband were placed simultaneously on the dominant arm of the participant. The highest armband (i.e., the one closest to the elbow) was set to its maximum diameter and slid up until the armband's circumference matched the participant's forearm circumference. For the first participant, the Myo Armband was the one placed closest to the elbow. This process was replicated for each following participant but alternating the armband closest to the elbow between each subject. The two possible armband configurations alongside examples of the range of armband placements on participants' forearm are shown in Figure 4.6. This method of positioning was adopted as to better represent the wide range of positions that nonexperts might use when wearing this type of hardware. The delay between putting the armband on the participant's forearm and the start of the experiment was approximately three minutes on average.

The proposed dataset is made of eleven hand/wrist gestures, which are presented in Figure 4.7. All gesture recordings were made with the participants standing up with their forearm parallel to the floor supported by themselves. Starting from the neutral gesture, the participants were instructed, with an auditory cue, to hold the next gesture for 5 s. The cue given to the participants were in the following form: *Gesture X, 3, 2, 1, Go*. The recording of each movement began just before the movement was started and held by the participant as to capture the ramp-up segment of the muscle activity and always started with the neutral gesture. The recording of the eleven gestures for 5 s each totaled 55 s of data and is referred to as a *cycle*. A total of four *cycles* (220 s of data) were recorded with no interruption between cycles. Then, a five min pause was observed, where the participant could relax (without removing the armbands). After the pause, another four *cycles* of data were recorded. The first four *cycles* of data are referred to and serves as the *training dataset*, while the second group of *cycles* is referred to and serves as the *test dataset*. Note that the ramp-up period is included in the labeled dataset for each gesture.



Figure 4.6 – The two different armband configurations (left/right) employed in this work with the 3DC being either above or below the Myo armband with respect to the participant’s wrist. This figure also showcases the wide variety of armband positions recorded in the proposed dataset.



Figure 4.7 – The eleven hand/wrist gestures employed in the proposed dataset.

4.8.2 Preprocessing

As the main use-case of the proposed armband is a real-time classification, a critical factor to consider is the input latency. The optimal latency (taking into account both classification performance and controller delay) was shown to be between 150–250 ms [36]. As the Myo Armband is limited to a sampling rate of 200 sps, a window size of 250 ms was selected as to not unduly give advantage to the proposed armband. Note that while the preprocessing is made so that it could be implemented for real-time classification, all results presented in this article are computed offline.

As mentioned in Section 4.7, the proposed 3DC Armband is band-pass filtered between 20–

500 Hz. However, to produce a dataset as close as possible to the raw sEMG signal, the high-pass filter was instead set at ~ 1 Hz using the SoC bioamplifier tunable pseudo-resistor bank. As such, the preprocessing of the dataset involves a fourth-order butterworth high-pass filter at 20 Hz as suggested in Ref. [32] for both armbands. An example of the signals recorded from both armbands after filtering is given in Figure 4.8.

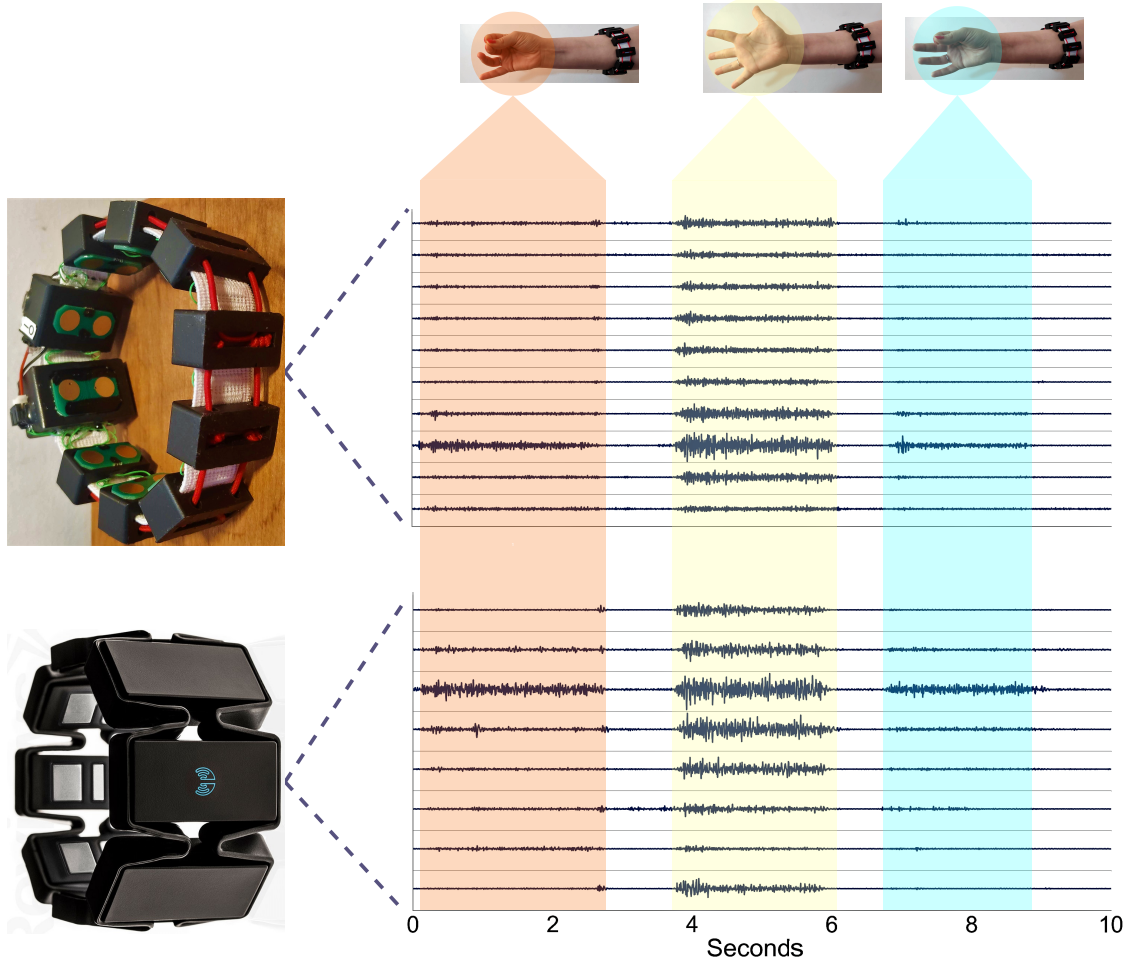


Figure 4.8 – Comparison of the signals recorded with the Myo Armband and the proposed 3DC Armband. The x-axis represents time in seconds, while the y-axis is the different channels of the armbands. The three gestures recorded in order are the *chuck grip*, *Open Hand*, and *Pinch Grip*. Note that these signals were not obtained using the *Comparison Dataset* recording protocol to show a wider array of gestures in a continuous way.

4.9 Comparison Methods

The dataset previously described (Section 4.8) is employed to qualitatively discriminate between the Myo and 3DC Armband. The comparison is rendered from three different input modalities: a baseline feature set, the raw sEMG signals, and the signals represented in the time–frequency domain. The remaining section describes the three classification methods in detail.

It should be noted that one of the goals of this work is to generate a comparison that is as fair as possible between the proposed armband and the Myo Armband. As such, several choices were made to achieve this goal, sometimes to the detriment of the classification accuracy. These choices are explicitly detailed below.

4.9.1 Baseline Method

The baseline method employs a feature set (Hudgins' Time-Domain Feature Set [13]) and a classifier (Linear Discriminant Analysis [35, 23]) widely used in the literature. Both are described in the following two subsections.

Hudgins' Time-Domain Feature Set (H-TD)

Historically, the literature on sEMG-based gesture recognition primarily centers on feature-set engineering as to characterize the sEMG signals in a discriminative way [29, 31].

Among all the feature sets proposed in the literature, the most commonly employed one is probably H-TD [32]. This set is comprised of four features from the time domain and are relatively inexpensive to compute:

- Mean Absolute Value
- Zero Crossing
- Slope Sign Changes
- Waveform Length

Detailed descriptions of these four features are given in Ref. [11]. H-TD often serves as the basis for bigger feature sets [29, 32]. As such, it is particularly well-suited as a baseline comparison between the Myo and the proposed armband.

Linear Discriminant Analysis

Several types of classifiers have been employed in the past for sEMG-based gesture recognition. Some of the most commonly employed are the Support Vector Machine (SVM) [32], Artificial Neural Networks (ANN) [29], and Linear Discriminant Analysis (LDA) [35, 23].

The latter is widely employed in the domain as it is a timely and computationally efficient classification technique both at training and prediction time while still being able to achieve high classification accuracies [25, 23].

While, SVM with has been shown in some work to be able to achieve higher classification accuracies than LDA [32], it requires hyperparameter optimization, which could bias the results

towards one specific armband. On the contrary, LDA does not require any hyperparameter optimization and can thus be employed to compare the armbands more fairly.

4.9.2 Raw sEMG Classification

With the recent advent of deep learning, the raw sEMG signal can be employed directly for gesture classification [11, 45], something which was considered “impractical” before [29].

The raw data is passed as an image of shape Channels X Samples (i.e., 8×50 for the Myo Armband and 10×250 for the 3DC Armband) to a ConvNet. Note that the raw signal is first band-pass filtered as described in Section 4.8.2. The ConvNet architecture, which can be seen in Figure 4.9, is based on the one presented in Ref. [11] as it was shown to be comparable to the current state-of-the-art. The main difference between the two is the use of Global Average Pooling in lieu of the fully connected layer to reduce the number of parameters.

The architecture used is the same for all armband configurations to not overly give advantage to one over the other. Adam [26] is employed for the ConvNet’s optimization with an initial learning rate of 0.0404709 (as used in Ref. [11]). Learning rate annealing is applied with a factor of five and a patience of five epochs. Training is done with batch size of 512, and the dropout rate is set at 0.5.

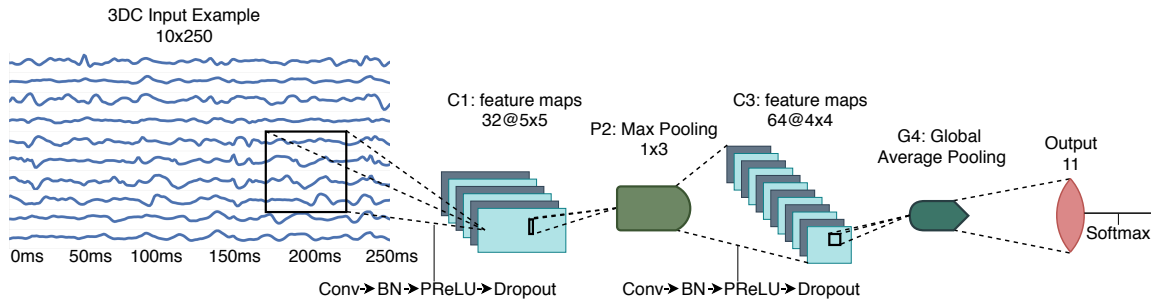


Figure 4.9 – The raw ConvNet architecture employing 34,667 parameters. In this figure, *Conv* refers to *Convolution* and *BN* refers to Batch Normalization. While the input represented in this figure is that of the 3DC, the architecture remains the same for all considered systems.

4.9.3 Time-Frequency Domain Classification

As to better investigate the potential impact of the greater sampling frequency of the 3DC Armband, the classification performance using features of the time-frequency domain is investigated.

Similar to References [10, 11], the short-time Fourier transform-based spectrograms are considered for the characterization of the sEMG signals in the time-frequency domain. For both armbands, the spectrograms are computed with Hann windows of 100 ms and an overlap of 50 ms. These hyperparameters were chosen using the *training dataset* and to use a similar ConvNet

architecture for both armbands. As suggested in References [10, 11] appropriate axis swaps are applied to yield a final image of $4 \times 8 \times 11$ and $4 \times 10 \times 51$ (i.e., time \times channel \times frequency bins) for the Myo and 3DC Armband respectively. This example's formatting allows the convolutions to be performed on spatial X frequency information, while the time is considered as different viewpoints of the same event.

The example is then fed to the ConvNet represented in Figure 7.6.3. Except for a learning rate of 0.00681292 (as used in Ref. [11]), all hyperparameters are as described in Section 4.9.2.

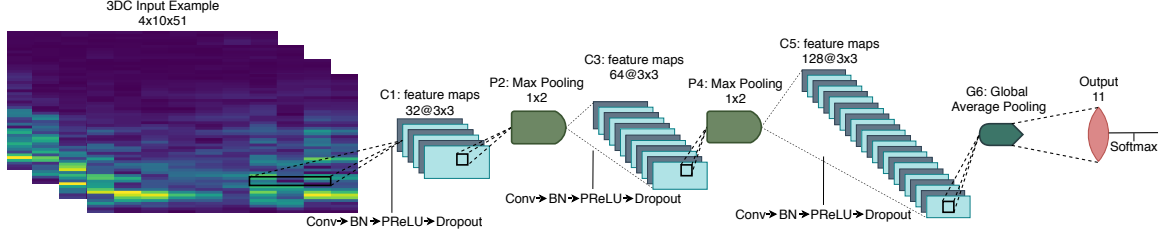


Figure 4.10 – The Spectrogram ConvNet architecture employing 95,627 parameters. In this figure, *Conv* refers to *Convolution* and *BN* refers to Batch Normalization. The input represented comes from the 3DC Armband with the channels on the x-axis and the frequency bins on the y-axis. Due to the Myo Armband associated input size, P4 and C5 were removed from the architecture when training on Myo's data.

4.10 Results

All results given in this section are computed from all four *cycles* of the *test dataset*. Training of each classifier is done with one, two, three, and four training *cycles* (i.e., 5, 10, 15, and 20 s of training data per gesture respectively). Additionally, due to the stochastic nature of the deep learning-based algorithms considered in this work, all results from each participant for each amount of training *cycles* are given as an average of 20 runs.

For statistical analysis purposes, each participant is considered as a separated dataset. As suggested in Ref. [12], the Wilcoxon Signed Rank test [40] ($n = 22$) is applied to compare between the Myo and 3DC Armband.

The comparison between the armbands with the LDA classifier is shown in Figure 4.11, while the confusion matrices for four cycles of training with the LDA classifier are given in Figure 4.12. Similarly, the comparison and associated confusion matrices for the *RAW* and *Spectrogram*-based classifiers are given in Figures 4.13, 4.14, 4.15 and 4.16 respectively. The rest of the training cycles' confusion matrices are shown in Appendix 4.A.

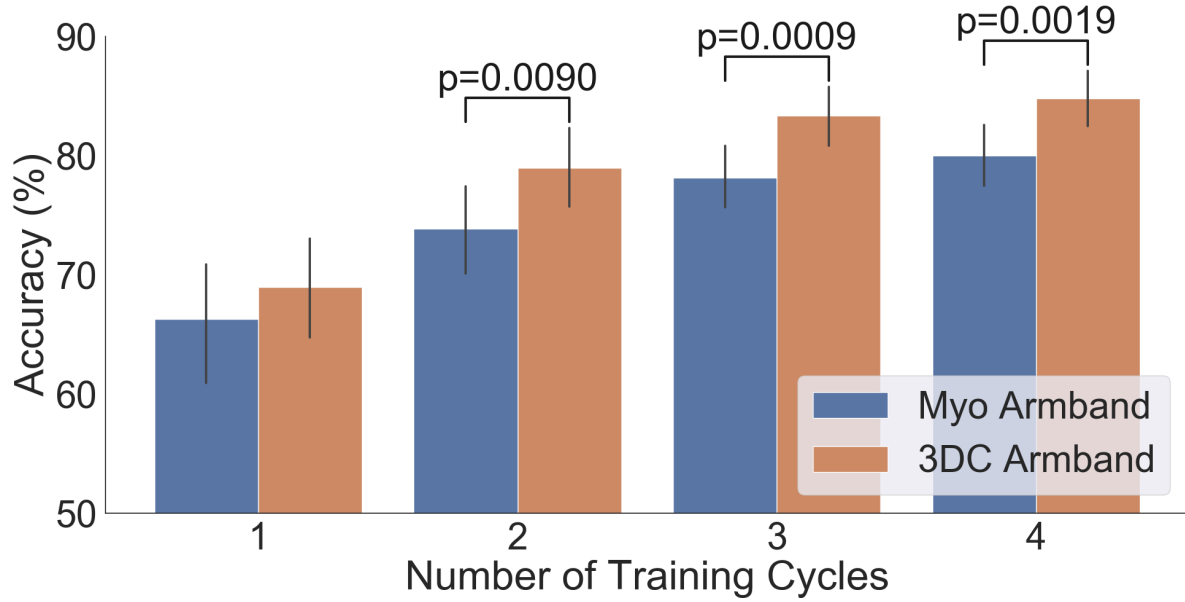


Figure 4.11 – Comparison between the Myo and the 3DC Armband employing LDA for classification: The number of cycles corresponds to the amount of data employed for training (one *cycle* equals 5 s of data per gesture). The Wilcoxon Signed Rank test is applied between the Myo and the 3DC Armband. The null hypothesis is that the median difference between pairs of observations (i.e., accuracy from the same participant with the Myo or the 3DC Armband) is zero. The p-value is shown when the null hypothesis is rejected (significant level set at $p = 0.05$). The black line represents the standard deviation calculated across all 22 participants.

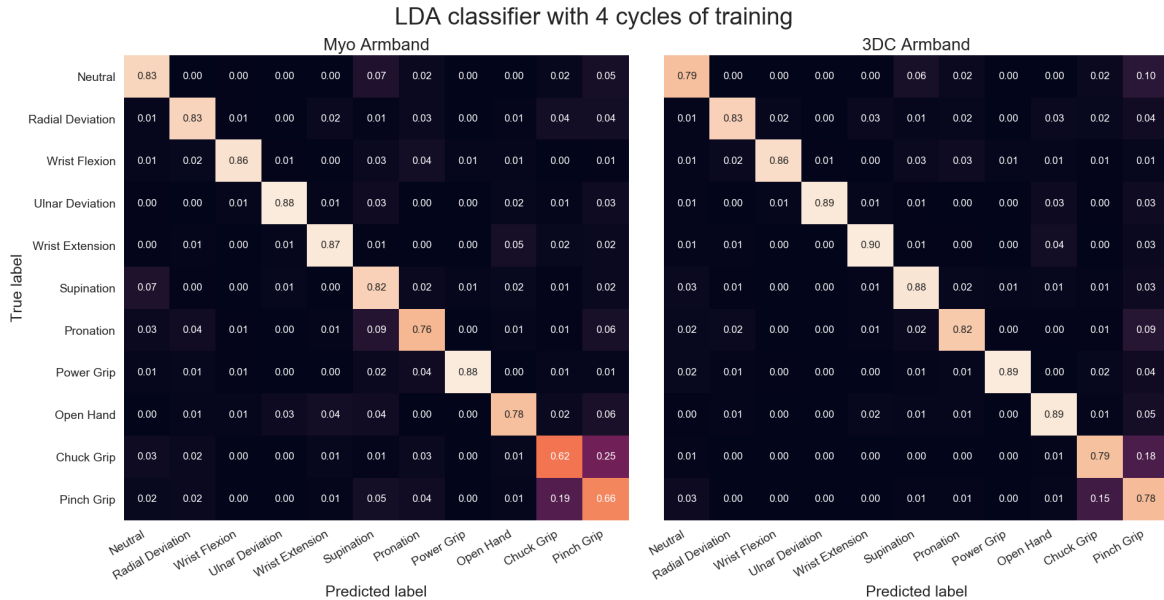


Figure 4.12 – Confusion Matrices for the Myo and the 3DC Armband employing linear discriminant analysis for classification and four cycles of training. A lighter color is better.

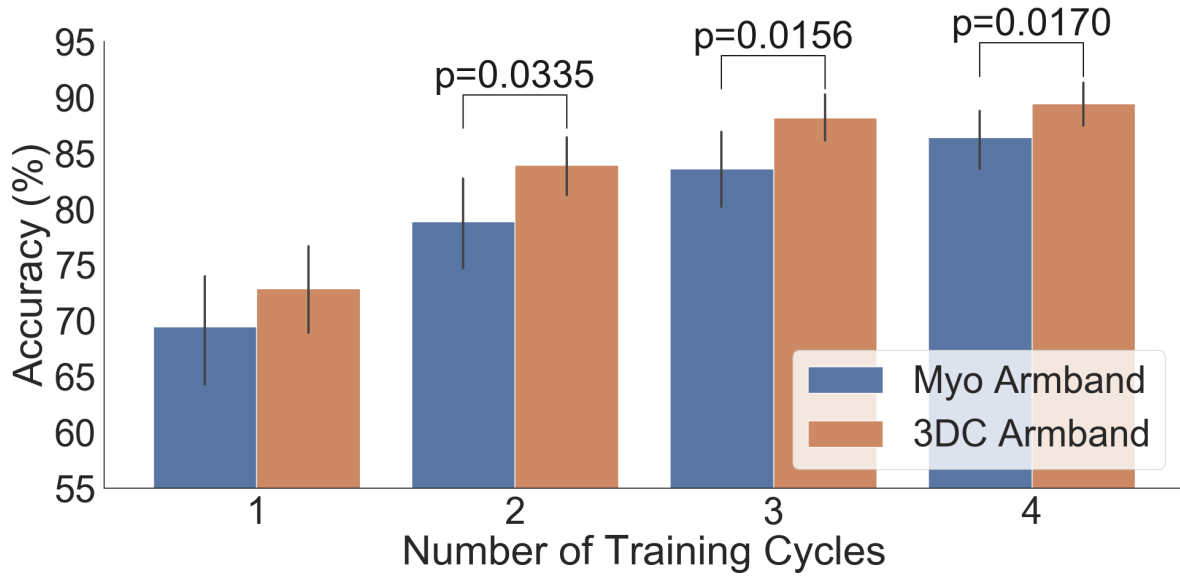


Figure 4.13 – Comparison between the Myo and the 3DC Armband employing *Raw* ConvNet for classification: The number of cycles corresponds to the amount of data employed for training (one *cycle* equals 5 s of data per gesture). The Wilcoxon Signed Rank test is applied between the Myo and the 3DC Armband. The null hypothesis is that the median difference between pairs of observations (i.e., accuracy from the same participant with the Myo or the 3DC Armband) is zero. The p-value is shown when the null hypothesis is rejected (significant level set at $p = 0.05$). The black line represents the standard deviation calculated across all 22 participants.

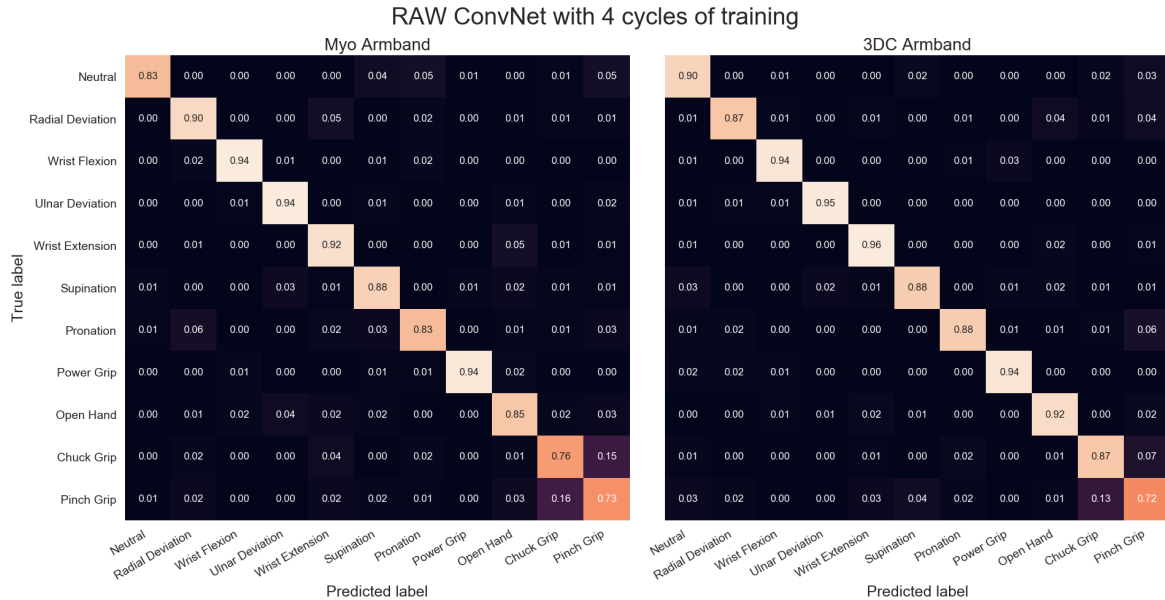


Figure 4.14 – Confusion Matrices for the Myo and the 3DC Armband employing the *Raw* ConvNet for classification and four cycles of training. A lighter color is better.

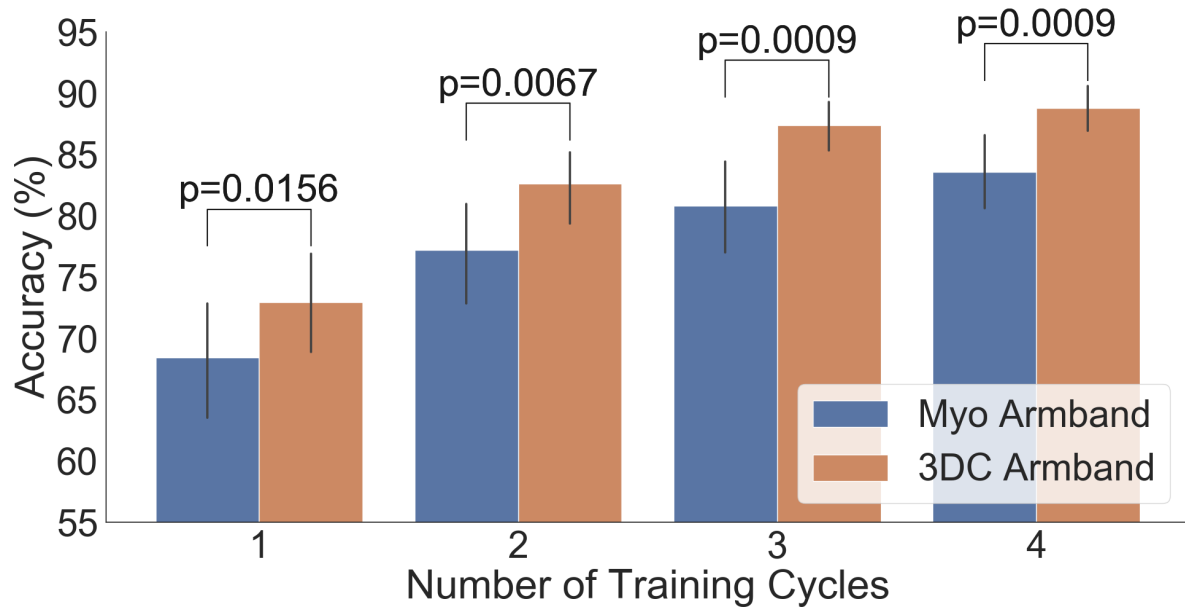


Figure 4.15 – Comparison between the Myo and the 3DC Armband employing the *Spectrogram* ConvNet for classification: The number of cycles corresponds to the amount of data employed for training (one *cycle* equals 5 s of data per gesture). The Wilcoxon Signed Rank test is applied between the Myo and the 3DC Armband. The null hypothesis is that the median difference between pairs of observations (i.e., accuracy from the same participant with the Myo or the 3DC Armband) is zero. The p-value is shown when the null hypothesis is rejected (significant level set at $p = 0.05$). The black line represents the standard deviation calculated across all 22 participants.

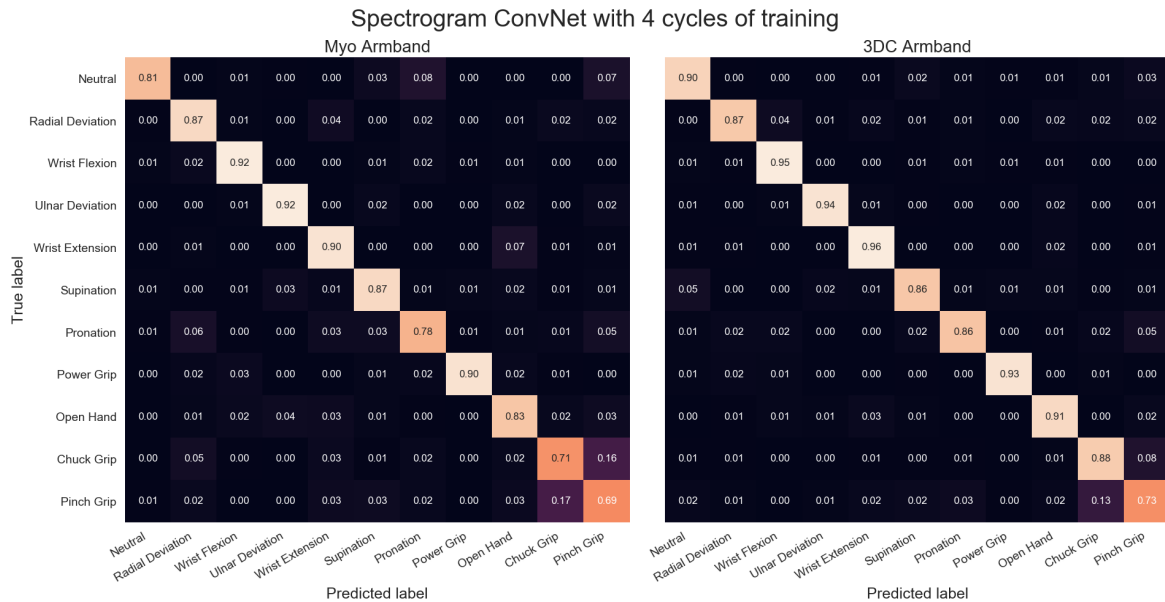


Figure 4.16 – Confusion Matrices for the Myo and the 3DC Armband employing the *Spectrogram* ConvNet for classification and four cycles of training. A lighter color is better.

4.11 Discussion

Figures 4.11, 4.13, and 4.15 show that, in all cases, the proposed armband outperforms the Myo Armband. This difference is judged significant for all instances involving two or more cycles of training by the two-tailed Wilcoxon signed-rank test. As expected, augmenting the amount of training examples systematically improves the performance of all tested classifiers for both armbands, corroborating the results in Ref. [11].

The confusion matrices show that the hardest gestures to differentiate between for all three classifiers and both armbands are the Chuck and Pinch Grip. This is expected considering they only differ by the flexion of the middle finger. Additionally, the 3DC Armband tends to outperform the Myo across all gestures. As such, one could expect that the higher spatial dimension and frequency rate yield an advantage to the proposed armband that is not gesture-specific.

Overall, the *Raw Convnet* was the best performing classifier for both armbands, achieving an average accuracy of 89.47% and 86.41% for four cycles of training with the 3DC and Myo Armband respectively. For comparison, LDA obtained 84.81% and 80.00% for four cycles of training with the 3DC and Myo Armband respectively. While more sophisticated feature engineered sets exist [29, 23], these results support previous findings showing the exciting potential of feature learning within the context of sEMG gesture recognition [33, 11].

When comparing the performance of the Myo and 3DC Armband with a single training cycle, the difference in accuracy is not judged statistically significant for both the LDA and Raw ConvNet classifier. This might be due in part to the increased spatial and frequency information provided by the 3DC Armband which naturally increases within-class variability. However, another hypothesis could be that the warm-up period (i.e., the time between putting the armband on the participant’s forearm and the start of the experiment) was not long enough for the 3DC Armband [28]. Indeed, as the 3DC electrodes are half the surface area of the Myo’s, a greater area of the skin was in contact with the Myo Armband. Thus, our hypothesis is that the Myo requires less sweat and humidity per square centimeter between the electrode and the skin to achieve a good ionic conduction. To verify this hypothesis, all three classifiers were retrained with only the last cycle of training recorded for each participant. This provides the longest warm-up period possible on the training dataset for both armbands.

For all three classifiers, the proposed sEMG acquisition system again outperforms the Myo armband. However, this difference is judged significant by the two tailed Wilcoxon signed-rank test only for the LDA classifier (p -value = 0.0309). This seems to suggest that while a longer warm up period might help the 3DC Armband, it cannot, on its own, explain why the two systems perform similarly when employing only a single cycle of training. Consequently, the proposed warm-up period hypothesis cannot be confirmed with the available results. As such, it might be that, when very few examples are available for training, the increase in computational

cost is not worth augmenting the spatial and frequency sampling rate resolution. It would be interesting to see how transfer learning algorithms developed for sEMG data affect these results [10, 11].

Future works will focus on slightly enlarging the contact area provided by 3DC Armband while making sure to not design overly large electrodes which would increase the noise of the signal from crosstalk [14]. A potential added benefit of enlarging the contact area is reducing the effect of electrode shift [43]. The relationship between warm-up time and electrode size will also be characterized. Additionally, shielding will be incorporated between the inter-connections of the electrodes.

4.12 Conclusions

This paper presents a new wearable sEMG acquisition system. The 3-D-printed armband features 10 sEMG recording channels and is cadenced at 1000 sps. The whole system is light (63 g) and incorporates a 9-axis IMU and a custom SoC. This SoC features competitive performances for this application with an input referred noise of $2.2 \mu\text{V}_{\text{rms}}$, resolution of 10 bits, dynamic range of 6 mV_{pp}, and a bandwidth of 20–500 Hz. The armband could be conceivably assembled for \sim \\$150 USD, making it more affordable and widely accessible than clinical-grade systems currently available.

The 3DC Armband was shown to significantly outperform the most widely used consumer-grade sEMG armband on a newly proposed dataset featuring 22 able-bodied participants performing 11 hand/wrist gestures.

Among the limitations of the proposed system is a possible longer warm up period than the Myo Armband. The relationship between electrode size and warm-up time will be investigated to provide a better skin-electrode interface. Shielding between the interconnections of the electrodes of the armband will also be added.

4.13 Authors Contribution

Conceptualization, U.C-A, G.G-T, F.L and B.G; methodology, U.C-A, G.G-T, F.L and B.G; software, U.C-A and G.G-T; validation, U.C-A and G.G-T; formal analysis, U.C-A and G.G-T; investigation, U.C-A and G.G-T; resources, F.L and B.G.; data curation, U.C-A and G.G-T; writing—original draft preparation, U.C-A and G.G-T; writing—review and editing, U.C-A, G.G-T, F.L and B.G; visualization, U.C-A and G.G-T; supervision, F.L and B.G; project administration, U.C-A and G.G-T; funding acquisition, U.C-A, G.G-T, F.L and B.G

4.14 Funding

This research was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) [funding reference numbers 401220434, 376091307, 114090] and the Institut de recherche Robert-Sauvé en santé et en sécurité du travail (IRSST). Cette recherche a été financée par le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG) [numéros de référence 401220434, 376091307, 114090].

4.15 Acknowledgments

The authors want to thank Alexandre Campeau-Lecours for his support without which this manuscript would not have been possible.

4.16 Conflicts of interest

The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of the data; in the writing of the manuscript; or in the decision to publish the results.

4.17 Ethical statement

All subjects gave their informed consent for inclusion before they participated in the study. The study was conducted in accordance with the Declaration of Helsinki, and the protocol was approved by the Ethics Committee of Research with human subject of Université Laval (2017-0256 A-1/10-09-2018)

4.18 Abbreviations

The following abbreviations are used in this manuscript:

ADC	Analog-to-digital converter
ANN	Artificial Neural Network
ASIC	Application-specific integrated circuit
<i>BML-UL</i>	Biomedical Microsystems Laboratory in Laval University
CIC	Cascaded integrator-comb
CMOS	Complementary metal-oxide-semiconductor
ConvNet	Convolutional Network
DMA	Direct memory access
ENOB	Effective number of bits
H-TD	Hudgins' Time-Domain Feature Set
IMU	sample per second
LDA	Linear Discriminant Analysis
LDO	Low-dropout regulator
MARG	Magnetic, Angular Rate, and Gravity
MCU	Microcontroller unit
<i>ms</i>	milliseconds
OSR	Oversampling ratio
OTA	Operational transconductance amplifier
sEMG	Surface Electromyography
SNR	Signal to Noise Ratio
SoC	System-on-chip
SPI	serial peripheral interface
<i>sps</i>	sample per second
SVM	Support Vector Machine
PCB	Printed circuit board
PMU	Power management unit
PWM	Pulse-width modulation
<i>USD</i>	United States dollar

4.19 Bibliography

- [1] Biometrics datalite wireless surface emg sensor. <http://www.biometricsltd.com/wireless-sensors.htm#emg>. Accessed: 2019-04-23.
- [2] Delsys trigno avanti sensor. <https://www.delsys.com/trigno/research/#trigno-avanti-sensor>. Accessed: 2019-04-23.
- [3] Noraxon ultium emg. <https://www.noraxon.com/our-products/ultium-emg/#1541097779421-89a192e6-7d8d>. Accessed: 2019-04-23.
- [4] Oymotion g-force pro. <http://www.oymotion.com/site/>. Accessed: 2019-04-23.

- [5] Thalmics labs myo armband. <https://support.getmyo.com/hc/en-us>. Accessed: 2019-04-23.
- [6] João Gabriel Abreu, João Marcelo Teixeira, Lucas Silva Figueiredo, and Veronica Teichrieb. Evaluating sign language recognition using the myo armband. In *2016 XVIII Symposium on Virtual and Augmented Reality (SVR)*, pages 64–70. IEEE, 2016.
- [7] Ulysse Côté Allard, François Nougrou, Cheikh Latyr Fall, Philippe Giguère, Clément Gosselin, François Laviolette, and Benoit Gosselin. A convolutional neural network for robotic arm guidance using semg based frequency-features. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2464–2470. IEEE, 2016.
- [8] Simone Benatti, Filippo Casamassima, Bojan Milosevic, Elisabetta Farella, Philipp Schönle, Schekeb Fateh, Thomas Burger, Qiuting Huang, and Luca Benini. A versatile embedded platform for emg acquisition and gesture recognition. *IEEE transactions on biomedical circuits and systems*, 9(5):620–630, 2015.
- [9] Kang-Ming Chang, Shin-Hong Liu, and Xuan-Han Wu. A wireless semg recording system and its application to muscle fatigue detection. *Sensors*, 12(1):489–499, 2012.
- [10] Ulysse Cote-Allard, Cheikh Latyr Fall, Alexandre Campeau-Lecours, Clément Gosselin, François Laviolette, and Benoit Gosselin. Transfer learning for semg hand gestures recognition using convolutional neural networks. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1663–1668. IEEE, 2017.
- [11] Ulysse Côté-Allard, Cheikh Latyr Fall, Alexandre Drouin, Alexandre Campeau-Lecours, Clément Gosselin, Kyrre Glette, François Laviolette, and Benoit Gosselin. Deep learning for electromyographic hand gesture signal classification using transfer learning. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2019.
- [12] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006.
- [13] Kevin Englehart and Bernard Hudgins. A robust, real-time control scheme for multifunction myoelectric control. *IEEE transactions on biomedical engineering*, 50(7):848–854, 2003.
- [14] Mert Ergeneci, Kaan Gokcesu, Erhan Ertan, and Panagiotis Kosmas. An embedded, eight channel, noise canceling, wireless, wearable semg data acquisition system with adaptive muscle contraction detection. *IEEE transactions on biomedical circuits and systems*, 12(1):68–79, 2018.
- [15] C. Latyr Fall et al. Wireless semg-based body-machine interface for assistive technology devices. *IEEE Journal of Biomedical and Health Informatics*, 21(4):967–977, 2017.

- [16] Yinfeng Fang, Xiangyang Zhu, and Honghai Liu. Development of a surface emg acquisition system with novel electrodes configuration and signal representation. In *International Conference on Intelligent Robotics and Applications*, pages 405–414. Springer, 2013.
- [17] Graham D Fraser, Adrian DC Chan, James R Green, and Dawn T MacIsaac. Automated biosignal quality analysis for electromyography using a one-class support vector machine. *IEEE Transactions on Instrumentation and Measurement*, 63(12):2919–2930, 2014.
- [18] Q. Mascaret M. Biemann L. Bouyer B. Gosselin G. Gagnon-Turcotte, C. L. Fall. A multichannel wireless semg sensor endowing a 0.13 μm cmos mixed-signal soc. *IEEE Life Sciences Conference (LSC)*, pages 1–4, 2018.
- [19] Gabriel Gagnon-Turcotte, Christian Ethier, Yves De Köninck, and Benoit Gosselin. A 0.13- μm cmos soc for simultaneous multichannel optogenetics and electrophysiological brain recording. *IEEE International Solid-State Circuits Conference (ISSCC)*, 32(3):466–468, 2018.
- [20] Gabriel Gagnon-Turcotte, Mehdi Noormohammadi Noormohammadi Khiarak, Christian Ethier, Yves De Koninck, and Benoit Gosselin. A 0.13- μm cmos soc for simultaneous multichannel optogenetics and neural recording. *IEEE J. Solid-State Circuits*, 53(11):3087–3100, 2018.
- [21] Marcus Georgi, Christoph Amma, and Tanja Schultz. Recognizing hand and finger gestures with imu based motion and emg based muscle activity sensing. In *Biosignals*, pages 99–108, 2015.
- [22] Alfonso Gómez-Espinosa, Nancy Espinosa-Castillo, and Benjamín Valdés-Aguirre. Foot-mounted inertial measurement units-based device for ankle rehabilitation. *Applied Sciences*, 8(11):2032, 2018.
- [23] Maria Hakonen, Harri Piitulainen, and Arto Visala. Current state of digital signal processing in myoelectric interfaces and related applications. *Biomedical Signal Processing and Control*, 18:334–359, 2015.
- [24] Matthias Janke and Lorenz Diener. Emg-to-speech: Direct generation of speech from facial electromyographic signals. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 25(12):2375–2385, 2017.
- [25] Shuo Jiang, Bo Lv, Weichao Guo, Chao Zhang, Haitao Wang, Xinjun Sheng, and Peter B Shull. Feasibility of wrist-worn, real-time hand, and surface gesture recognition via semg and imu sensing. *IEEE Transactions on Industrial Informatics*, 14(8):3376–3385, 2018.
- [26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

- [27] Sebastian OH Madgwick, Andrew JL Harrison, and Ravi Vaidyanathan. Estimation of imu and marg orientation using a gradient descent algorithm. In *2011 IEEE international conference on rehabilitation robotics*, pages 1–7. IEEE, 2011.
- [28] Jens Muhlstell and Olaf Such. Dry electrodes for monitoring of vital signs in functional textiles. *IEEE Engineering in Medicine and Biology Society (EMBS)*, 1:2212–2215, 2004.
- [29] Mohammadreza Asghari Oskoei and Huosheng Hu. Myoelectric control systems—a survey. *Biomedical signal processing and control*, 2(4):275–294, 2007.
- [30] Helen O’Shea and Aidan Moran. To go or not to go? pupillometry elucidates inhibitory mechanisms in motor imagery. *Journal of Cognitive Psychology*, 30(4):466–483, 2018.
- [31] Angkoon Phinyomark, S Hirunviriya, C Limsakul, and P Phukpattaranont. Evaluation of emg feature extraction for hand movement recognition based on euclidean distance and standard deviation. In *ECTI-CON2010: The 2010 ECTI International Confernce on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, pages 856–860. IEEE, 2010.
- [32] Angkoon Phinyomark, Rami N Khushaba, and Erik Scheme. Feature extraction and selection for myoelectric control based on wearable emg sensors. *Sensors*, 18(5):1615, 2018.
- [33] Angkoon Phinyomark and Erik Scheme. Emg pattern recognition in the era of big data and deep learning. *Big Data and Cognitive Computing*, 2(3):21, 2018.
- [34] Stefano Pizzolato, Luca Tagliapietra, Matteo Cognolato, Monica Reggiani, Henning Müller, and Manfredo Atzori. Comparison of six electromyography acquisition setups on hand movement classification tasks. *PloS one*, 12(10):e0186132, 2017.
- [35] Erik Scheme and Kevin Englehart. Electromyogram pattern recognition for control of powered upper-limb prostheses: state of the art and challenges for clinical use. *Journal of Rehabilitation Research & Development*, 48(6), 2011.
- [36] Lauren H Smith, Levi J Hargrove, Blair A Lock, and Todd A Kuiken. Determining the optimal window length for pattern recognition-based myoelectric control: balancing the competing effects of classification error and controller delay. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 19(2):186–192, 2011.
- [37] David St-Onge, Ulysse Côté-Allard, Kyrre Glette, Benoit Gosselin, and Giovanni Beltrame. Engaging with robotic swarms: Commands from expressive motion. *ACM Transactions on Human-Robot Interaction*, 8(2), 2019.
- [38] Dick F Stegeman, Bert U Kleine, Bernd G Lapatki, and Johannes P Van Dijk. High-density surface emg: techniques and applications at a motor unit level. *Biocybernetics and biomedical engineering*, 32(3):3–27, 2012.

- [39] Aaron Tabor, Scott Bateman, and Erik Scheme. Evaluation of myoelectric control learning using multi-session game-based training. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(9):1680–1689, 2018.
- [40] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80–83, 1945.
- [41] Michael T Wolf, Christopher Assad, Adrian Stoica, Kisung You, Henna Jethani, Matthew T Vernacchia, Joshua Fromm, and Yumi Iwashita. Decoding static and dynamic arm and hand gestures from the jpl biosleeve. In *2013 IEEE Aerospace Conference*, pages 1–9. IEEE, 2013.
- [42] Jian Wu, Lu Sun, and Roozbeh Jafari. A wearable system for recognizing american sign language in real-time using imu and surface emg sensors. *IEEE journal of biomedical and health informatics*, 20(5):1281–1290, 2016.
- [43] Aaron J Young, Levi J Hargrove, and Todd A Kuiken. The effects of electrode size and orientation on the sensitivity of myoelectric pattern recognition systems to electrode shift. *IEEE Transactions on Biomedical Engineering*, 58(9):2537–2544, 2011.
- [44] Xu Zhang, Xiang Chen, Yun Li, Vuokko Lantz, Kongqiao Wang, and Jihai Yang. A framework for hand gesture recognition based on accelerometer and emg sensors. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 41(6):1064–1076, 2011.
- [45] Muhammad Zia ur Rehman, Asim Waris, Syed Gilani, Mads Jochumsen, Imran Niazi, Mohsin Jamil, Dario Farina, and Ernest Kamavuako. Multiday emg-based classification of hand motions with deep learning techniques. *Sensors*, 18(8):2497, 2018.

4.A Confusion Matrices

4.A.1 LDA Classifier Confusion Matrices

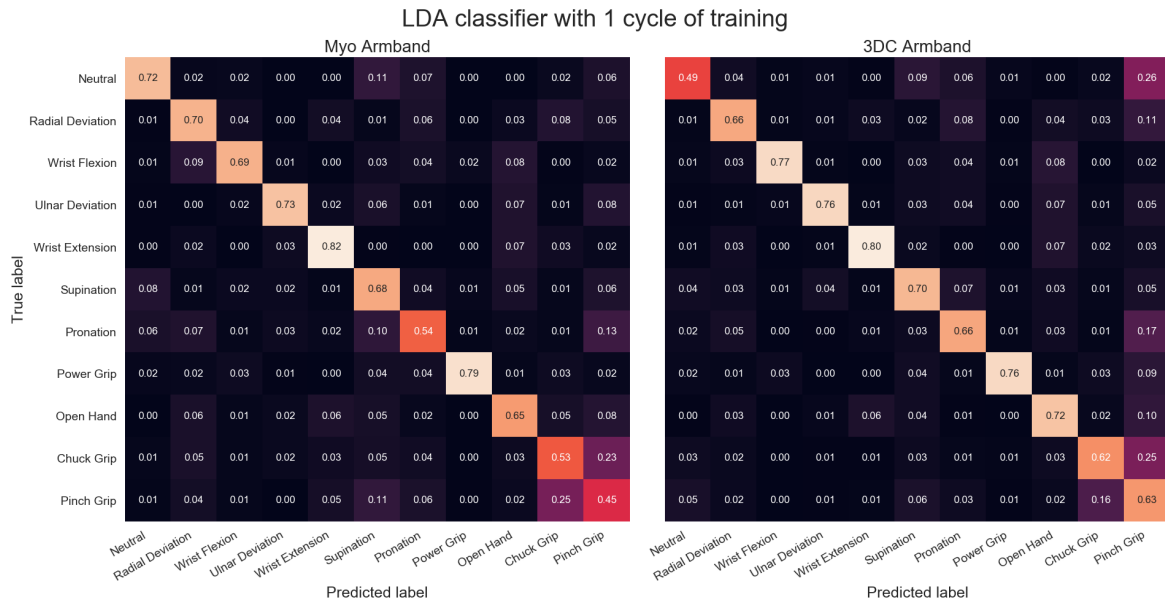


Figure 4.17 – Confusion Matrices for the Myo and the 3DC Armband employing LDA for classification and one cycle of training. Lighter is better.

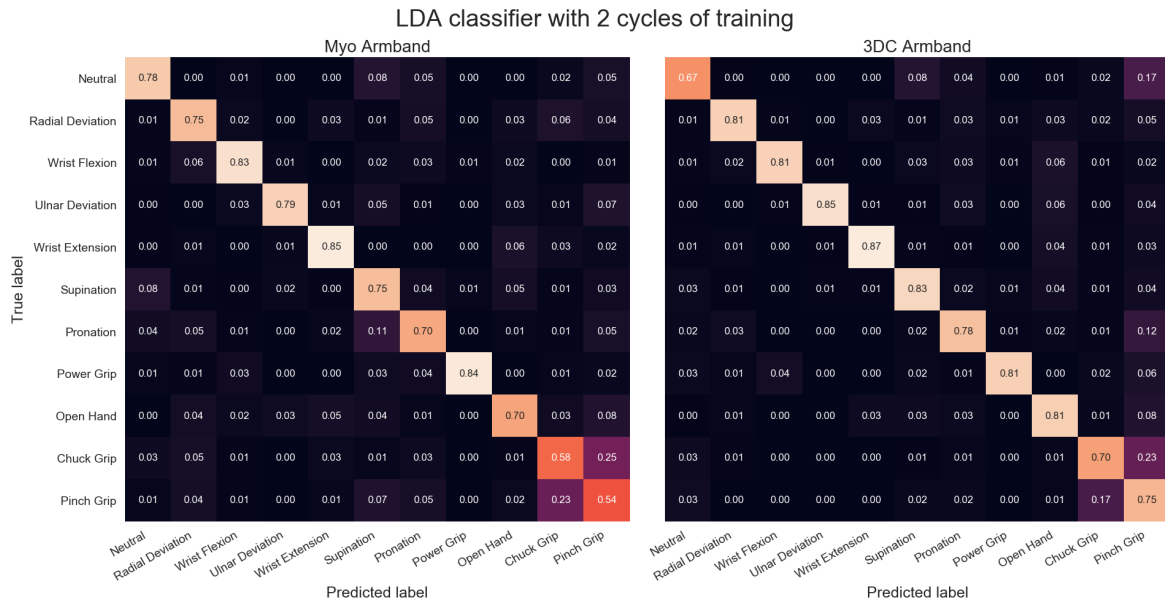


Figure 4.18 – Confusion Matrices for the Myo and the 3DC Armband employing LDA for classification and two cycles of training. Lighter is better.

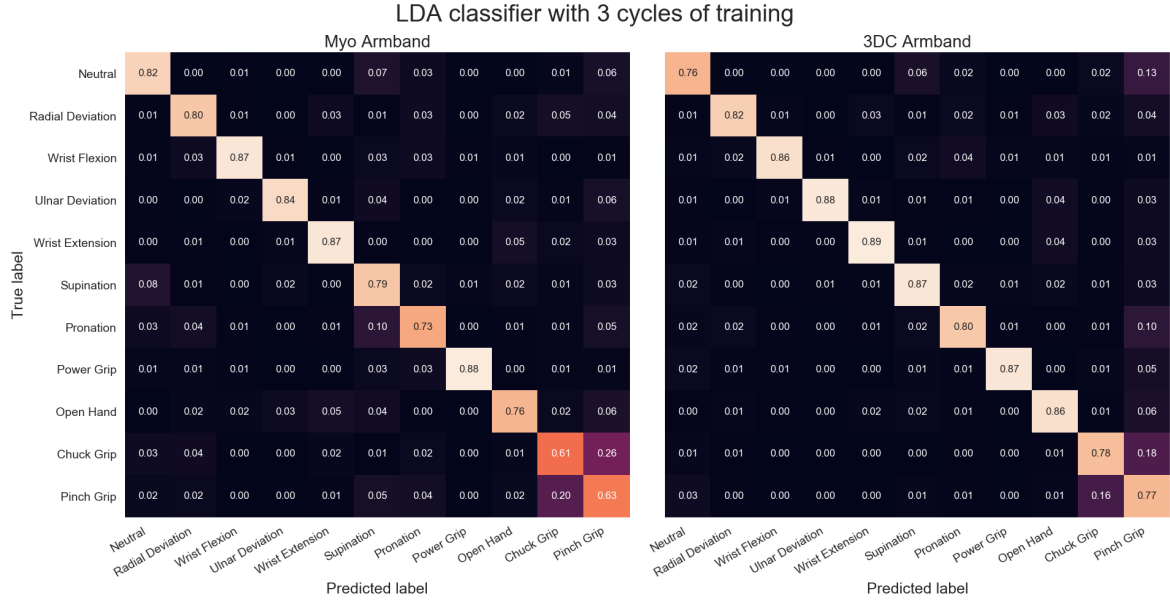


Figure 4.19 – Confusion Matrices for the Myo and the 3DC Armband employing LDA for classification and three cycles of training. Lighter is better.

4.A.2 Raw ConvNet Classifier Confusion Matrices

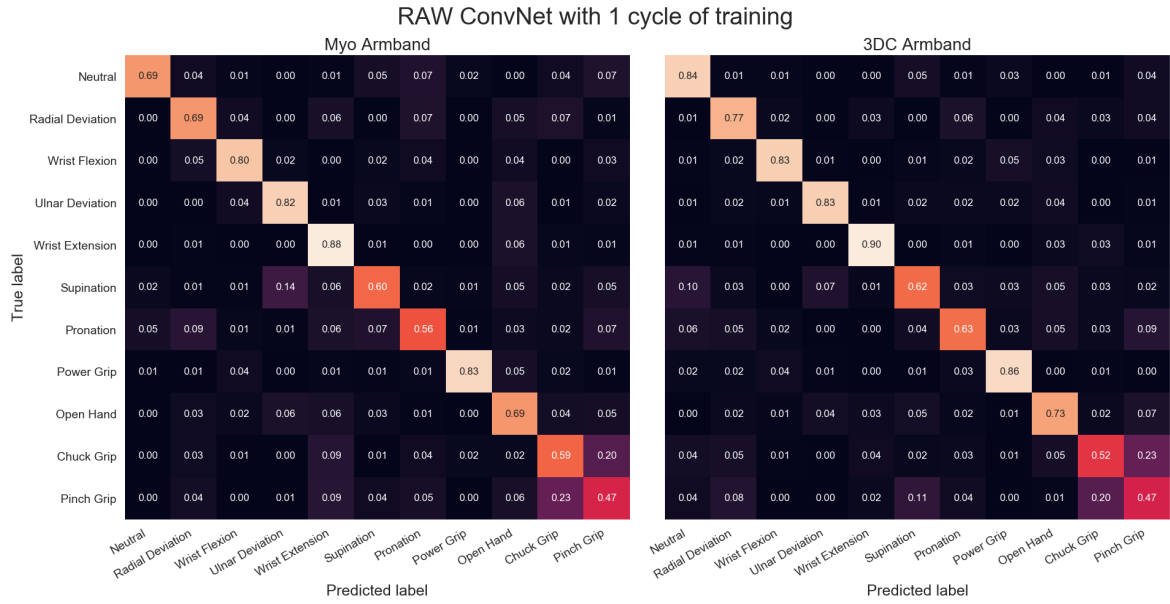


Figure 4.20 – Confusion Matrices for the Myo and the 3DC Armband employing the Raw ConvNet for classification and one cycle of training. Lighter is better.

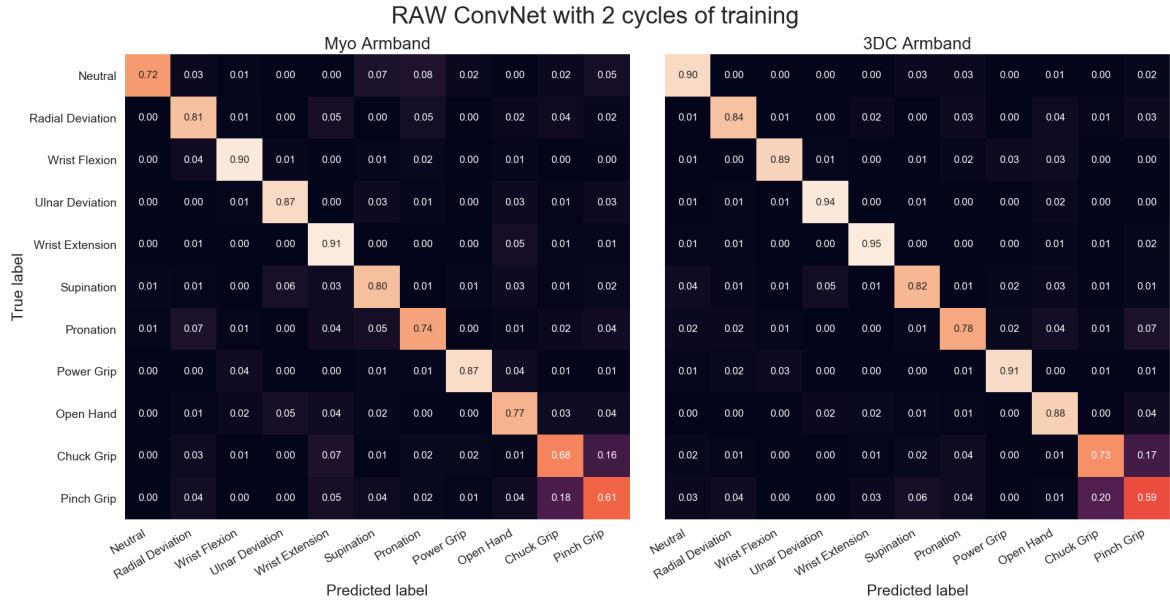


Figure 4.21 – Confusion Matrices for the Myo and the 3DC Armband employing the *Raw* ConvNet for classification and two cycles of training. Lighter is better.

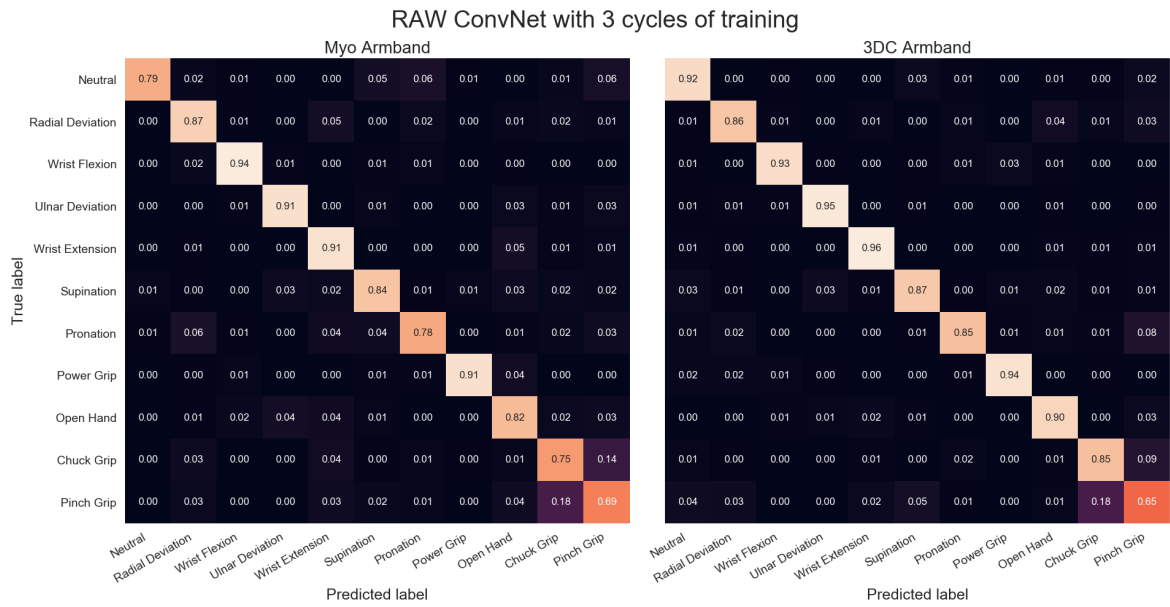


Figure 4.22 – Confusion Matrices for the Myo and the 3DC Armband employing the *Raw* ConvNet for classification and three cycles of training. Lighter is better.

4.A.3 Spectrogram ConvNet Classifier Confusion Matrices

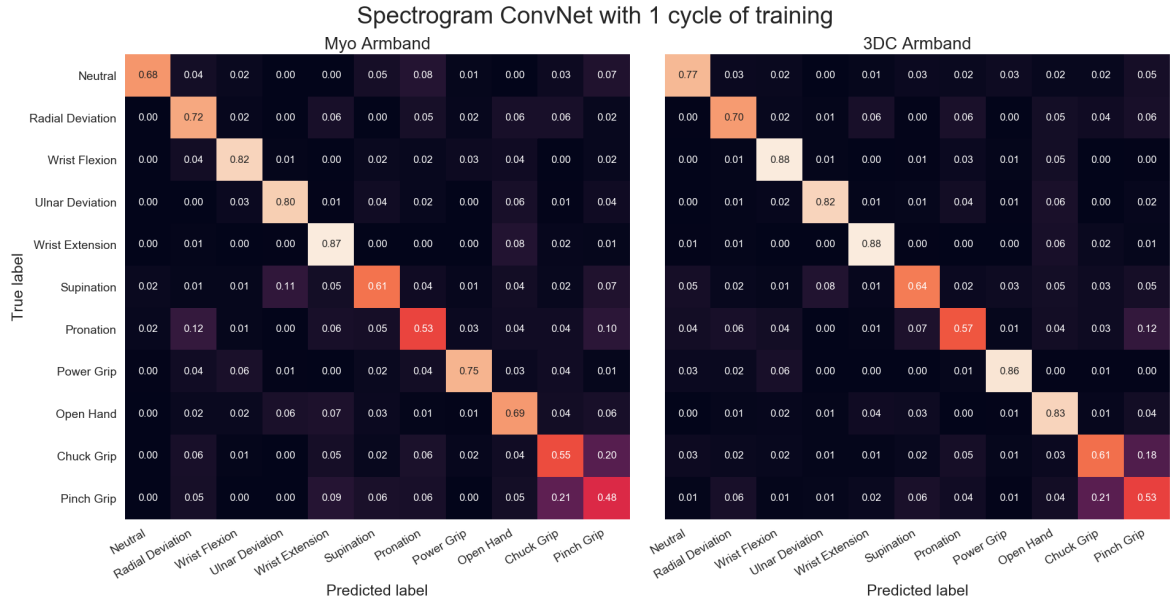


Figure 4.23 – Confusion Matrices for the Myo and the 3DC Armband employing the *Spectrogram* ConvNet for classification and one cycle of training. Lighter is better.

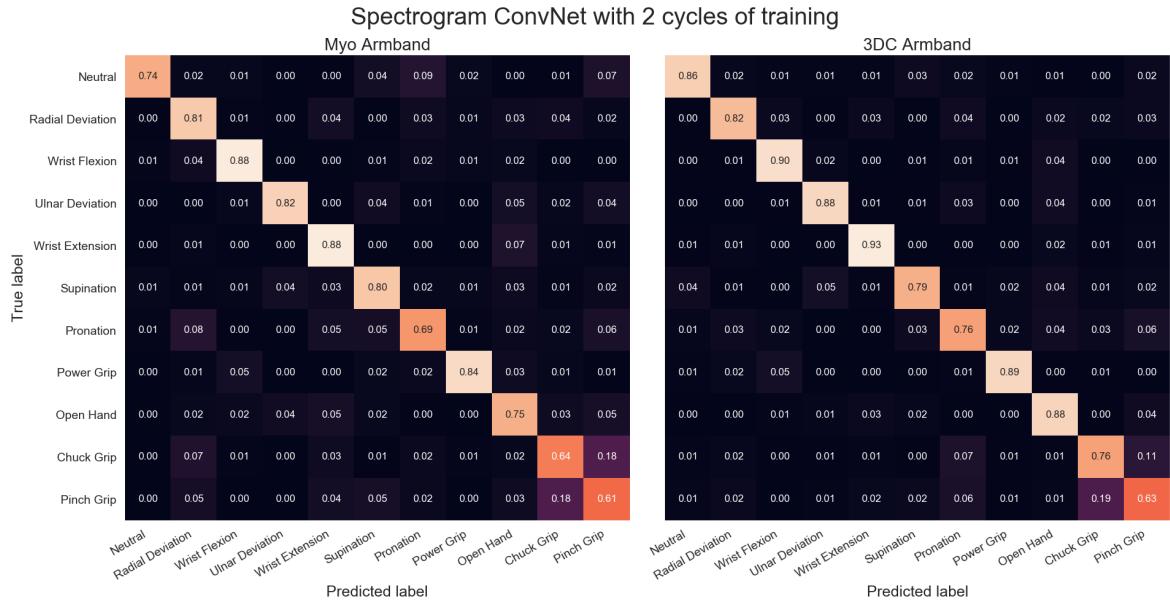


Figure 4.24 – Confusion Matrices for the Myo and the 3DC Armband employing the *Spectrogram* ConvNet for classification and two cycles of training. Lighter is better.

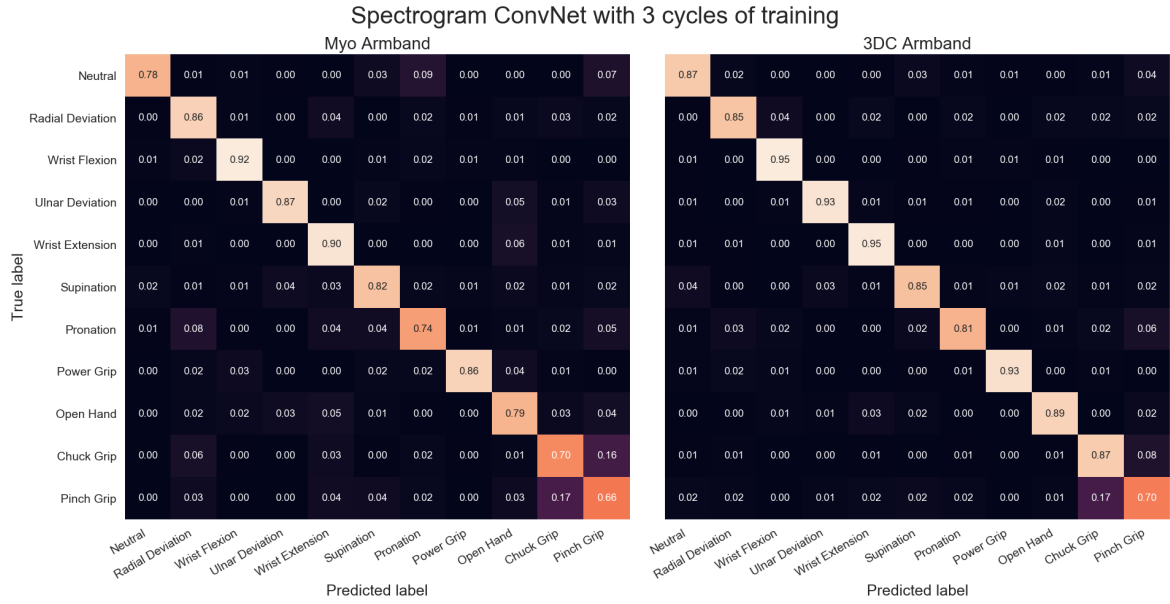


Figure 4.25 – Confusion Matrices for the Myo and the 3DC Armband employing the *Spectrogram* ConvNet for classification and three cycles of training. Lighter is better.

Chapter 5

Interpreting Deep Learning Features for Myoelectric Control: A Comparison with Handcrafted Features

5.1 Reference

Ulysse Côté-Allard[†], Evan Campbell[†], Angkoon Phinyomark, François Laviolette[‡], Benoit Gosselin[‡] and Erik Scheme[‡]. "Interpreting Deep Learning Features for Myoelectric Control: A Comparison with Handcrafted Features." *Frontiers in Bioengineering and Biotechnology*, no. 8 (2020), p.158.

[†]These authors contributed equally to this work.

[‡]These authors share senior authorship

5.2 Context

Deep learning became increasingly popular within the field of sEMG-based gesture recognition. However, contrary to natural language processing and image recognition, no work tried to understand what the networks are learning when trained on sEMG raw data. More importantly, given the historic importance of handcrafted features within the field of sEMG-based gesture recognition, it would be valuable to understand in what ways does the deep learning features relate to the features engineered over the decades. This paper is a first attempt in understanding furthering our understanding of deep learning within that field.

5.3 Résumé

La recherche sur les systèmes de contrôle myoélectrique se concentre principalement sur l'extraction de représentations discriminantes du signal électromyographique (EMG) en construisant des caractéristiques à la main. Récemment, des techniques d'apprentissage profond ont été appliquées à la tâche difficile de la reconnaissance des gestes basée sur l'EMG. L'adoption de ces techniques son lentement en train de changer les efforts de recherche de la création par des humains de caractéristiques intéressantes vers l'apprentissage automatique de nouvelles caractéristiques. Cependant, de par la nature des algorithmes d'apprentissage profond, il est difficile de comprendre quel type d'information est réellement apprise par les réseaux et de savoir comment cette information est liée aux caractéristiques construites par l'humain. De plus, en raison de la grande variabilité des enregistrements EMG entre les participants, les caractéristiques apprises automatiquement ont tendance à mal se généraliser entre différents sujets lorsque les réseaux sont entraînés avec des méthodes classiques. Par conséquent, ce travail introduit un nouvel algorithme d'apprentissage multidomaines, nommé ADANN, qui améliore de façons significatives ($p=0.00004$) la précision de la classification inter-sujets de 19.40% en moyenne par rapport à l'apprentissage standard.

À l'aide des caractéristiques apprises par ADANN, la principale contribution de ce travail est de fournir la première analyse topologique de données de reconnaissance de gestes basée sur l'EMG pour la caractérisation des informations encodées dans un réseau profond, en utilisant les caractéristiques conçues par l'humain comme points de repère. Cette analyse révèle que les caractéristiques fabriquées à la main et les caractéristiques apprises (dans les premières couches du réseau) tentent toutes les deux de faire la distinction entre tous les gestes, mais n'encodent pas les mêmes informations pour ce faire. En outre, l'utilisation de techniques de visualisation de réseaux convolutionnels révèle que les caractéristiques apprises ont tendance à ignorer le canal le plus activé durant une contraction musculaire pour produire un geste. Cela contraste fortement avec la prévalente philosophie derrière une grande partie des caractéristiques fabriquées à la main, puisque ces dernières sont conçues pour capturer des informations d'amplitude du signal. Pour finir, cet article ouvre la voie à des ensembles de fonctionnalités hybrides en montrant clairement le type d'information complémentaire encodé dans les caractéristiques apprises et les caractéristiques fabriquées à la main.

5.4 Abstract

Existing research on myoelectric control systems primarily focuses on extracting discriminative characteristics of the electromyographic (EMG) signal by designing handcrafted features. Recently, however, deep learning techniques have been applied to the challenging task of EMG-based gesture recognition. The adoption of these techniques slowly shifts the focus from feature engineering to feature learning. Nevertheless, the black-box nature of deep

learning makes it hard to understand the type of information learned by the network and how it relates to handcrafted features. Additionally, due to the high variability in EMG recordings between participants, deep features tend to generalize poorly across subjects using standard training methods. Consequently, this work introduces a new multi-domain learning algorithm, named ADANN (Adaptive Domain Adversarial Neural Network), which significantly enhances ($p = 0.00004$) inter-subject classification accuracy by an average of 19.40% compared to standard training.

Using ADANN-generated features, this work provides the first topological data analysis of EMG-based gesture recognition for the characterisation of the information encoded within a deep network, using handcrafted features as landmarks. This analysis reveals that handcrafted features and the learned features (in the earlier layers) both try to discriminate between all gestures, but do not encode the same information to do so. In the later layers, the learned features are inclined to instead adopt a one-versus-all strategy for a given class. Furthermore, by using convolutional network visualization techniques, it is revealed that learned features actually tend to ignore the most activated channel during contraction, which is in stark contrast with the prevalence of handcrafted features designed to capture amplitude information. Overall, this work paves the way for hybrid feature sets by providing a clear guideline of complementary information encoded within learned and handcrafted features.

5.5 Introduction

Surface Electromyography (sEMG) is a technique employed in a vast array of applications from assistive technologies [71, 57] to bio-mechanical analysis [5], and more generally as a way to interface with computers and robots [99, 83]. Traditionally, the sEMG-based gesture recognition literature primarily focuses on feature engineering as a way to increase the information density of the signal to improve gesture discrimination [48, 72, 58]. In the last few years, however, researchers have started to leverage deep learning [7, 3, 63], shifting the paradigm from feature engineering to feature learning.

Deep learning is a multi-level representation learning method (i.e. methods that learn an embedding from an input to facilitate detection or classification), where each level generates a higher, more abstract representation of the input [35]. Conventionally, the output layer (i.e., classifier or regressor) only has direct access to the output of the highest representation level [35, 4]. In contrast, several works have also fed the intermediary layers' output directly to the network's head [75, 95, 38]. Arguably, the most successful approach using this design philosophy is DenseNet [30], a type of convolutional network (ConvNet) where each layer receives the feature maps of all preceding layers as input. Features learned by ConvNets were also extracted to be employed in conjunction with (or replace) handcrafted features when training conventional machine learning algorithms (e.g., support vector machine, linear

discriminant analysis, decision tree) [65, 45, 37, 14]. Within the context of sEMG-based gesture recognition, deep learning was shown to be competitive with the current state of the art [18] and when combined with handcrafted features, to outperform it [14]. This last result seems to indicate that, for sEMG signals, deep-learned features provide useful information that may be complementary to those that have been engineered throughout the years. However, the black box nature of these deep networks means that understanding what type of information is encapsulated throughout the network, and how to leverage this information, is challenging.

The main contribution of this work is, therefore, to provide the first extensive analysis of the relationship between handcrafted and learned features within the context of sEMG-based gesture recognition. Understanding the feature space learned by the network could shed new insights on the type of information contained in sEMG signals. In turn, this improved understanding will allow the creation of better handcrafted features and facilitate the creation of new hybrid feature sets using this feature learning paradigm.

An important challenge arises when working with biosignals, as extensive variability exists between subjects [26, 9, 42, 13, 28]. Especially within the context of sEMG-based gesture recognition [13, 28]. Consequently, features learned using traditional deep learning training methods can be highly participant-specific, which would hinder the goal of this work of learning a general feature representation of sEMG signals. By defining each participant as a different *domain*, however, this issue can be framed as a Multi-Domain Learning problem (MDL) [96], with the added restriction that the network’s weights should be participant-agnostic. Multiple popular and effective MDL algorithms have been proposed over the years [43, 44, 67]. For example, [44] proposed to use a shared network across multiples domains with one predictive head per domain. In [96], a single head was shared across two parallel networks with one of them receiving the example’s representation as input, while the other receives a vector representation of the associated domain of the example. These algorithms however are ill-suited for this work’s context as they: do not explicitly impose domain-agnostic weight learning [96], can scale poorly with the number of domains (i.e. participants) [44], or are restricted to encode a single domain within their learned features (and use adaptor blocks to bridge the gap between domains) [67]. Unsupervised domain-adversarial training algorithms [1, 23, 88, 76] predict an unlabeled dataset by learning a representation on a labeled dataset that makes it hard to distinguish between examples from either distribution. However, these algorithms are often not designed to learn a unique representation across more than two domains simultaneously [1, 23, 88, 76], can be destructive to the source domain representation (through iterative process) [76], and by nature of the problem they are trying to solve, do not leverage the labels of the target domains. As such, this work presents a new multi-domain adversarial training algorithm, named ADANN (Adaptive Domain Adversarial Neural Network). ADANN trains a network across multiple domains simultaneously while explicitly penalizing any domain-variant representations to study learned features that generalize well across participants.

In this work, the sEMG information encapsulated within the general deep learning features learned by ADANN, is characterized using handcrafted features as landmarks in a topological network. This network is generated via the Mapper algorithm [79], with t -Stochastic Neighbor Embedding (t-SNE) [39], a non-linear dimensionality reduction visualization method, as the filter function. Mapper is a Topological Data Analysis (TDA) tool that excels at determining the shape of high dimensional data, by providing a faithful representation of it through a topological network. This TDA tool has been applied as a solution to numerous challenging applications across a wide array of domains; for example, uncovering the dynamic organization of brain activity during various tasks [68] or identifying a subgroup of breast cancer with 100% survival rate and no metastasis [46]. Mapper has also been applied to determine relationships between feature space for physiological signal pain recognition [11], and EMG-based gesture recognition [54]. However, to the best of the authors’ knowledge, the use of TDA to interpret information harnessed within deep-learned features using handcrafted features as landmarks has yet to be explored.

In this paper, convNet visualization techniques are also leveraged as a way to highlight how the network makes class-discriminant decisions. Several works [77, 22, 98, 82] have proposed to visualize network’s predictions by emphasizing which input-pixels have the most impact on the network’s output, consequently, fostering a better understanding of what the network has learned. For example, [77] used partial derivatives to compute pixel-relevance for the network output. Another example is Guided Backpropagation [82], which modifies the computation of the gradient to only include paths within the network that positively contribute to the prediction of a given class. When compared with saliency maps [77], Guided Backpropagation results in qualitative visualization improvements [73]. While these methods produce resolutions at a pixel level, the images produced with respect to different classes are nearly identical [73]. Other types of algorithms provide highly class-discriminative visualizations, but at a lower resolution [100, 74] and sometimes require a specific ConvNet architecture [100] to use. Within this work, Guided Gradient-weighted Class Activation Mapping (Guided Grad-CAM) [73] is employed as it provides pixel-wise input resolution while being class-discriminative. Another advantage of this technique is that it can be implemented on any ConvNet-based architecture without requiring re-training. To the best of the authors’ knowledge, this is the first time that deep learning visualization techniques are applied to EMG signals.

5.6 Material and Methods

A flowchart of the material, methods and experiment is shown in Figure 5.1. This section is divided as follows: first, a description of the dataset and preprocessing used in this work is given in Section 5.6.1. Then, the handcrafted features are presented in Section 5.6.2. The ConvNet architecture and the new multi-domain adversarial training algorithm (ADANN) are presented in Section 5.6.3 and 5.6.3, respectively. A brief overview of Guided Grad-CAM

is given in Section 5.6.3, while Section 5.6.3 and 5.6.3 present single feature classification and handcrafted feature regression, respectively. Finally, the Mapper algorithm is detailed in Section 5.6.4.

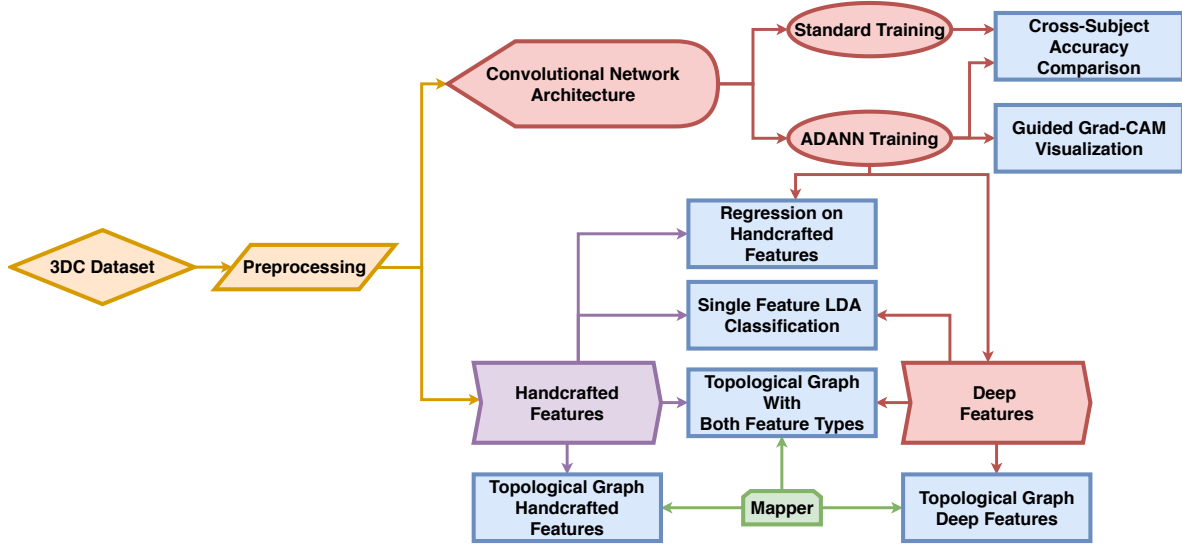


Figure 5.1 – Diagram of the workflow of this work. The *3DC Dataset* is first preprocessed before being used to train the network using standard training and the proposed ADANN training procedure. The handcrafted features are directly calculated from the preprocessed dataset, while the deep features are extracted from the ConvNet trained with ADANN. In the diagram, the blue rectangles represent experiments and the arrows show which methods/algorithms are required to perform them.

5.6.1 EMG Data

The dataset employed in this work is the *3DC Dataset* [19], featuring 22 able-bodied participants performing ten hand/wrist gestures + neutral (see Figure 5.2 for the list of gestures). This dataset was recorded with the 3DC Armband; a wireless, 10-channel, dry-electrode, 3D printed sEMG armband. The device samples data at 1000 Hz per channel, allowing the feature extraction to take advantage of the full spectra of sEMG signals [64]. Informed consent was obtained from all participants, as approved by Laval University’s Research Ethics Committee [19].

The dataset was built as follows: Each participant was asked to perform and hold each gesture for a period of five seconds starting from the neutral position to produce a *cycle*. Three more cycles were recorded to serve as the *training dataset*. After a five minute break, four new cycles were recorded to serve as the *test dataset*. Note that the validation set and hyperparameter selection are made from the training dataset.

As this work aims to understand the type of features learned by deep network in the context of myoelectric control systems, a critical factor to consider is the input latency. [81] showed



Figure 5.2 – The eleven hand/wrist gestures recorded in the *3DC Dataset* (image re-used from [19])

that the optimal guidance latency was between 150 and 250 ms. As such, the data from each participant was segmented into 151 ms frames with an overlap of 100 ms. The raw data was then band-pass filtered between 20-495 Hz using a 4th-order Butterworth filter.

5.6.2 Handcrafted Features

Handcrafted features are characteristics extracted from windows of the EMG signal using established mathematical equations. The purpose of these feature extraction methods is to enhance the information density of the signal so as to improve discrimination between motion classes [48, 58]. Across the myoelectric control literature, hundreds of handcrafted feature extraction methods have been presented [48, 58, 61]. As such, implementing the exhaustive set of features that has been proposed is impractical. Instead, within this study a comprehensive subset of 79 of the most commonly used features is employed. With a comprehensive set of features, past literature has identified five functional groups that summarize all sources of information current handcrafted feature extraction techniques describe: signal amplitude and power (SAP), nonlinear complexity (NLC), frequency information (FI), time-series modeling (TSM), and unique (UNI) [54, 10]. The SAP functional group includes time-domain energy or power features (e.g. Root Mean Squared, Mean Absolute Value). The FI functional group generally refers to features extracted from the frequency domain, or features that describe spectral properties (e.g. Mean Frequency, Zero Crossings). The NLC functional group corresponds to features that describe entropy or similarity based information (e.g. Sample Entropy, Maximum Fractal Length). The TSM functional group represents features that attempt to reconstruct the data provided through stochastic or other algorithmic models (e.g. Autoregressive Coefficients, Cepstral Coefficients). Finally, the UNI functional group represents features that capture various other modalities of information, such as measures of signal quality

or a combination of other functional groups (e.g. Signal to Motion Artefact Ratio, Time Domain Power Spectral Descriptors).

Ref	Feature Extraction Method	Name	Group
[58]	Amplitude of the First Burst	AFB	SAP
[33]	Difference Absolute Mean Value	DAMV	SAP
[33]	Difference Absolute Standard Deviation Value	DASDV	SAP
[97]	Difference Log Detector	DLD	SAP
[58]	Difference Temporal Moment	DTM	SAP
[97]	Difference Variance Value	DVARV	SAP
[97]	Difference v-Order	DV	SAP
[50]	Integral of Electromyogram	IEMG	SAP
[97]	Log Detector	LD	SAP
[2]	Second-Order Moment	M2	SAP
[49]	Modified Mean Absolute Value 1	MMAV1	SAP
[49]	Modified Mean Absolute Value 2	MMAV2	SAP
[69]	Mean Absolute Value	MAV	SAP
[58]	Maximum	MAX	SAP
[20]	Multiple Hamming Windows	MHW	SAP
[20]	Mean Power	MNP	SAP
[20]	Multiple Trapezoidal Windows	MTW	SAP
[69]	Root Mean Squared	RMS	SAP
[20]	Spectral Moment	SM	SAP
[20]	Sum of Squared Integral	SSI	SAP
[58]	Temporal Moment	TM	SAP
[20]	Total Power	TTP	SAP
[97]	Variance	VAR	SAP
[97]	v-Order	V	SAP
[58]	Waveform Length	WL	SAP
[47, 49]	Frequency Ratio	FR	FI
[86, 85]	Median Frequency	MDF	FI
[86, 85]	Mean Frequency	MNF	FI
[58]	Slope Sign Change	SSC	FI
[97]	Zero Crossings	ZC	FI
[62]	Sample Entropy	SAMPEN	NLC
[62]	Approximate Entropy	APEN	NLC
[97]	Willison's Amplitude	WAMP	NLC
[25]	Box-Counting Fractal Dimension	BC	NLC
[27]	Katz Fractal Dimension	KATZ	NLC
[6]	Maximum Fractal Length	MFL	NLC
[50]	Autoregressive Coefficients	AR	TSM
[50]	Cepstral Coefficients	CC	TSM
[50]	Difference Autoregressive Coefficient	DAR	TSM
[50]	Difference Cepstral Coefficients	DCC	TSM
[60, 59]	Detrend Fluctuation Analysis	DFA	TSM
[66]	Power Spectrum Ratio	PSR	TSM
[78, 41]	Signal to Noise Ratio	SNR	TSM
[55, 56]	Critical Exponent	CE	UNI
[78, 41]	Maximum to Minimum Drop in Power Density Ratio	DPR	UNI
[58]	Histogram	HIST	UNI
[87, 89]	Kurtosis	KURT	UNI
[58]	Mean Absolute Value Slope	MAVS	UNI
[78, 41]	Power Spectrum Deformation	OHM	UNI
[62]	Peak Frequency	PKF	UNI
[84]	Power Spectrum Density Fractal Dimension	PSDFD	UNI
[87, 89]	Skewness	SKEW	UNI
[78, 41]	Signal to Motion Artefact Ratio	SMR	UNI
[2]	Time Domain Power Spectral Descriptors	TSPSD	UNI
[58]	Variance of Central Frequency	VCF	UNI
[62]	Variance Fractal Dimension	VFD	UNI

Table 5.1 – Handcrafted features extracted for topological landmarks sorted by functional group.

Table 5.1 presents the 56 handcrafted feature methods considered in this work. Note that some methods produce multiple features (e.g. Cepstral Coefficients, Histogram), resulting in a total of 79 features. The SAP, FI, NLC, TSM, and UNI feature groups are represented here by 25, 5, 6, 7, and 13 feature extraction methods respectively. In the TDA of the deep learned features (see Section 5.6.4), these handcrafted features serve as landmarks for well-understood properties of the EMG signal. In the regression model analysis (see Section 5.6.3), the flow of information through the ConvNet is visualized by employing the handcrafted features methods as the target of the network.

5.6.3 Convolutional Network

The following subsections present the deep learning architecture, training methods and visualization techniques employed in this paper. The PyTorch [51] implementation employed in this work is [available here](#).

Architecture

Recent works on sEMG-based gesture recognition using deep learning have shown that ConvNets trained with the raw sEMG signal as input were able to achieve similar classification accuracy to the current state of the art [101, 18]. Consequently, and to reduce bias, the preprocessed raw data (see Section 5.6.1) is passed directly as an image of shape 10×151 (Channel \times Sample) to the ConvNet.

The ConvNet’s architecture, which is depicted in Figure 5.3, contains six *blocks* followed by a fully connected layer for gesture-classification. The network’s topology was selected to obtain a deep network with a limited number of learnable parameters (to avoid overfitting) with simple layer connections to enable an easier, and thus more thorough analysis. All architecture choices and hyperparameter selection were performed using the training set of the *3DC Dataset* or inspired by previous works ([18] and [19]). Each block encapsulates a convolutional layer [35], followed by batch normalization (BN) [31], leaky ReLU (slope=0.1) [94] and dropout [21] (with a drop rate set at 0.35 following [18]). The number of blocks within the network was selected to obtain a sufficiently deep network to study how the type of learned features evolve with respect to their layer. The depth of the network was limited by the number of examples available for training and more complex layer connections (e.g. residual network [29], dense network [30]) were avoided to not ambiguate the analysis performed in this work. The number of feature maps (64) was kept uniform for each layer, allowing for easier comparisons of learned features across the convolutional layers. The filter size was 1×26 so that, similarly to the handcrafted features, the learned features are channel independent. Due to the selected filter size, the dimensions of feature maps at the final layer is 10×1 .

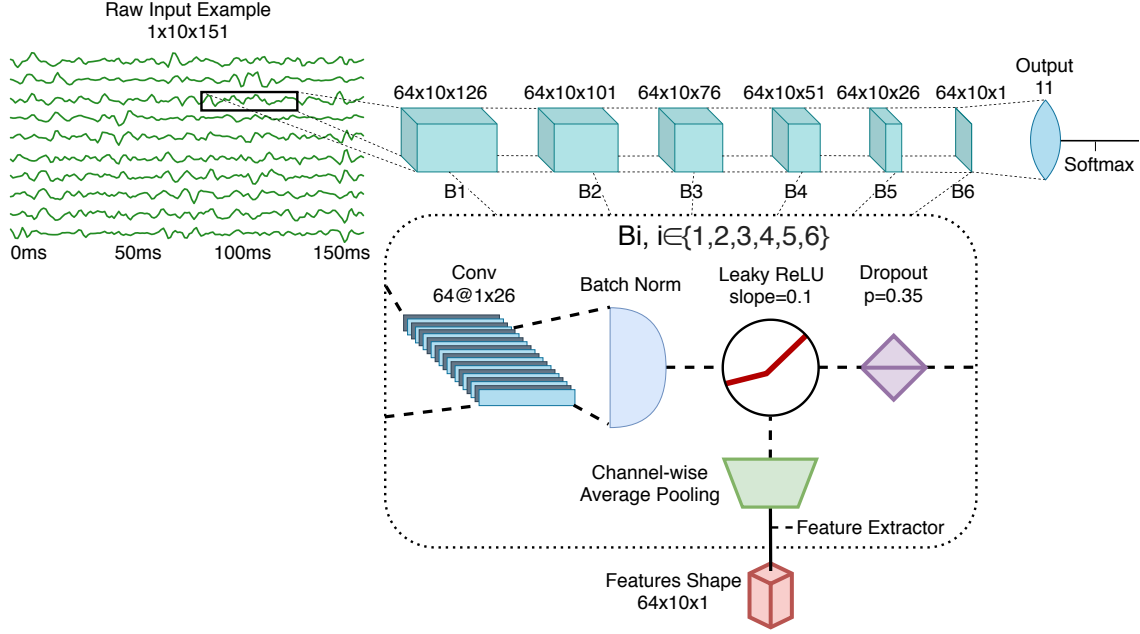


Figure 5.3 – The ConvNet’s architecture, employing 543,629 learnable parameters. In this figure, B_i refers to the i th feature extraction block ($i \in \{1, 2, 3, 4, 5, 6\}$). Conv refers to Convolutional layer. As shown, the feature extraction is performed after the non-linearity (leaky ReLU).

Adam [34] was employed to optimize the ConvNet with an initial learning rate of 0.0404709 and batch size of 512 (as used in [19]). The training dataset was divided into training and validation sets using the first three cycles and last cycle, respectively. Employing this validation set, learning rate annealing was applied with a factor of five and a patience of fifteen with early stopping applied when two consecutive annealings occurred without achieving a better validation loss.

For the purpose of the TDA, features maps were extracted after the non-linearity using per feature-map channel-wise average pooling. That is, the number of feature maps remained the same, but the feature map’s value per channel was averaged to a single scalar (as is common with handcrafted features).

Multi-Domain Adversarial Training

To better understand what type of features are commonly learned at each layer of the network, it is desirable that the model generalizes well across participants. This feature generality principle also motivates the design of the handcrafted features (presented in Section 5.6.2), as it would be impractical to create new features for each new participant. Learning a general feature representation across participants, however, cannot be achieved by simply aggregating the training data of all participants and then training a classifier normally. As, even when precisely controlling for electrode placement, cross-subject accuracy using standard learning

methods is poor ([13]). This problem is compounded by the fact that important differences exist between subjects of the *3DC Dataset* (i.e. position and rotation of the armband placed on the left or right arm).

Learning a participant-agnostic representation can be framed as a multi-domain learning problem [43]. In the context of sEMG-based gesture recognition, AdaBN, a domain adaptation algorithm presented in [36], was successfully employed as a way to learn a general representation across participants in [17, 18]. The hypothesis of AdaBN is that label-related information (i.e. hand gestures) will be contained within the network’s weights, while the domain-related information (i.e. participants) are stored in their BN statistics. Training is thus performed by sharing the weights of the network across the subjects dataset while tracking the BN statistics independently for each participant.

To inhibit the shared network’s weights from learning subject-specific representation, Domain-Adversarial Neural Networks (DANN) training [23] is employed. DANN is designed to learn domain-invariant features across two domains from the point of view of the desired task. The approach used by DANN to achieve this objective consists of adding a second head (referred to as the *domain classification head*) to the network presented in Section 5.6.3, which receives the output of block B6. The goal of this second head is to learn to discriminate between the domains. However, during backpropagation, the gradient computed from the domain loss is multiplied by a negative constant (set to -1 in this work) as it exits the domain classification head. This gradient reversal explicitly forces the feature distributions over the domains to be similar. Note that the backpropagation algorithm proceeds normally for the first head (gesture classification head). The loss function used for both heads is the cross-entropy loss. The two losses are combined as follows: $\mathcal{L}_y + \lambda\mathcal{L}_d$, where \mathcal{L}_y and \mathcal{L}_d are the prediction and domain loss, respectively (see Figure 5.4), while λ is a scalar that weights the domain loss (set to 0.1 in this work).

Using this approach, each participant of the 3DC Dataset represents a different domain ($n=22$). A direct application of DANN would thus initialize the domain classification head with 22 output neurons. This, however, could create a pitfall where the network is able to differentiate between the domains perfectly while simply predict one of the 21 other domains to maximize \mathcal{L}_d . Instead, the domain classification head is initialized with only two output neurons. At each epoch, a batch is created that contains examples from a single participant (this batch is referred to as the *source batch*, and is assigned the domain label 0). A second batch, referred to as the *target batch*, is also created that contains examples from one of the other participants selected at random, and is assigned the domain label 1. As every participants data is used as the source batch at each epoch, this ensures that the network is forced to learn a domain-independent feature representation. ADANN’s goal is thus to force the network to be unable to accurately associate a participant with their examples while achieving a highly discriminative gesture representation across all participants. During training, the BN statistics are tracked individually

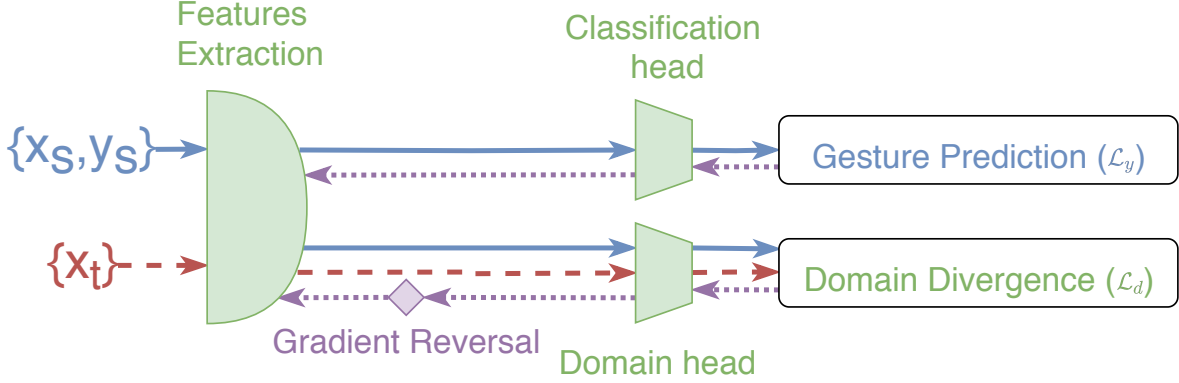


Figure 5.4 – Overview of the training steps of ADANN (identical to DANN) for one labeled batch from the source ($\{x_s, y_s\}$, blue lines) and one unlabeled batch from the target ($\{x_t\}$, red dashed lines). The purple dotted lines correspond to the backpropagated gradient. The gradient reversal operation is represented by the purple diamond.

for each subject. Therefore, when learning from a source or target batch, the network uses the BN statistics associated with the corresponding participant. Note that, by construction, the participant associated with the source is necessarily different from the participant associated with the target. Consequently, the network is fed the source and target batch consecutively (i.e. not both batch simultaneously). Also note that the BN statistics are updated only in association with the source batch to ensure equal training updates across all participants. For a given iteration, once the source and target batch are constructed, the training step proceeds as described for DANN (see Figure 5.4).

To assess the performance of the proposed MDL algorithm, two identical ConvNet (as described in Section 5.6.3) were created. One of the ConvNets was trained with ADANN, whereas the other used a standard training loop (i.e., aggregating the data from all participants), with both using the same hyperparameters. The networks trained with both methods were then tested on the test dataset with no participant-specific fine-tuning.

Learning Visualization

One of the main problems associated with deep learning is interpretability of how and why a model makes a prediction given a particular input. A first step in understanding a network prediction is through the visualization of the learned weights, feature maps and gradients resulting from a particular input. Consequently, several sophisticated visualisation techniques have been developed, which are aimed at facilitating a better comprehension of the hierarchical learning that takes place within a network [77, 82, 100]. One popular such technique is Guided Grad-CAM, which combines high resolution pixel-space gradient visualization and

class-discriminative visualization [73]. Guided Grad-CAM is thus employed to visualize how the ConvNet trained with ADANN makes its decisions, both on real examples from the *3DC Dataset* and on an artificially generated signals.

Given an image that was used to compute a forward pass in the network and a label y , the output of Guided Grad-CAM is calculated from four distinct steps (note that steps two and three are computed independently from each other using the output of step one):

1. Set all the gradients of the output neurons to zero, except for the gradient of the neuron associated with the label y (which is set to one) and name the gradient of the neuron of interest y^g .
2. Set all negative activations to zero. Then, perform backpropagation, but before propagating the gradient at each step, set all the negative gradients to zero again. Save the final gradients corresponding to the input image. This step corresponds to computing the guided backpropagation [82].
3. Let $F_{j,i}$ be the activation of the i th feature map of the j th layer with feature maps of the network. Select a layer F_j of interest (in this work F_j correspond to the rectified convolutional layer of B6). Backpropagate the signal from the output layer to $F_{j,i}$ (i.e. $\frac{\partial y^g}{\partial F_{j,i}}$). Then for each i compute the global average pooling of $\frac{\partial y^g}{\partial F_{j,i}}$ and name it $w_{j,i}$. Finally, compute: $ReLU(\sum_i w_{j,i} F_{j,i})$

This third step corresponds to computing the Gradient-weighted Class Activation Mapping (Grad-CAM) [74].

4. Finally, fuse the output of the two previous steps using point-wise multiplication to obtain the output of Guided Grad-CAM [73].

Learned feature classification

Similarly to [14], the learned features were extracted to train a Linear Discriminant Analysis (LDA) classifier to show the discriminative ability of the learned features. LDA was selected as it was shown to provide robust classification within the context of sEMG-based gesture recognition [12], does not require hyperparameter tuning, and creates linear boundaries within the input feature space. LDA was trained in a cross-subject framework on the training dataset and tested on the test dataset. For comparison purposes, LDA was also trained on the handcrafted features described in Section 5.6.2. Note that the implementation was from scikit-learn [52].

Regression Model

One method of highlighting the information content encoded throughout a network is to see how well known handcrafted features can be predicted from the network’s feature maps at different stages. This can be achieved using an added output neuron (*regression head*) at the feature extraction stage (i.e. after the non-linearity, but before the average pooling (before the green trapezoid of Figure 5.3)) of each block. The goal of this output is to map from the learned features to the handcrafted features of interest. As all the features considered in Section 5.6.2 are calculated channel-wise, only the information from the first sEMG channel (arbitrarily selected) of the feature maps will be fed to the regression head.

The training procedure to implement this is as follows: first, pre-train the network using ADANN (presented in Section 5.6.3). Second, freeze all the weights of the network, except for the weights associated with the regression head of the block of interest. The Mean Square Error (MSE) is then employed as the loss function with the target being the value of the handcrafted feature of interest from the first sEMG channel. Due to the stochastic nature of the algorithm, the training was performed 20 times for each participant and the results were given as the average MSE computed on the test dataset across of all participants. Note that the targets derived from multi-output feature extraction methods (e.g. Autoregressive Coefficients) corresponded to the first principal component returned by Principal Component Analysis (PCA) (where singular value decomposition was performed on the training and test set for the training and test phase, respectively).

5.6.4 Topological Data Analysis - Mapper

Conventional TDA methods such as Isomap [8] produce a low dimensional embedding by retaining geodesic distances between neighboring points. However, they often have limited topological stability [16] and lack the ability to produce a simplicial complex (a ball-and-stick simplification of the shape of the dataset) with size smaller than the original dataset [80]. The Mapper algorithm [80] is a TDA method that creates interpretable simplifications of high-dimensional data sets that remain true to the shape of the data set. Mapper can thus produce a stable representation of the topological shape of the dataset at a specified resolution, where the shape of the network has been simplified during a partial clustering stage. Further, the shape of the dataset is defined such that it is coordinate, deformation, and compression invariant. Consequently, this TDA algorithm can be employed to better understand how handcrafted and deep-learned features relate to one-another. In this work, Mapper is employed on three scenarios; (A), (B) and (C). In scenario (A), the algorithm only uses the handcrafted features as a way to validate the hyperparameters selected by cross-referencing the results with previous EMG works using Mapper [54, 10]. For scenario (B), only the learned features are

used to determine if features within the same block extract similar or dissimilar sources of information (i.e. the degree at which the features within the same block are dispersed across the topological network). Finally, in scenario (C), Mapper is applied to the combination of learned and handcrafted features to better understand their relationship and to provide new avenues of research for sEMG-based gesture recognition.

Sections 5.6.4 through 5.6.4, below, provide additional details about the approach, mathematical basis and implementation of Mapper in this work. Readers who are familiar with, or prefer to avoid these details, may jump directly to Section 6.9.

Mapper Algorithm

The construction of the topological network created using the Mapper algorithm can be seen as a five stage pipeline:

1. *prepare*: organize the data set to produce a point cloud of features in high dimensional space.
2. *lens*: filter the high dimensional data into a lower dimensional representation using a lens.
3. *resolution*: divide the filtration into a set of regions.
4. *partial clustering*: for each region, cluster the contents in the original high dimensional space.
5. *combine*: combine the region isolated clusters into a single topological network using common points across regions [24].

Mathematical definition of Mapper

A mathematical definition of the Mapper algorithm for feature extraction using a multi-channel recording device is as follows:

Let $\mathbf{x} \stackrel{\text{def}}{=} (\vec{x}_1, \dots, \vec{x}_C)$ be a series of samples for each C channels, where $\vec{x}_c \in \mathbb{R}^S, \forall c \in \{1, \dots, C\}$ and S is the length of a consecutive series of data. Define $\mathcal{X} \stackrel{\text{def}}{=} \{\mathbf{x}_n\}_{n=1}^N$ a set of N examples. Let also $\Phi \stackrel{\text{def}}{=} \{\phi_m\}_{m=1}^M$ be a set of M feature-generating functions of the form $\phi_m : \mathbb{R}^S \rightarrow \mathbb{R}$. Given $\mathbf{x}_{n,c}$ the c th element of $\mathbf{x}_n \in \mathcal{X}$, the resulting feature $f_{n,c}^m \in \mathbb{R}$ is obtained by applying ϕ_m such that $f_{n,c}^m \stackrel{\text{def}}{=} \phi_m(\mathbf{x}_{n,c})$. Consequently, the vector $\vec{f}_m \in \mathbb{R}^{N \times C}$ is obtained such that $\vec{f}_m \stackrel{\text{def}}{=} (f_{1,1}^m, f_{1,2}^m, \dots, f_{1,C}^m, f_{2,1}^m, f_{2,2}^m, \dots, f_{2,C}^m, \dots, f_{N,C}^m)$.

The first step of the Mapper algorithm is to consider $\mathcal{F} \stackrel{\text{def}}{=} \{\vec{f}_m\}_{m=1}^M$, the transformed data points from \mathcal{X} . Then define $\psi : \mathbb{R}^{N \times C} \rightarrow \mathbb{R}^Z$, with $0 < Z \ll N \times C$ and consider the set

$\mathcal{Z} \stackrel{\text{def}}{=} \{\psi(\vec{f}) | \vec{f} \in \mathcal{F}\}$. This dimensionality reduction ($N \times C \rightarrow Z$) is employed to reduce the computational cost of the rest of the Mapper algorithm and can be considered as a hyperparameter of the Mapper algorithm.

In the second step of the algorithm, define $\sigma : \mathbb{R}^Z \rightarrow \mathbb{R}^W$, with $0 < W \ll Z$ and consider the set $\mathcal{W} \stackrel{\text{def}}{=} \{\sigma(\vec{z}) | \vec{z} \in \mathcal{Z}\}$. In the literature [79], the function σ is called filter function and \mathcal{W} is the image or lens.

Third, let \mathfrak{C} be the smallest hypercube of \mathbb{R}^W which covers \mathcal{W} entirely. As \mathcal{X} is a finite set, each dimension of \mathfrak{C} is a finite interval. Let $k \in \mathbb{N}^*$, be a hyperparameter that subdivides \mathfrak{C} evenly into k^W smaller hypercubes. Note that the side lengths of these smaller hypercubes are $H = \frac{1}{k} \times$ the length size of \mathfrak{C} . Denotes \mathcal{V} the set of all vertices of these smaller hypercubes. Next, fix $D > H$ as another hyperparameter. For each $\vec{v} \in \mathcal{V}$, consider the hypercube $\mathfrak{c}_{\vec{v}}$ of length D centered on \vec{v} . A visualization of step 3 is given in Figure 5.5.

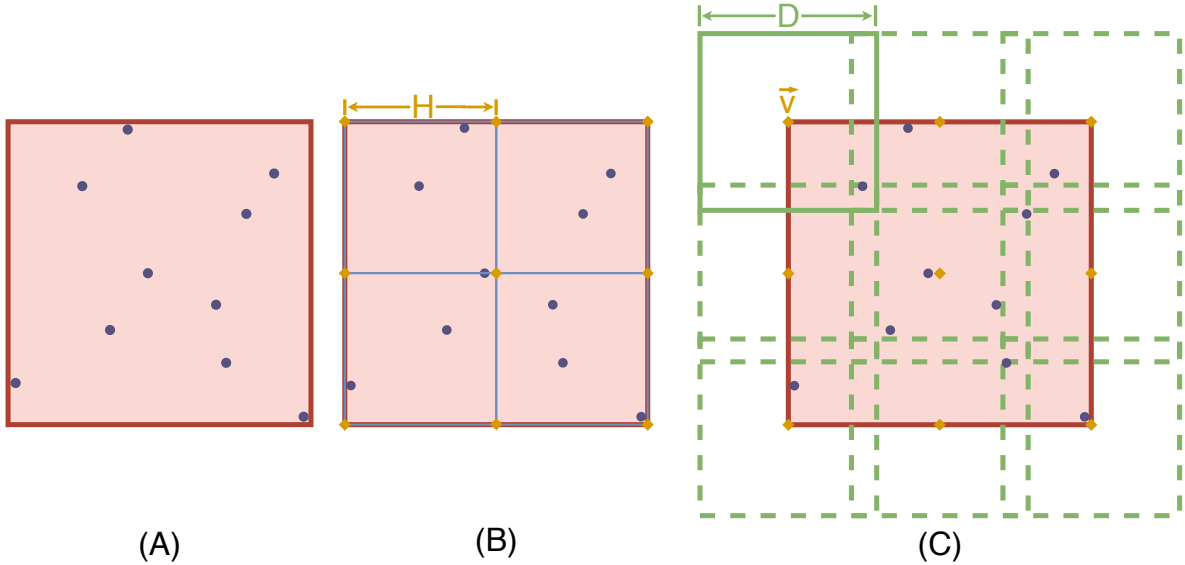


Figure 5.5 – An example of step 3 of the Mapper algorithm with $W = 2$. The purple dots represent the elements of \mathcal{W} . In (A), the red square corresponds to \mathfrak{C} . In (B), \mathfrak{C} is subdivided using k^2 squares of length H (with $k = 2$ in this case). The orange diamonds, in both (B) and (C), represent the elements of \mathcal{V} . Finally, the square $\mathfrak{c}_{\vec{v}}$ of length D is shown on the upper left corner of (C), overlapping other squares centered on other elements of \mathcal{V} (dotted lines).

Fourth, define $\mathcal{Z}_{\vec{v}} \stackrel{\text{def}}{=} \{\vec{z} \in \mathcal{Z} | \sigma(\vec{z}) \in \mathfrak{c}_{\vec{v}}\}$, the set of all elements of \mathcal{Z} that is projected in the hypercube $\mathfrak{c}_{\vec{v}}$. Let ξ be a clustering algorithm and $\xi(\mathcal{Z}_{\vec{v}})$ be the resulting set of clusters. Define \mathcal{B} as the set that consist of all so obtained clusters for all $\mathcal{Z}_{\vec{v}}$.

Fifth, compute the topological graph \mathcal{G} using each element of \mathcal{B} as a vertex and create an edge between vertices \mathcal{G}_i and \mathcal{G}_j ($i, j \in \{1, \dots, |\mathcal{B}|\}, i \neq j$) if $\mathcal{G}_i \cap \mathcal{G}_j \neq \emptyset$.

Mapper implementation within this work

In this work, as described in Section 5.6.1 the dataset was recorded using the 3DC Armband which offers 10 channel-recording ($C=10$) and an example is comprised of 151 data-points ($S=151$) for each channel. The number of considered features in scenarios (A), (B) and (C), are 79, 384, and 465, respectively. Note that multi-output feature extraction techniques (e.g. AR, HIST), consider each component of that vector as a separate feature. Each element of \mathcal{F} is obtained by computing the result of a feature from Section 5.6.2 (corresponding to $\phi_m()$ in the mathematical definition given previously) over each channel of each example of the *Training Dataset*. The dataset undergoes the first dimensionality reduction ($\Psi()$) using PCA [93], where the number of principal components used corresponds to 99% of the total variance. For scenarios (A), (B) and (C), 99% of the variance resulted in 44, 77, and 119 components, respectively, extracted from 971,860 channel-wise examples.

A second dimensionality reduction is then performed ($\sigma()$), referred to as the filter function, with the goal of representing meaningful characteristics of the relationship between features [80]. Within this study, *t*-Stochastic Neighborhood Embedding (t-SNE) [40] is used to encapsulate important local structure between features. The two-dimensional (2D) t-SNE lens was constructed with a perplexity of 30, as this configuration resulted in the most stable visualization over many repetitions (tested on scenario (A)). Using t-SNE as part of the Mapper algorithm instead of on its own leverages its ability to represent local structure while avoiding the use of a low-dimensional manifold to encapsulate global structure. Instead, the global structure is predominantly incorporated into the topological network produced by Mapper during the fifth stage.

The 2D lens was then segmented into a set of overlapped bins (the hypercubes centered on the elements of \mathcal{V}), called the cover. A stable topological network was obtained when each dimension was divided into 5 regions, forming a grid of 25 cubes that were overlapped by 65%. The number of regions correspond to the topological network’s resolution, while the overlap has an influence on the amount of connection formed between nodes [80].

Data points in each region are then clustered in isolation to provide insight into the local structure of the feature space (the elements of $\mathcal{Z}_{\bar{v}}$ correspond to the data-point of a specific region). For each region, Ward’s hierarchical clustering (ξ) was applied to construct a dendrogram that grouped similar features together according to a reduction in cluster variance [91].

Finally, the dendrograms produced using neighboring regions are combined to form the topological network (\mathcal{G}) using the features that lie in the overlapped area to construct the edges between the nodes.

The implementation of the Mapper algorithm was facilitated by a combination of the KeplerMapper [90] and the DyNeuSR (Dynamical Neuroimaging Spatiotemporal Representations) [24]

Python modules. An extended coverage of processing pipelines for time-series TDA is given in [53].

5.7 Results

5.7.1 Handcrafted features

Figure 5.6 shows the topological network produced using only the handcrafted features. The Kullback-Leibler divergence of the t-SNE embedding of the handcrafted features plateaued at 0.50, indicating that the perplexity and number of iterations used was appropriate for the dataset. The topological network consisted of 125 nodes and 524 edges.

The color of the nodes within the network indicates the percentage of members that belong to the feature group of interest ((A):SAP, (B): NLC), (C): FI, (D): TSM, and (E): UNI). The presence of an edge symbolizes common features present in the connected nodes, which can be used at a global scale to verify that functional groups (similar information) cluster together. Due to the topological nature of the graph, information similarity between nodes is measured using the number of edges that separate two nodes and not the length of the edges. Detailed interpretation of the TDA networks are given in the discussion.

5.7.2 Deep Features

The average cross-subject accuracy on the test set when using the proposed ADANN framework was $84.43\% \pm 0.05\%$. Using a Wilcoxon signed-rank test [92] with $n = 22$, and considering each participant as a separate dataset, this was found to significantly outperform ($p < 0.0001$) the average accuracy of $65.03\% \pm 0.08\%$ obtained when training the ConvNet conventionally. Furthermore, based on Cohen’s d , this difference in accuracy was considered to be huge [70]. The accuracy obtained per participant for each training method is given in Figure 5.7A, and the confusion matrices calculated on the gestures are shown in Figure 5.7B.

Figure 5.8A provides visualizations of the ConvNet trained with ADANN using Guided Grad-CAM for several examples from the *3DC Dataset*. These visualizations highlight what the network considers "important" (i.e., which part of the signals had the most impact in predicting a given class) for the prediction of a particular gesture.

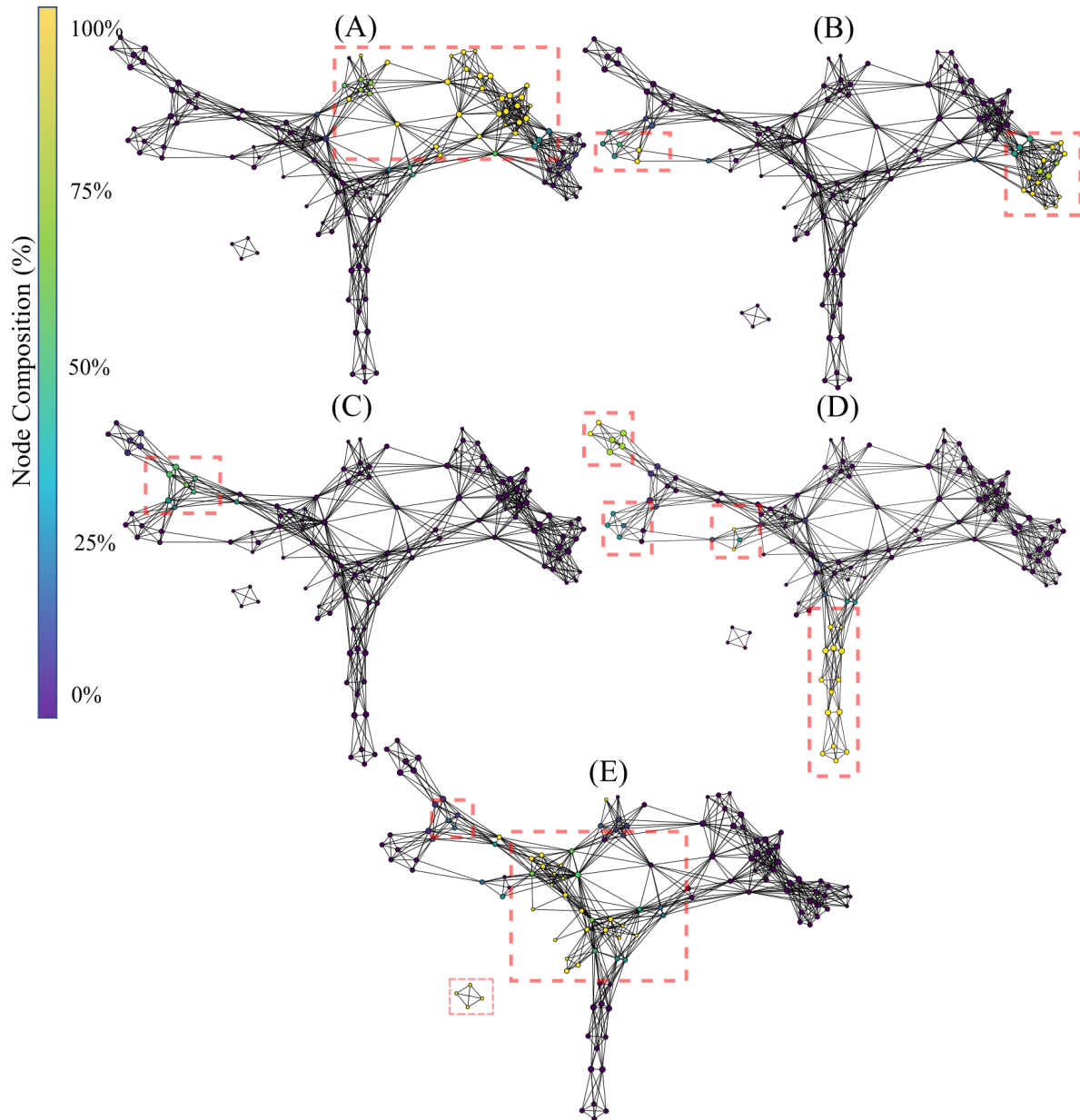


Figure 5.6 – Topological network generated exclusively for the handcrafted features, where nodes are colored to indicate percent composition of: (a) signal amplitude and power features (SAP), (b) nonlinear complexity (NLC), (c) frequency information features (FI), (d) time series modeling features (TSM), and (e) unique features (UNI). Dashed boxes highlight dense groupings of the specified functional group in each of the networks.

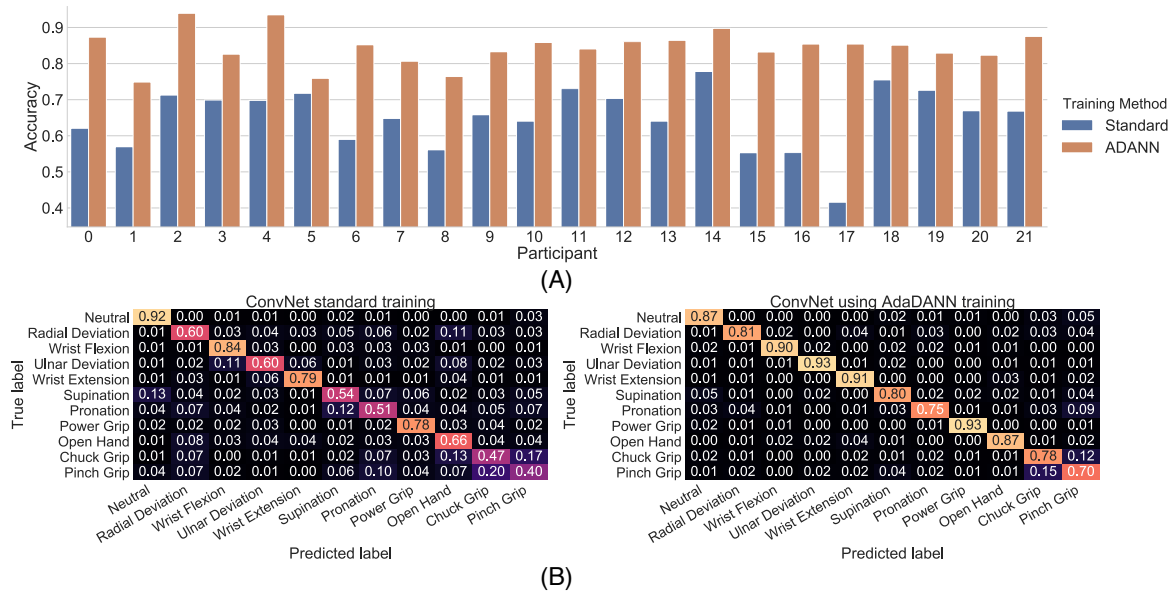


Figure 5.7 – Classification results of deep learning architectures. A) Per-participant test set accuracy comparison when training the network with and without ADANN, B) Confusion matrices on the test set for cross-subject training with and without ADANN.

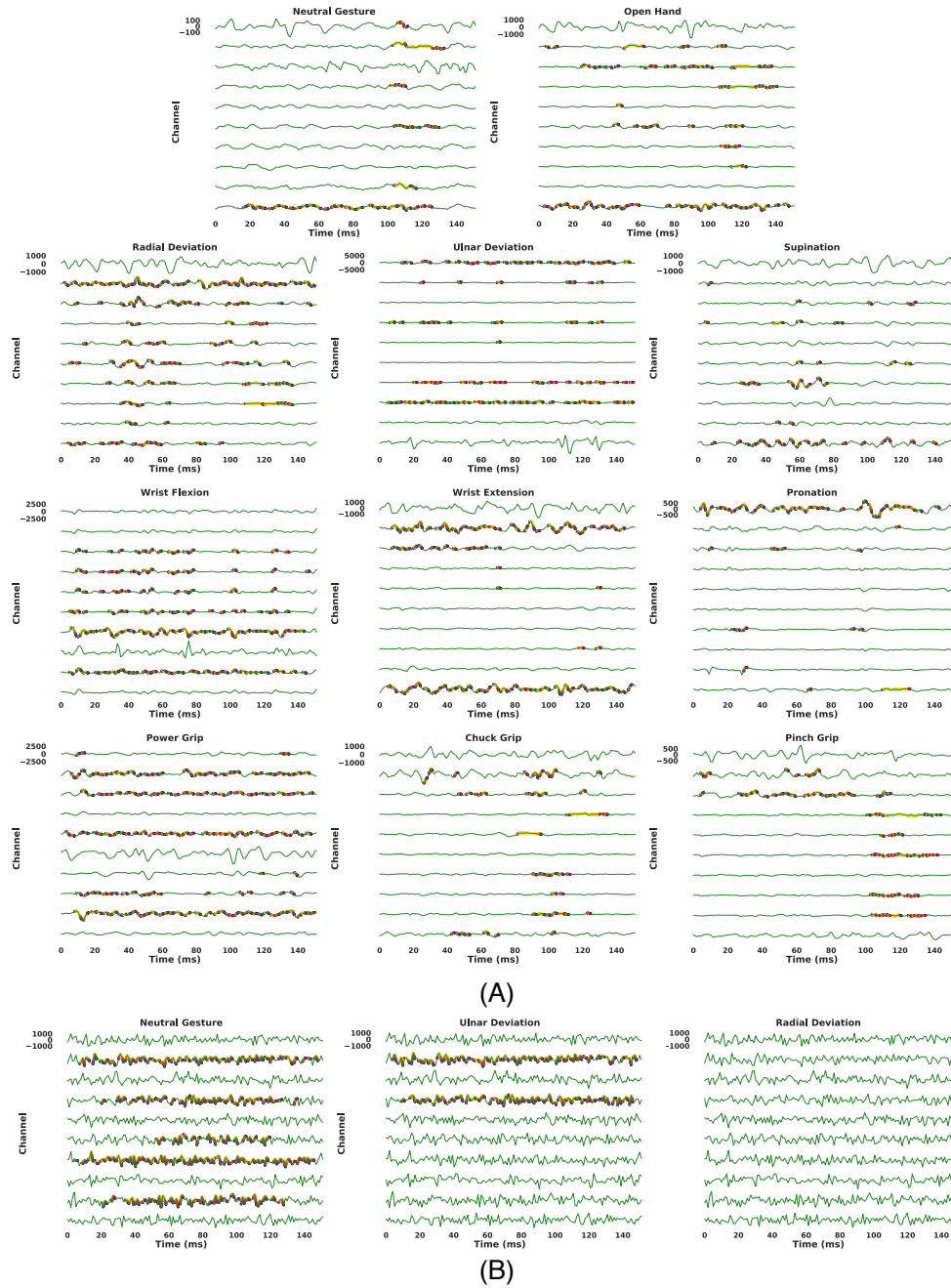


Figure 5.8 – Output of Guided Grad-CAM when asked to highlight specific gestures in an example. For all graphs, the y-axis of each channel are scaled to the same range of value (indicated on the first channel of each graph). Warmer colors indicate a higher 'importance' of a feature in the input space for the requested gesture. The coloring use a logarithmic scale. For visualization purposes, only features that are within three order of magnitudes to the most contributing feature are colored. (A) The examples shown are real examples and correspond to the same gestures that Guided Grad-CAM is asked to highlight. (B) A single example, generated using Gaussian noise of mean 0 and standard deviation 450, is shown three times. While the visualization algorithm does highlight features in the input space (when the requested gesture is not truly present in the input), the magnitude of these contributions is substantially smaller (half or less) than when the requested gesture is present in the input.

Instead of using Guided Grad-CAM to visualize how the network arrived at a decision for a known gesture, Figure 5.8B presents the results of the visualization algorithm when the network is told to find a gesture that is not present in the input. This is akin to using a picture of a cat as an input to the network and displaying the parts of the image that most resemble a giraffe. In Figure 5.8B, the input was randomly generated from a Gaussian distribution of mean 0 and standard deviation of 450 (chosen to have the same scale as the EMG signals of the *3DC Dataset*). For six of the eleven gestures (Radial Deviation, Wrist Extension, Supination, Open Hand, Chuck Grip and Pinch Grip) the network correctly identifies no relevant areas pertaining to these classes. While the network does highlight features in the input space associated with the other gestures, the magnitude of these contributions was substantially smaller (half or less) than when the requested gesture was actually present in the input signal.

The topological network produced using only the learned features is given in Figure 5.9. The color of the nodes within the network indicates the percentage of members that belong to the feature group of interests ((A): B1, (B): B2, (C): B3, (D): B4, (E): B5, and (F): B6). Interpretation of the TDA network follows the rational stated in Section 5.7.1. The Kullback-Leibler divergence of the t-SNE embedding of the handcrafted features plateaued at 0.37, again indicating that the perplexity and number of iterations used was appropriate for the dataset. The topological network consisted of 115 nodes and 672 edges.

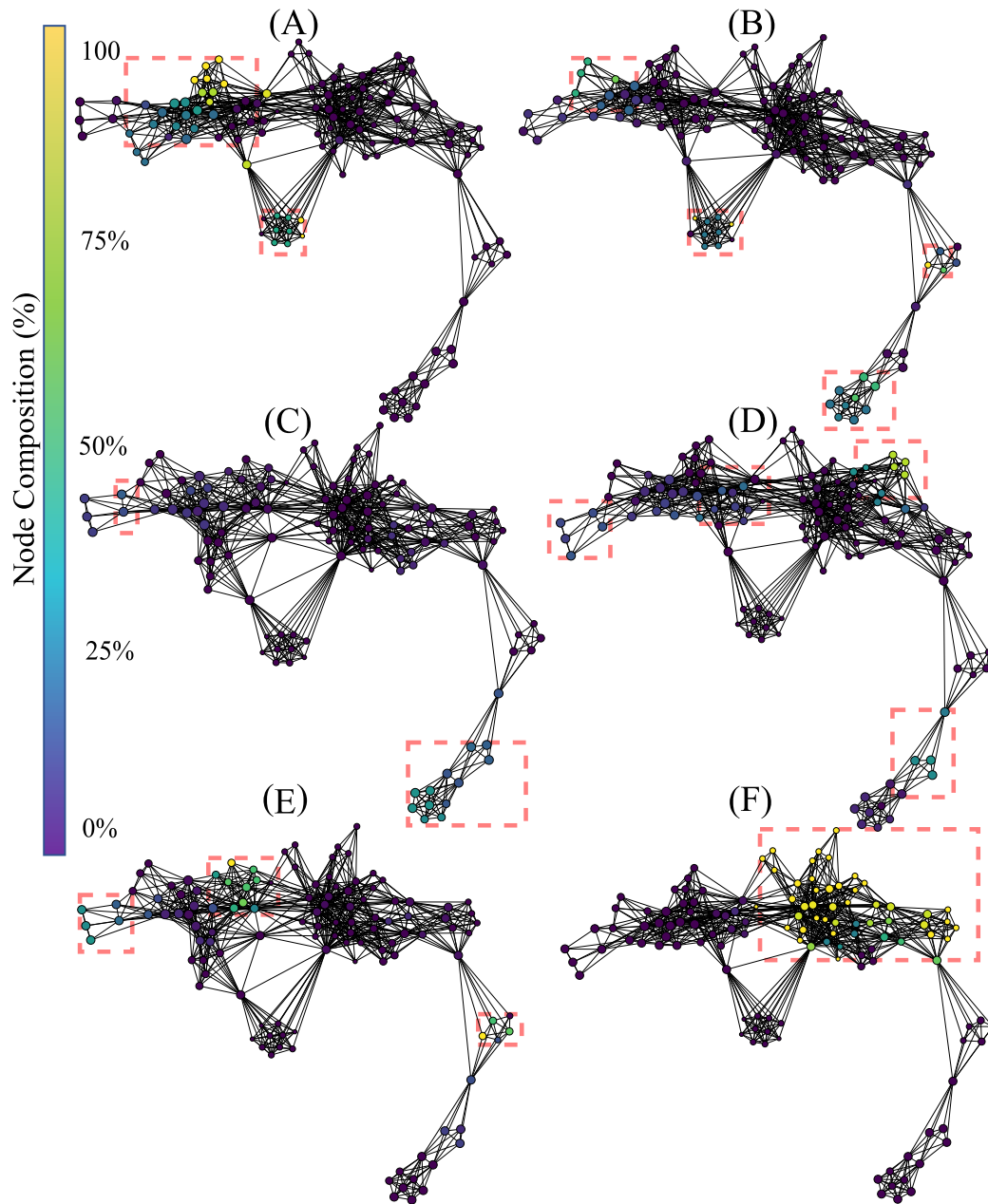


Figure 5.9 – Topological network generated for exclusively the learned features, where nodes are colored to indicate percent composition of: (A) Block 1’s features, (B) Block 2’s features, (C) Block 3’s features, (D) Block 4’s features, (E) Block 5’s features, and (F) Block 6’s features. Dashed boxes highlight dense groupings of the specified block features in each of the networks.

5.7.3 Hybrid Features

The topological network produced using both handcrafted and learned features is shown in Figure 5.10. The Kullback-Leibler divergence of the t-SNE embedding of all features plateaued at 0.53, again indicating that the perplexity and number of iterations used was appropriate for the dataset. The topological network consisted of 115 nodes and 770 edges. From this network, only a subset of nodes were occupied by both handcrafted and learned features. Those nodes were indicated in Figure 5.10.

The color of the nodes within the network indicates the percentage of members that belong to the feature group of interests (learned features). Information similarity was shown through a zoomed-in region of the network, where learned and handcrafted features clustered together. The feature members of the numbered nodes were listed in Table 5.2. Interpretation of the TDA network follows the rational stated in Section 5.7.1.

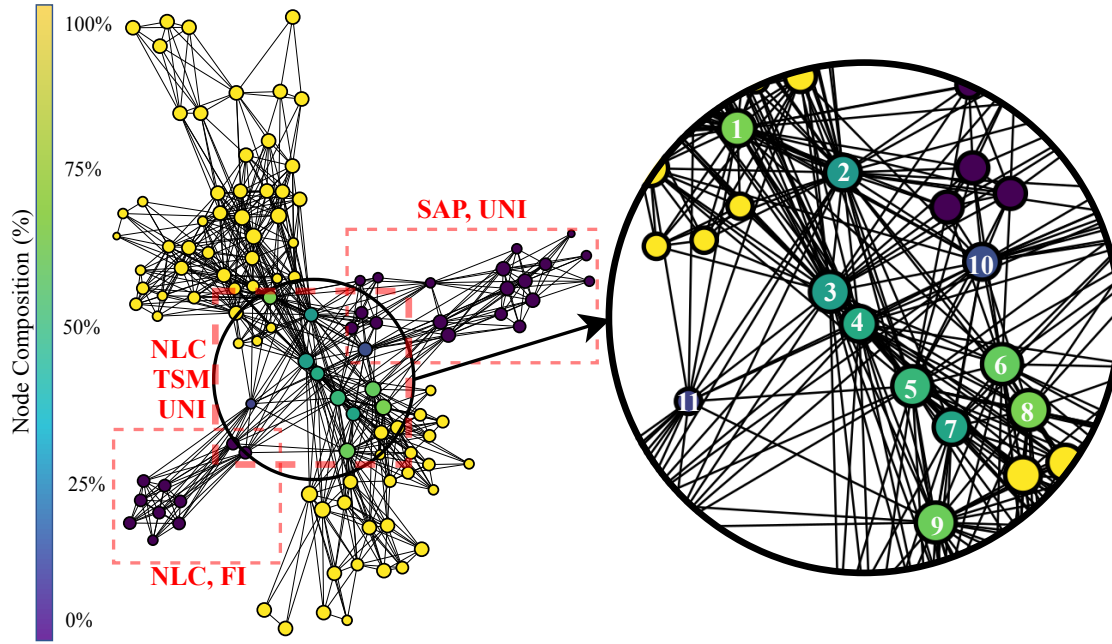


Figure 5.10 – Topological network generated for all features, where nodes were colored to indicate percent composition of learned features. The dashed boxes highlight dense grouping of handcrafted features with their associated type.

#	Summary	Members
1	TSM+LeF5	AR2 AR4 DAR2 DAR4 CC1 CC4 DCC1 DCC3 SNR 8xLeF1 1xLeF2 4xLeF4 10xLeF5 13xLe5
2	TSM+UNI+LeF6	APEN AR2 AR4 DAR2 DAR4 CC1 CC4 DCC1 DCC3 DCC4 CE DFA DPR HIST123 SKEW MAVS OHM PSDFD PSR SMR SNR VCF VFD 1xLeF1 3xLeF2 3xLeF5 21xLeF6
3	TSM+UNI+LeF6	APEN AR2 AR4 DAR2 DAR4 CC1 CC4 DCC1 DCC3 DCC4 CE DFA DPR HIST12 SKEW MAVS OHM PSDFD PSR SMR SNR VCF VFD 1xLeF1 1xLeF2 1xLeF5 27xLeF6
4	UNI+LeF6	APEN DCC4 CE DFA DPR HIST123 SKEW MAVS OHM PSDFD PSR SMR VCF VFD 2xLeF2 2xLeF5 21xLeF6
2	TSM+UNI+LeF6	APEN CC1 CC4 DCC4 CE DFA DPR HIST123 SKEW MAVS OHM PSDFD PSR SMR SNR VCF VFD 37xLeF6
6	TSM+UNI+LeF6	CC1 CC4 DCC4 CE DPR HIST123 SKEW MAVS PSDFD SMR SNR VCF VFD 5xLeF2 5xLeF4 1xLeF5 37xLeF6
7	UNI+LeF6	DCC4 CE DPR HIST123 SKEW MAVS PSDFD SMR VCF VFD 2xLeF2 15xLeF6
8	UNI+LeF6	DCC4 CE DPR HIST123 SKEW MAVS PSDFD SMR VCF VFD 5xLeF2 5xLeF4 1xLeF5 37xLeF6
9	UNI+LeF6	APEN DCC4 CE DFA DPR HIST2 SKEW MAVS OHM PSDFD PSR SMR VCF VFD 15xLeF2 36xLeF6
10	All Handcrafted+LeF6	APEN CC14 DCC4 CE DFA DPR HIST123 KURT SKEW M2 MAVS MAX MHW23 MTW123 MNP TTP OHM PSDFD PSR SM SMR SNR SSI TM DTM VAR DVARV VCF VFD 11xLeF6
11	NLC+LeF6	APEN SAMPEN BC KATZ 1xLeF6

Table 5.2 – Members of nodes labeled in Figure 5.6. LeF \mathbf{X} refers to a *Learned Feature* from block \mathbf{X} .

Table 5.3 shows the average accuracy (grouped by block for the learned features and by group for the handcrafted features) obtained when training an LDA on each feature and when using all features within a category (i.e. within a block or within a group of handcrafted feature). Note that for the learned features, PCA is applied to the feature map and the first component is employed to represent a given learned feature. Figure 5.11 shows examples of confusion matrices computed from the LDA classifications of singular features (both handcrafted and learned). Figure 5.11, also shows some confusion matrices obtained from the LDA’s classification result when using all features within a category.

	Single Feature		All Features
	Average Accuracy	STD	Accuracy
SAP	26.80%	7.0%	41.61%
FI	19.95%	2.87%	34.80%
NLC	22.32%	7.15%	31.49%
TSM	22.24%	3.33%	37.18%
UNI	15.32%	5.11%	48.37%
Block 1	28.49%	3.84%	74.59%
Block 2	28.28%	4.66%	78.26%
Block 3	28.90%	5.06%	79.19%
Block 4	29.21%	5.15%	78.77%
Block 5	28.18%	5.48%	79.23%
Block 6	26.62%	6.19%	81.38%

Table 5.3 – Accuracy obtained on the test set using the handcrafted features and the learned features from their respective block. The *Single Feature* accuracies are given as the average accuracy over all the features of their respective block/category.

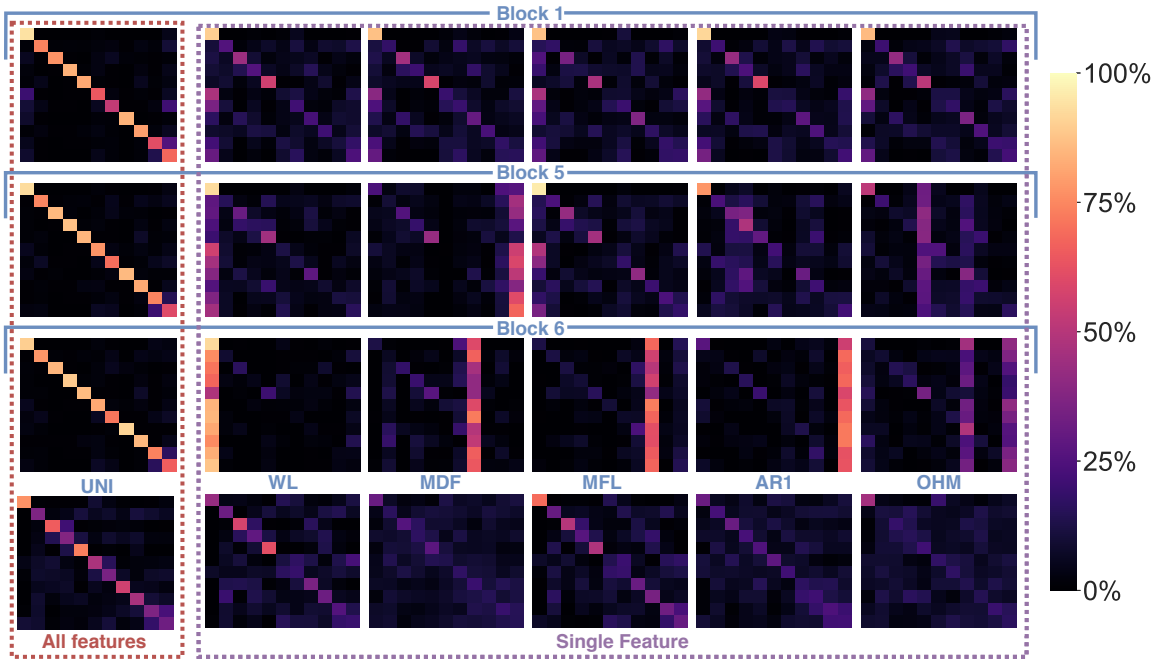


Figure 5.11 – Confusion matrices using the handcrafted features and the learned features from the first, penultimate and last block as input and a LDA as the classifier. The first column, denoted as *All features*, shows the confusion matrices when using all 64 learned features of Block 1, 5 and 6 respectively (from top to bottom) and the set of UNI handcrafted features. The next five columns, denoted as *Single Feature*, show the confusions matrices for handcrafted feature exemplars and from the same network’s blocks but when training the LDA on a single feature. The subset of learned features was selected as representative of the typical confusion matrices found at each block. The exemplars of the handcrafted features were selected from each handcrafted features’ category (in order: SAP, FI, NLC, TSM and UNI).

Figure 5.12 shows the average mean square error computed when regressing from the ConvNet’s learned features (see Section 5.6.3) to fifteen handcrafted features (three per Functional Group). Note that the mean squared error is obtained by computing the regression using only the output of the block of interest.

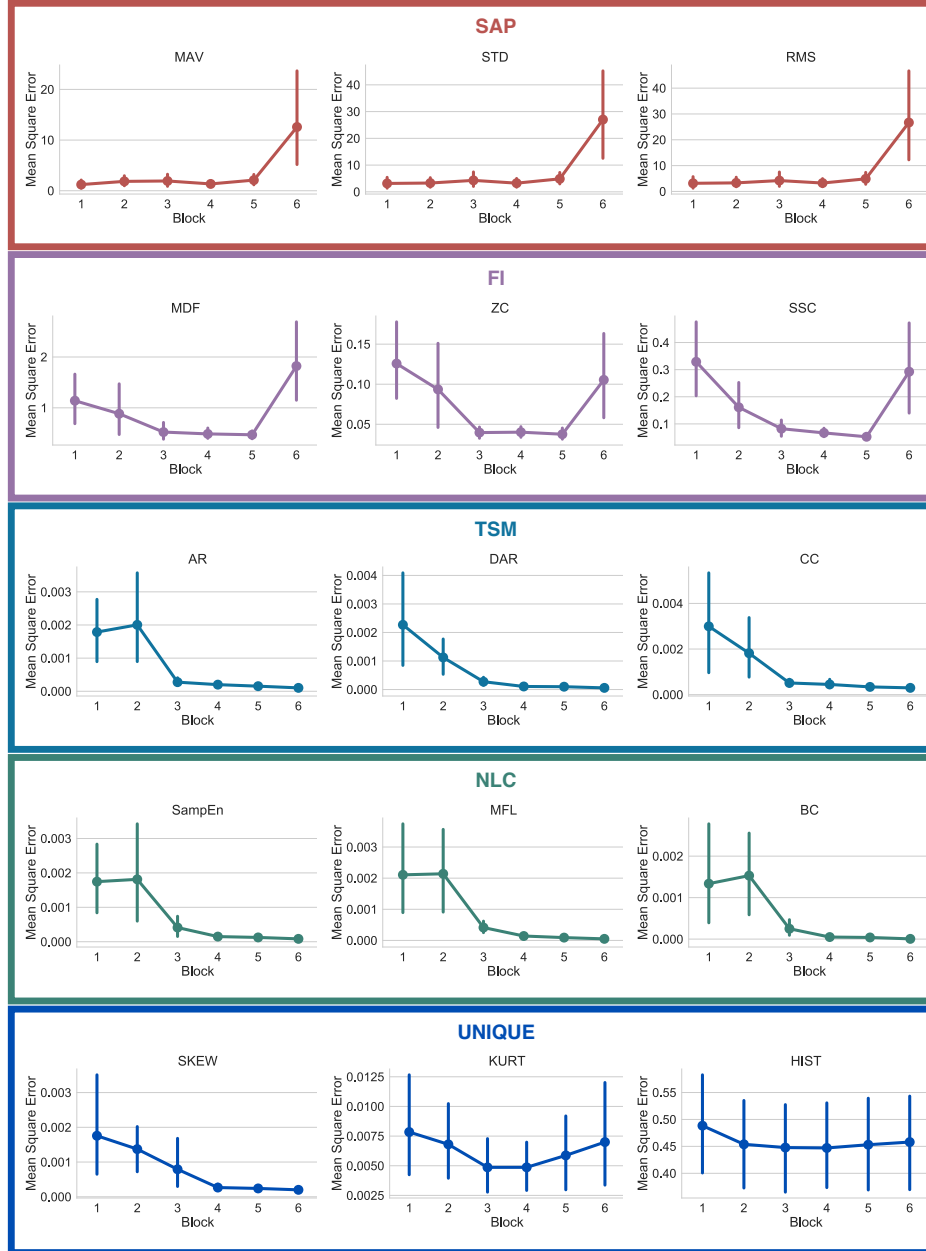


Figure 5.12 – Mean squared error of the regressions from learned features to handcrafted features, with respect to the number of blocks employed for the regression. The features are grouped with their respective functional groups.

5.8 Discussion

5.8.1 Handcrafted Features

The result of the Mapper algorithm applied to handcrafted features (see Figure 5.6) showed that the handcrafted features agglomerated mostly with their respective groups, and that the topological graph is Y-shaped. This shows that the hyperparameters selected in this work are consistent with those found in previous EMG literature [53, 10].

5.8.2 ADANN and Deep Learning Visualization

Figure 5.7B shows that training the network with ADANN outperforms the standard training method in cross-subject classification. One advantage of ADANN in the context of this work is that the weights of the network have strong incentives to be subject-agnostic. As such, the learned features extracted from the network can be thought of as general features (and to a certain extent subject-independent) for the task of sEMG-based hand gesture recognition.

Applying Guided Grad-CAM, as in 5.8, shows that the network mostly focuses on different channels for the detection of antagonist gestures. This suggests that the ConvNet was able to extract spatial features despite having access only to one dimensional convolutional kernels. Furthermore, it is notable that for all the examples given in Figure 5.8A, the most active channel was not the primary channel used for the gesture prediction. In fact, for the vast majority of gestures, the channel with the highest amplitude did not contribute in a meaningful way to the network’s prediction. This observation held true while looking at several other examples from the *3DC Dataset*. This might indicate that the common practice of placing the recording channel directly on the most prominent muscle for a given gesture within the context of gesture recognition may not be optimal. One could thus use the type of information provided by algorithms such as Guided Grad-CAM as another way of performing channel selection (instead of simply using classification accuracy). The absence of importance on amplitude characteristics is in contrast to conventional practices of handcrafted feature engineering - where the feature set typically relies heavily on amplitude characteristics. This perhaps explains the growing interest in handcrafted feature extraction techniques that do not capture amplitude information, such as TDPSD, that have been demonstrated to outperform conventional amplitude-reliant features in terms of accuracy and robustness to confounding factors [32].

When applying Guided Grad-CAM on a noise input (one where the target gesture is not present, as seen in Figure 5.8B), the reported activation level is substantially lower, and in some cases nonexistent. When the standard deviation of the Gaussian noise was increased by 33%, the network did not find any features resembling any gesture. This is most likely due to the fact that increasing the spread of the noise leads to a potentially greater gap in value between two adjacent data-points (reduced smoothness) fostering the condition for a more unrealistic signal. One could thus imagine training a generative adversarial network with the

discriminative function based on the activation level calculated by Guided Grad-CAM, and modulating the difficulty by augmenting the signal’s amplitude. This could facilitate training a network to not only be able to generate realistic, synthetic EMG signal, but also have the signal resemble actual gestures.

In contrast to the topological networks based on handcrafted features, those based on the learned features appear as a long flair with a loop. From Figure 5.9a, the learned features from block 1 are concentrated in the left segment of the flare, and the lower segment of the loop. From Figure 5.9b, the learned features from block 2 were located slightly more central to the network than the block 1 features. Additionally, a small subset of block 2 features appeared at the right segment of the flare, indicating a second distinct source of information was being harnessed. From Figure 5.9c, d, and e, the features of block 3, 4, and 5 relocate their concentration of features to converge in the center of the network. Finally from Figure 5.9f, the concentration of all block 6 features lies in the center of the network. Thus, it can be seen that learned features from the same block tend to cluster together and remain close in the map to adjacent blocks in the network. The only exception to this is from the first block to the second, where substantially different features were generated by the latter. This suggests that the first layer may serve almost as a preprocessing layer which conditions the signal for the other layers.

5.8.3 Hybrid Features Visualization

The topological network generated from using both the handcrafted and learned features (see Figure 5.10) followed two orthogonal axes with the handcrafted features on one and the learned features on the other. The middle of the graph (where the two axis intercept) is where any nodes containing both handcrafted and learned features are found. The vast majority of these nodes are populated by features from block 6 and the NLC, TSM and UNI functional groupings. No nodes in the graph contained both handcrafted features and features from block 3, suggesting that block 3 extracted features not captured by current feature designs. Conversely, no learned features shared a node with features from the FI family, suggesting that these features may not have been extracted by the network.

While this topological network informs the type of information encoded within each individual feature, it is important to note that information can still be present but encoded in a more complex way within the weights of the deep network. This information flow can be visualized from the regression graphs of Figure 5.12. Features from the SAP family are more easily predicted within the early blocks whereas features from the TSM and NLC family require the latter blocks of the network to achieve the best predictions. Interestingly, while features from the FI family did not share any nodes learned features, one can see that the deep network is able to better extract this type of information within the intermediary blocks. This indicates (from Figure 5.10, 5.12) that, while frequency information is not explicitly used by the ConvNet, this type of information is nonetheless indirectly used to compute the features from the latter

blocks. An example of a feature for which the ConvNet was unable to leverage its topology is the HIST (see Figure 5.12).

5.8.4 Understanding deep features predictions

The topological network of Figure 5.10 showed that the type of information encoded within the lower blocks of the ConvNet tended to be highly dissimilar to what the handcrafted features encoded. Interestingly, however, Figure 5.11 shows that the role fulfilled by these features is similar. That is, both the handcrafted and learned features (from the lower blocks) try to encode general properties that can distinguish between all classes. The confusion matrices obtained from training an LDA on a single feature highlight this behavior (see Figure 5.11 for some examples) as both the handcrafted features and the learned features (before the last block) are able to distinguish between gestures relatively equally. In contrast, the features extracted from the last block (and to a lesser extent from the penultimate block) have been optimized to be a gesture detector instead of a feature detector. A clear visual of this behavior is illustrated in Figure 5.11, where the main line highlighted in the confusion matrices from block 6 was a single column (corresponding to the prediction of a single gesture), instead of the typical diagonal. In other words, during training, the neurons of the final block are encoded to have maximum activation when a particular class was provided in the input window and minimum activation when other classes were provided; effectively creating a one-versus-all (OVA) classifier. This behavior is consistent with the feature visualization literature found in image classification and natural language processing, where semantic dictionaries or saliency maps have depicted neuron representations becoming more abstract at later layers [77, 35]. This also explains why the features from the last block obtained the worst average accuracy when taken individually while achieving the highest accuracy as a group (see Table 5.3). That is, as each feature map of the last layer tries to detect a particular gesture, its activation for the other gestures should be minimal, making the distinction between the other gestures significantly harder. The final decision layer of the network can then be thought of as a weighted average of these OVA classifiers to maximize the performance of the learned feature maps. Note that in Table 5.3, the lower accuracies obtained from the handcrafted features as a group were expected as each feature within the same family provides similar type of information, even more so than the learned features of the network (as seen in Figure 5.6, 5.9, 5.10). Overall, the best performing handcrafted feature set as a group was the features from the UNI family despite the fact that they were the worst on average when alone. This is most likely due to the fact that by definitions, features within this family are more heterogeneous.

5.9 Conclusion

This paper presents the first in-depth analysis of features learned using deep learning for EMG-based hand gesture recognition. The type of information encoded within learned features and their relationship to handcrafted features were characterized employing a mixture of topological data analysis (Mapper), network interpretability visualization (Guided Grad-CAM), machine learning (feature classification prediction) and by visualizing the information flow using feature regression. As a secondary, but significant contribution, this work presented ADANN, a novel multi-domain training algorithm particularly suited for EMG-based gesture recognition shown to significantly outperform traditional training on cross-subject classification accuracy.

This manuscript paves the way for hybrid classifiers that contain both learned and handcrafted features. An ideal application for the findings of this work would rely on a mix of handcrafted features and learned features taken from all four extremities of the hybrid topological network, and at the center to provide complementary, and general features to the classifier. A network could then be trained to augment its sensitivity to similar classes. For example, to alleviate ambiguity between pinch grip and chuck grip, a learned feature that encodes the one-versus-all information of pinch grip could be included into the original feature set or into an otherwise handcrafted only feature set. Alternatively, handcrafted feature extraction stages may be installed within the deep learning architecture by means of neuroevolution of augmenting topologies [15], a genetic algorithm that optimizes the weights and connections of deep learning architectures.

The main limitation of this study was the use of a single architecture to generate the learned features. Though this architecture was chosen to be representative of current practices in myoelectric control and be extensible to other applications, the current work did study the impact of varying the number of blocks and the composition of these block on the different experiments. Additionally, although the set of handcrafted features was selected to be comprehensive over the sources of information available from the EMG signal, explicit time-frequency features such as those based on spectrograms and wavelet were not included in the current work, as they were ill-adapted to the framework employed in this study. Furthermore, an analysis including a larger amount of gestures should also be conducted. Importantly, these results are presented for a single 1D electrode array, and may not be representative of larger 2D arrays such as those used in high density EMG applications. Similarly, explicit spatio-temporal features, such as coherence between electrodes, were not explored, and the convolutional kernels were restricted to 1D (although as seen in Figure 5.8A the network was still able to learn spatial information to a certain extent). Omitting these type of complex features was a design choice as this work represents a first step in understanding and characterizing learned features within the context of EMG signal. As such, using this manuscript as a basis, future works should study the impact of diverse architectures on the type of learned features and will incorporate

spatio-temporal features (both handcrafted and from 2D convolutional kernels). Additionally, formal feature set generation and hybrid classifiers should be investigated using the tools presented in this work.

Conflict of Interest Statement

The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of the data; in the writing of the manuscript; or in the decision to publish the results.

Author Contributions

Conceptualization: UCA, EC, AP, FL, BG and ES; methodology, UCA, EC, AP, FL and ES; software, UCA and EC; validation, UCA and EC; formal analysis, UCA, EC, AP, FL and ES; investigation, UCA, EC, AP, FL and ES; resources, FL, BG, ES; data curation, UCA and EC; writing—original draft preparation, UCA and EC; writing—review and editing, UCA, EC, AP, FL, BG and ES; visualization, UCA and EC; supervision, FL, BG and ES; project administration, UCA, EC, AP, FL, BG and ES; funding acquisition, UCA, AP, FL, BG and ES;

Funding

This research was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) [funding reference numbers 401220434, 376091307, 114090], the Institut de recherche Robert-Sauvé en santé et en sécurité du travail (IRSST) and the Canada Research Chair in Smart Biomedical Microsystems [funding reference number 950-232064]. Cette recherche a été financée par le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG) [numéros de référence 401220434, 376091307, 114090].

Acknowledgments

The authors would like to thank Gabriel Dubé for his valuable input in relation to the Mapper algorithm.

Data Availability Statement

The datasets analyzed and the source code for this study can be found at the following link: https://github.com/UlysseCoteAllard/sEMG_handCraftedVsLearnedFeatures.

5.10 Bibliography

- [1] Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. Domain-adversarial neural networks. *arXiv preprint arXiv:1412.4446*, 2014.
- [2] Ali H Al-Timemy, Rami N Khushaba, Guido Bugmann, and Javier Escudero. Improving the performance against force variation of emg controlled multifunctional upper-limb prostheses for transradial amputees. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 24(6):650–661, 2015.
- [3] Ulysse Côté Allard, François Nougrou, Cheikh Latyr Fall, Philippe Giguère, Clément Gosselin, François Laviolette, and Benoit Gosselin. A convolutional neural network for robotic arm guidance using semg based frequency-features. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2464–2470. IEEE, 2016.
- [4] Md Zahangir Alom, Tarek M Taha, Christopher Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Brian C Van Esesn, Abdul A S Awwal, and Vijayan K Asari. The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv preprint arXiv:1803.01164*, 2018.
- [5] Vidar Andersen, Marius S Fimland, Dag-Andrè Mo, Vegard M Iversen, Torbjørn Vederhus, Lars R Rockland Hellebø, Kristina I Nordaune, and Atle H Saeterbakken. Electromyographic comparison of barbell deadlift, hex bar deadlift, and hip thrust exercises: A cross-over study. *The Journal of Strength & Conditioning Research*, 32(3):587–593, 2018.
- [6] Sridhar Poosapadi Arjunan and Dinesh Kant Kumar. Decoding subtle forearm flexions using fractal features of surface electromyogram from single and multiple sensors. *Journal of neuroengineering and rehabilitation*, 7(1):53, 2010.
- [7] Manfredo Atzori, Matteo Cognolato, and Henning 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.
- [8] Mukund Balasubramanian and Eric L Schwartz. The isomap algorithm and topological stability. *Science*, 295(5552):7–7, 2002.
- [9] Velislav Batchvarov and Marek Malik. Individual patterns of qt/rr relationship. *Cardiac electrophysiology review*, 6(3):282–288, 2002.
- [10] E. Campbell, A. Phinyomark, A. H. Al-Timemy, R. N. Khushaba, G. Petri, and E. Scheme. Differences in EMG feature space between able-bodied and amputee subjects for myoelectric control. In *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 33–36, March 2019.

- [11] Evan Campbell, Angkoon Phinyomark, and Erik Scheme. Feature extraction and selection for pain recognition using peripheral physiological signals. *Frontiers in Neuroscience*, 13:437, 2019.
- [12] Evan D Campbell, Angkoon Phinyomark, and Erik Scheme. Linear discriminant analysis with bayesian risk parameters for myoelectric control. In *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP) (GlobalSIP 2019)*, Ottawa, Canada, November 2019.
- [13] Claudio Castellini, Angelo Emanuele Fiorilla, and Giulio Sandini. Multi-subject/daily-life activity emg-based control of mechanical hands. *Journal of neuroengineering and rehabilitation*, 6(1):41, 2009.
- [14] Hongfeng Chen, Yue Zhang, Gongfa Li, Yinfeng Fang, and Honghai Liu. Surface electromyography feature extraction via convolutional neural network. *International Journal of Machine Learning and Cybernetics*, pages 1–12, 2019.
- [15] L. Chen and D. Alahakoon. Neuroevolution of augmenting topologies with learning for data classification. In *2006 International Conference on Information and Automation*, pages 367–371, Dec 2006.
- [16] Heeyoul Choi and Seungjin Choi. Robust kernel isomap. *Pattern Recognition*, 40(3):853–862, 2007.
- [17] Ulysse Cote-Allard, Cheikh Latyr Fall, Alexandre Campeau-Lecours, Clément Gosselin, François Laviolette, and Benoit Gosselin. Transfer learning for semg hand gestures recognition using convolutional neural networks. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1663–1668. IEEE, 2017.
- [18] Ulysse Côté-Allard, Cheikh Latyr Fall, Alexandre Drouin, Alexandre Campeau-Lecours, Clément Gosselin, Kyrre Glette, François Laviolette, and Benoit Gosselin. Deep learning for electromyographic hand gesture signal classification using transfer learning. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(4):760–771, 2019.
- [19] Ulysse Côté-Allard, Gabriel Gagnon-Turcotte, François Laviolette, and Benoit Gosselin. A low-cost, wireless, 3-d-printed custom armband for semg hand gesture recognition. *Sensors*, 19(12):2811, 2019.
- [20] Sijiang Du and Marko Vuskovic. Temporal vs. spectral approach to feature extraction from prehensile emg signals. In *Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration, 2004. IRI 2004.*, pages 344–350. IEEE, 2004.
- [21] Yarín Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

- [22] Chuang Gan, Naiyan Wang, Yi Yang, Dit-Yan Yeung, and Alex G Hauptmann. Devnet: A deep event network for multimedia event detection and evidence recounting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2568–2577, 2015.
- [23] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.
- [24] Caleb Geniesse, Olaf Sporns, Giovanni Petri, and Manish Saggar. Generating dynamical neuroimaging spatiotemporal representations (dyneusr) using topological data analysis. *Network Neuroscience*, 3(3):763–778, 2019.
- [25] J Andrew Gitter and M Joseph Czerniecki. Fractal analysis of the electromyographic interference pattern. *Journal of Neuroscience Methods*, 58(1-2):103–108, 1995.
- [26] Laura Guidetti, Gianfranco Rivellini, and Francesco Figura. Emg patterns during running: Intra-and inter-individual variability. *Journal of Electromyography and Kinesiology*, 6(1):37–48, 1996.
- [27] Vineet Gupta, Srikanth Suryanarayanan, and Narender P Reddy. Fractal analysis of surface emg signals from the biceps. *International journal of medical informatics*, 45(3):185–192, 1997.
- [28] Mark Halaki and Karen Ginn. Normalization of emg signals: to normalize or not to normalize and what to normalize to? In *Computational intelligence in electromyography analysis-a perspective on current applications and future challenges*. IntechOpen, 2012.
- [29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [30] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.
- [31] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
- [32] Rami N. Khushaba, Ali Al-Timemy, Sarath Kodagoda, and Kianoush Nazarpour. Combined influence of forearm orientation and muscular contraction on emg pattern recognition. *Expert Systems with Applications*, 61:154 – 161, 2016.
- [33] Kang Soo Kim, Heung Ho Choi, Chang Soo Moon, and Chi Woong Mun. Comparison of k-nearest neighbor, quadratic discriminant and linear discriminant analysis in classification

- of electromyogram signals based on the wrist-motion directions. *Current applied physics*, 11(3):740–745, 2011.
- [34] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
 - [35] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.
 - [36] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*, 2016.
 - [37] Xueliang Liu, Rongjie Zhang, Zhijun Meng, Richang Hong, and Guangcan Liu. On fusing the latent deep cnn feature for image classification. *World Wide Web*, 22(2):423–436, 2019.
 - [38] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.
 - [39] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.
 - [40] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.
 - [41] Paul McCool, Graham D Fraser, Adrian DC Chan, Lykourgos Petropoulakis, and John J Soraghan. Identification of contaminant type in surface electromyography (emg) signals. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(4):774–783, 2014.
 - [42] Jed A Meltzer, Michiro Negishi, Linda C Mayes, and R Todd Constable. Individual differences in eeg theta and alpha dynamics during working memory correlate with fmri responses across subjects. *Clinical Neurophysiology*, 118(11):2419–2436, 2007.
 - [43] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4293–4302, 2016.
 - [44] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4293–4302, 2016.
 - [45] Loris Nanni, Stefano Ghidoni, and Sheryl Brahnam. Handcrafted vs. non-handcrafted features for computer vision classification. *Pattern Recognition*, 71:158–172, 2017.

- [46] Monica Nicolau, Arnold J. Levine, and Gunnar Carlsson. Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. *Proceedings of the National Academy of Sciences*, 108(17):7265–7270, 2011.
- [47] Mohammadreza Asghari Oskoei and Huosheng Hu. Ga-based feature subset selection for myoelectric classification. In *2006 IEEE International Conference on Robotics and Biomimetics*, pages 1465–1470. IEEE, 2006.
- [48] Mohammadreza Asghari Oskoei and Huosheng Hu. Myoelectric control systems—a survey. *Biomedical signal processing and control*, 2(4):275–294, 2007.
- [49] Mohammadreza Asghari Oskoei and Huosheng Hu. Support vector machine-based classification scheme for myoelectric control applied to upper limb. *IEEE transactions on biomedical engineering*, 55(8):1956–1965, 2008.
- [50] Sang-Hui Park and Seok-Pil Lee. Emg pattern recognition based on artificial intelligence techniques. *IEEE transactions on Rehabilitation Engineering*, 6(4):400–405, 1998.
- [51] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.
- [52] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [53] A. Phinyomark, E. Ibanez-Marcelo, and G. Petri. *Topological Data Analysis of Biomedical Big Data*, volume 1 of 1, chapter 11, pages 209–233. Boca Raton: CRC Press, The address of the publisher, 1 edition, 7 2018.
- [54] Angkoon Phinyomark, Rami N. Khushaba, Esther Ibáñez-Marcelo, Alice Patania, Erik Scheme, and Giovanni Petri. Navigating features: a topologically informed chart of electromyographic features space. *Journal of The Royal Society Interface*, 14(137), 2017.
- [55] Angkoon Phinyomark, Montri Phothisonothai, Pornchai Phukpattaranont, and Chusak Limsakul. Critical exponent analysis applied to surface emg signals for gesture recognition. *Metrology and Measurement Systems*, 18(4):645–658, 2011.
- [56] Angkoon Phinyomark, Montri Phothisonothai, Pornchai Phukpattaranont, and Chusak Limsakul. Evaluation of movement types and electrode positions for emg pattern classification based on linear and non-linear features. *Eur. J. Sci. Res*, 62(1):24–34, 2011.

- [57] Angkoon Phinyomark, Pornchai Phukpattaranont, and Chusak Limsakul. A review of control methods for electric power wheelchairs based on electromyography signals with special emphasis on pattern recognition. *IETE Technical Review*, 28(4):316–326, 2011.
- [58] Angkoon Phinyomark, Pornchai Phukpattaranont, and Chusak Limsakul. Feature reduction and selection for emg signal classification. *Expert systems with applications*, 39(8):7420–7431, 2012.
- [59] Angkoon Phinyomark, Pornchai Phukpattaranont, and Chusak Limsakul. Fractal analysis features for weak and single-channel upper-limb emg signals. *Expert Systems with Applications*, 39(12):11156–11163, 2012.
- [60] Angkoon Phinyomark, Pornchai Phukpattaranont, Chusak Limsakul, and Montri Phothisonothai. Electromyography (emg) signal classification based on detrended fluctuation analysis. *Fluctuation and Noise Letters*, 10(03):281–301, 2011.
- [61] Angkoon Phinyomark, Franck Quaine, Sylvie Charbonnier, Christine Serviere, Franck Tarpin-Bernard, and Yann Laurillau. Emg feature evaluation for improving myoelectric pattern recognition robustness. *Expert Systems with applications*, 40(12):4832–4840, 2013.
- [62] Angkoon Phinyomark, Franck Quaine, Sylvie Charbonnier, Christine Serviere, Franck Tarpin-Bernard, and Yann Laurillau. Emg feature evaluation for improving myoelectric pattern recognition robustness. *Expert Systems with applications*, 40(12):4832–4840, 2013.
- [63] Angkoon Phinyomark and Erik Scheme. Emg pattern recognition in the era of big data and deep learning. *Big Data and Cognitive Computing*, 2(3):21, 2018.
- [64] Angkoon Phinyomark and Erik Scheme. A feature extraction issue for myoelectric control based on wearable emg sensors. In *2018 IEEE Sensors Applications Symposium (SAS)*, pages 1–6. IEEE, 2018.
- [65] Soujanya Poria, Erik Cambria, and Alexander Gelbukh. Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2539–2544, 2015.
- [66] Zhang Qingju and Luo Zhizeng. Wavelet de-noising of electromyography. In *2006 International Conference on Mechatronics and Automation*, pages 1553–1558. IEEE, 2006.
- [67] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Efficient parametrization of multi-domain deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8119–8127, 2018.

- [68] Manish Saggar, Olaf Sporns, Javier Gonzalez-Castillo, Peter Bandettini, Gunnar Carlsson, Gary Glover, and Reiss Allan. Towards a new approach to reveal dynamical organization of the brain using topological data analysis. *Nature Communications*, 9(1):1399, 2018.
- [69] T Scott Saponas, Desney S Tan, Dan Morris, and Ravin Balakrishnan. Demonstrating the feasibility of using forearm electromyography for muscle-computer interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 515–524. ACM, 2008.
- [70] Shlomo S Sawilowsky. New effect size rules of thumb. *Journal of Modern Applied Statistical Methods*, 8(2):26, 2009.
- [71] Erik Scheme and Kevin Englehart. Electromyogram pattern recognition for control of powered upper-limb prostheses: state of the art and challenges for clinical use. *Journal of Rehabilitation Research & Development*, 48(6), 2011.
- [72] Erik Scheme and Kevin Englehart. Electromyogram pattern recognition for control of powered upper-limb prostheses: state of the art and challenges for clinical use. *Journal of Rehabilitation Research & Development*, 48(6), 2011.
- [73] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017.
- [74] Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*, 2016.
- [75] Pierre Sermanet, Koray Kavukcuoglu, Soumith Chintala, and Yann LeCun. Pedestrian detection with unsupervised multi-stage feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3626–3633, 2013.
- [76] Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. *arXiv preprint arXiv:1802.08735*, 2018.
- [77] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.
- [78] C Sinderby, L Lindstrom, and AE Grassino. Automatic assessment of electromyogram quality. *Journal of Applied Physiology*, 79(5):1803–1815, 1995.

- [79] Gurjeet Singh, Facundo Mémoli, and Gunnar E Carlsson. Topological methods for the analysis of high dimensional data sets and 3d object recognition. In *SPBG*, pages 91–100, 2007.
- [80] Gurjeet Singh, Facundo Mémoli, and Gunnar E Carlsson. Topological methods for the analysis of high dimensional data sets and 3d object recognition. In *SPBG*, pages 91–100, 2007.
- [81] Lauren H Smith, Levi J Hargrove, Blair A Lock, and Todd A Kuiken. Determining the optimal window length for pattern recognition-based myoelectric control: balancing the competing effects of classification error and controller delay. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 19(2):186–192, 2010.
- [82] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.
- [83] David St-Onge, Ulysse Côté-Allard, Kyrre Glette, Benoit Gosselin, and Giovanni Beltrame. Engaging with robotic swarms: Commands from expressive motion. *ACM Transactions on Human-Robot Interaction (THRI)*, 8(2):11, 2019.
- [84] Mehran Talebinejad, Adrian DC Chan, Ali Miri, and Richard M Dansereau. Fractal analysis of surface electromyography signals: a novel power spectrum-based method. *Journal of electromyography and kinesiology*, 19(5):840–850, 2009.
- [85] Sirinee Thongpanja, Angkoon Phinyomark, Huosheng Hu, Chusak Limsakul, and Pornchai Phukpattaranont. The effects of the force of contraction and elbow joint angle on mean and median frequency analysis for muscle fatigue evaluation. *ScienceAsia*, 41(4):263–263, 2015.
- [86] Sirinee Thongpanja, Angkoon Phinyomark, Pornchai Phukpattaranont, and Chusak Limsakul. Mean and median frequency of emg signal to determine muscle force based on time-dependent power spectrum. *Elektronika ir Elektrotechnika*, 19(3):51–56, 2013.
- [87] Sirinee Thongpanja, Angkoon Phinyomark, Franck Quaine, Yann Laurillau, Chusak Limsakul, and Pornchai Phukpattaranont. Probability density functions of stationary surface emg signals in noisy environments. *IEEE Transactions on Instrumentation and Measurement*, 65(7):1547–1557, 2016.
- [88] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017.
- [89] Egon L Van Den Broek, Marleen H Schut, Joyce HDM Westerink, Jan van Herk, and Kees Tuinenbreijer. Computing emotion awareness through facial electromyography. In *European Conference on Computer Vision*, pages 52–63. Springer, 2006.

- [90] Hendrik Jacob van Veen and Nathaniel Saul. Keplermapper. <http://doi.org/10.5281/zenodo.1054444>, Jan 2019.
- [91] Joe H. Jr. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.
- [92] Frank Wilcoxon. Individual comparisons by ranking methods. In *Breakthroughs in statistics*, pages 196–202. Springer, 1992.
- [93] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.
- [94] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.
- [95] Songfan Yang and Deva Ramanan. Multi-scale recognition with dag-cnns. In *Proceedings of the IEEE international conference on computer vision*, pages 1215–1223, 2015.
- [96] Yongxin Yang and Timothy M Hospedales. A unified perspective on multi-domain and multi-task learning. *arXiv preprint arXiv:1412.7489*, 2014.
- [97] Mahyar Zardoshti-Kermani, Bruce C Wheeler, Kambiz Badie, and Reza M Hashemi. Emg feature evaluation for movement control of upper extremity prostheses. *IEEE Transactions on Rehabilitation Engineering*, 3(4):324–333, 1995.
- [98] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.
- [99] Xu Zhang, Xiang Chen, Wen-hui Wang, Ji-hai Yang, Vuokko Lantz, and Kong-qiao Wang. Hand gesture recognition and virtual game control based on 3d accelerometer and emg sensors. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pages 401–406. ACM, 2009.
- [100] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.
- [101] Muhammad Zia ur Rehman, Asim Waris, Syed Gilani, Mads Jochumsen, Imran Niazi, Mohsin Jamil, Dario Farina, and Ernest Kamavuako. Multiday emg-based classification of hand motions with deep learning techniques. *Sensors*, 18(8):2497, 2018.

Chapter 6

Virtual Reality to Study the Gap Between Offline and Real-Time EMG-based Gesture Recognition

6.1 Reference

Ulysse Côté-Allard, Gabriel Gagnon-Turcotte, Angkoon Phinyomark, Kyrre Glette, Erik Scheme[‡], François Laviolette[‡] and Benoit Gosselin[‡]. "Virtual Reality to Study the Gap Between Offline and Real-Time EMG-based Gesture Recognition" arXiv preprint (2019).

[‡]These authors share senior authorship

6.2 Context

This article prepare the ground to test unsupervised domain adaptation algorithms for self-calibration. Using the 3DC armband which was presented in the previous article, this paper proposes a multi-day experiment within a virtual reality interface. The VR interface allow to easily integrate the four main dynamic factors of sEMG gesture recognition and provide visual feedbacks to the participant on three of these dynamic factors. This article thus proposes to first look at the impact of these dynamic factors tries to quantify the difficulty of the task. The next article will then try to adapt the classifier without using labeled data.

6.3 Résumé

Dans la littérature sur la reconnaissance de gestes basée sur l'EMG de surface, il existe un gouffre entre la précision d'un classificateur obtenue hors-ligne et ses performances dans une situation en temps réel. Cet écart provient principalement des quatre principaux facteurs

dynamiques de la reconnaissance gestuelle basée sur l’EMG de surface: l’intensité des gestes, la position du bras, le déplacement d’électrode et les changements transitoires du signal. Ces facteurs sont malheureusement difficiles à inclure dans un ensemble de données hors ligne, car chacun d’eux augmente de façon exponentielle le nombre de segments à enregistrer. De plus, les expériences en ligne ont tendance à être effectuées à l’aide de robot coûteux comme un bras robotisé, ou sur un écran d’ordinateur 2D, ce qui limite le nombre de degrés de liberté qui peuvent être contrôlés intuitivement. Par conséquent, cet article propose un environnement en réalité virtuelle (VR) et un protocole expérimental à partir duquel les quatre principaux facteurs dynamiques peuvent être facilement étudiés. 20 personnes ont participé à cette étude en prenant part à trois ou quatre séances sur une période comprise entre 14 et 21 jours. L’environnement VR associé au Leap Motion permet de suivre la position de l’avant-bras du participant de manière continue, ce qui donne une nouvelle façon de voir l’impact de la position du bras sur la classification des gestes basée sur l’EMG de surface.

6.4 Abstract

Within sEMG-based gesture recognition, a chasm exists in the literature between offline accuracy and real-time usability of a classifier. This gap mainly stems from the four main dynamic factors in sEMG-based gesture recognition: gesture intensity, limb position, electrode shift and transient changes in the signal. These factors are hard to include within an offline dataset as each of them exponentially augment the number of segments to be recorded. On the other hand, online datasets are biased towards the sEMG-based algorithms providing feedback to the participants, limiting the usability of such datasets as benchmarks. This paper proposes a virtual reality (VR) environment and a real-time experimental protocol from which the four main dynamic factors can more easily be studied. During the online experiment, the gesture recognition feedback is provided through the leap motion camera, enabling the proposed dataset to be re-used to compare future sEMG-based algorithms. 20 able-bodied persons took part in this study, completing three to four sessions over a period spanning between 14 and 21 days. Finally, TADANN, a new transfer learning-based algorithm, is proposed for long term gesture classification and significantly ($p < 0.05$) outperforms fine-tuning a network.

6.5 Introduction

Muscle activity as a control interface has been extensively applied to a wide range of domains from assistive robotics [17] to serious gaming for rehabilitation [38] and artistic performances [40]. This activity can be recorded non-invasively through surface electromyography (sEMG), a widely adopted technique both in research and clinical settings [17, 25]. Intuitive interfaces can then be created by applying machine learning on the sEMG signal to perform gesture recognition [28].

Despite decades of research in the field [26] however, an important gap still exists between offline classifiers' performance and real-time applications [28]. This disconnect mainly stems from the four main dynamic factors of sEMG signals [35]: Gesture intensity, limb position, electrode shift and the transient nature of EMG signal. Myoelectric signals are also time-consuming to obtain and must be recorded for each user, as extensive variability exists between subjects [7]. This last factor means that, in practice, sEMG datasets used as benchmarks for offline classification rarely contain even a single of these dynamic factors. On the other hand, online myoelectric control naturally provides feedback to the participant. In turn, this feedback biases the recorded online dataset towards the algorithm used for control, as the participants will learn to adapt its behavior to improve the system's usability [33, 32, 10]. Consequently, obtaining a fair comparison of EMG-based gesture recognition algorithms is problematic. Thus recording a new online dataset is often needed to test a new algorithm fairly. Recording such a dataset however, is not only time-consuming, but can also require expensive hardware (e.g. prosthetic arm, robotic arms) [3]. A common alternative of using these costly equipment is through computer simulation (e.g. Fitts' law test [4]) running on a 2D computer screen. These type of simulation however, limits the number of degrees of freedom that can be intuitively controlled. In contrast, virtual reality (VR) offers an attractive and affordable environment for sEMG-based real-time 3D control simulations [31, 2, 6].

As such, this work's main contribution is to have created a virtual reality environment from which an online dataset, featuring 20 participants, recorded specifically to contain the four main dynamic factors is made publicly available. An important innovation of this dataset is that the real-time, gesture recognition feedback is provided solely by a leap motion camera [18]. In other words, the proposed online dataset is not biased towards a particular sEMG-based gesture recognition algorithm and can thus be re-used as a benchmark to compare new algorithms. The VR environment in conjunction with the leap motion tracks the participant's limb orientation in 3D, allowing for more precise understanding of the effect of limb position. The recording sessions, which were "gamified" to better engage the participants, features between three to four recording sessions (equally distant) per participant, spanning a period of 14 to 21 days.

This work proposes an analysis of the effect of the four main dynamic factors on a deep learning classifier. The feature learning paradigm offered by deep learning allows the classifier to directly receive the raw sEMG data as input and achieve classification results comparable with the state of the art [10, 45], something considered "impractical" before [25]. This type of input can be viewed as a sequence of one dimensional images. While ConvNets have been developed to encode spatial information, recurrent neural network-based architectures (RNN) have been particularly successful in classifying sequences of information [22]. Hybrid architectures combining these two types of network are particularly well suited when working with sequences of spatial information [13, 16]. In particular, such hybrids have successfully applied to sEMG-based gesture recognition [19]. Compare to the hybrid ConvNet-RNN,

Temporal Convolutional Networks (TCN) [23, 24] are a purely convolutional approach to the problem of sequence classification which are parallelizable, less complex to train and have low memory requirement. within the context of real-time sEMG-based gesture recognition, especially if applied to prosthetic control, these computational advantages are particularly important. Additionally, TCNs have been shown to outperform RNN-based architectures in a variety of domains and benchmarks using sequential data [5]. Consequently, this work proposes leveraging a TCN-based architecture to perform gesture recognition.

Another contribution of this work is a new transfer learning algorithm for long-term recalibration, named TADANN, combining the transfer learning algorithm presented in [9, 10] and the multi-domain learning algorithm presented in [8].

This paper is divided as follows. The VR experimental protocol and environment is first presented in Section 6.6. Section 6.7 then presents the deep learning classifiers and transfer learning method used in this work. Finally, the results and the associated discussion are covered in Section 6.9 and 7.10 respectively. A flowchart of the material, methods and experiments presented in this work is shown in Figure 6.1.

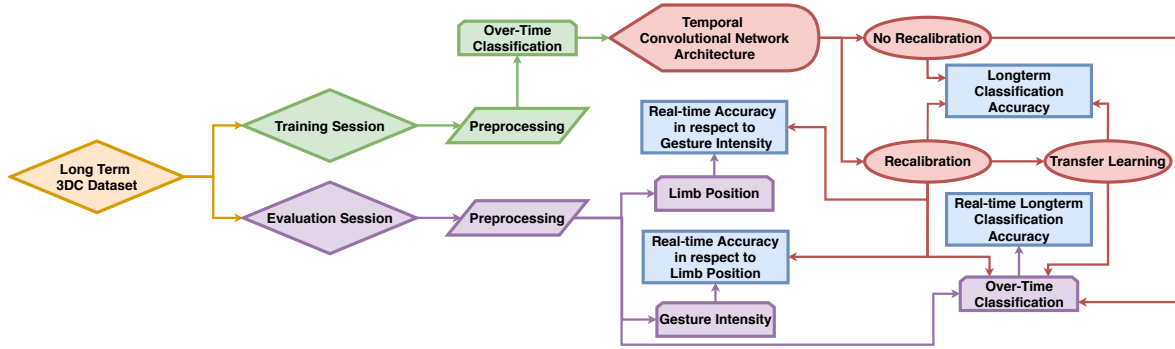


Figure 6.1 – Diagram of the workflow of this work. The two type of recording session from the Long Term 3DC Dataset are first preprocessed. Then, a Temporal Convolutional Network is used with different training schemes. The data from both the evaluation and training session are used in various comparisons/experiments based on the different learning scheme. In the diagram, the blue rectangles represent experiments, while the arrows show which methods/algorithms are required to perform them.

6.6 Long-term sEMG Dataset

This work provides a new, publicly available (<https://github.com/UlysseCoteAllard/LongTermEMG>), multimodal dataset to study the four main dynamic factors in sEMG-based hand gesture recognition. The dataset, referred as the *Long-term 3DC dataset*, features 20 able-bodied participant (5F/15M) aged between 18 and 34 years old (average 26 ± 4 years old) performing the eleven hand/wrist gestures depicted in Figure 7.1. For each participant, the experiment was

recorded in virtual reality over three sessions spanning 14 days (see Section 6.6.3 for details). In addition to this minimum requirement, six of them completed a fourth session, so that the experiment spanned 21 days. Note that originally, 22 persons took part in this study, however, two of them (both male) had to drop out before completing three sessions, due to external circumstances. Consequently, the incomplete data of these two individuals are not included in the results and analysis of this work.



Figure 6.2 – The eleven hand/wrist gestures recorded in the *Long-term 3DC dataset* (image re-used from [11])

The data acquisition protocol was approved by the Comités d'Éthique de la Recherche avec des êtres humains de l'Université Laval (approbation number: 2017-026 A2-R2/26-06-2019). Informed consent was obtained from all participants.

6.6.1 sEMG Recording Hardware

The electromyographic activity of each participant's forearm was recorded with the 3DC Armband [11]; a wireless, 10-channel, dry-electrode, 3D printed sEMG armband. The device, which is shown in Figure 6.3, samples data at 1000 Hz per channel, allowing to take advantage of the full spectra of sEMG signals [29]. In addition to the sEMG acquisition interface, the armband also features a 9-axis Magnetic, Angular Rate, and Gravity (MARG) sensor cadenced at 50 Hz. The dataset features the data of both the sEMG and MARG sensors at 1000 and 50 Hz respectively for each session of every participant.

6.6.2 Stereo-Camera Recording Hardware

During the experiment, in addition to the 3DC Armband, the Leap Motion camera [18] mounted on a VR headset was also used for data recording. The Leap Motion (<https://www.leapmotion.com/>) is a consumer-grade sensor using infrared emitters and two infrared cameras [42] to track a subject's forearm, wrist, hand and fingers in 3D. In addition to the software-generated

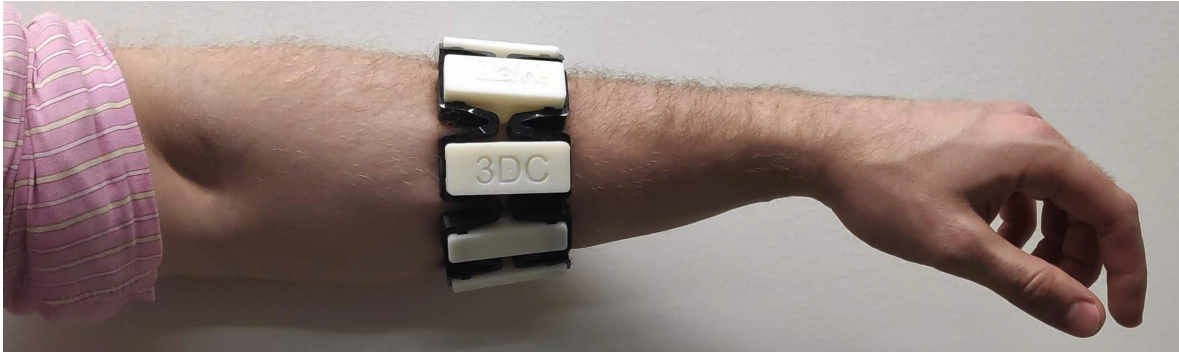


Figure 6.3 – The 3DC Armband used in this work records electromyographic and orientation (9-axis Magnetic, Angular rate and gravity sensor) data. The wireless, dry-electrode armband features 10 channels, each cadenced at 1 kHz.

representation of the hand, the *Long-term 3DC dataset* also contains the raw output of the stereo-camera recorded at ~ 10 Hz.

6.6.3 Experimental Protocol in Virtual Reality

Each recording session is divided in two parts: the *Training Session* and the *Evaluation Session*, both of which are conducted in VR. Figure 6.4 helps visualizes the general interface of the software while [this video \(https://youtu.be/BnDwcw8ol6U\)](https://youtu.be/BnDwcw8ol6U) shows the experiment in action. Note that, for every training session, two evaluation sessions were also performed. All three sessions were recorded within a timespan of an hour.

Before any recording started, the 3DC Armband was placed on the dominant arm of the participant. The armband was slid up until its circumference matched that of the participant's forearm. A picture was then taken to serve as reference for the armband placement. In subsequent sessions, the participant placed the armband on their forearm themselves, aided only with the reference picture. Hence, electrode displacement between sessions is expected.

Training Session

The training session's main purpose was to generate labeled data, while familiarizing the participants with the VR setup. To do so, the participants were asked to put on and adjust the VR headset to maximize comfort and minimize blurriness. The VR platform employed in this work is the Vive headset (<https://www.vive.com/us/>). After a period of adjustment of a few minutes the recording started. All in all, the delay between a participant putting the armband on their forearm and the start of the recording was approximately five minutes on average.

The VR environment showed the participant the gesture to perform using an animation of a 3D arm performing the gesture. All gesture recordings were made with the participants standing up with their forearm parallel to the floor unsupported. Starting from the neutral gesture,

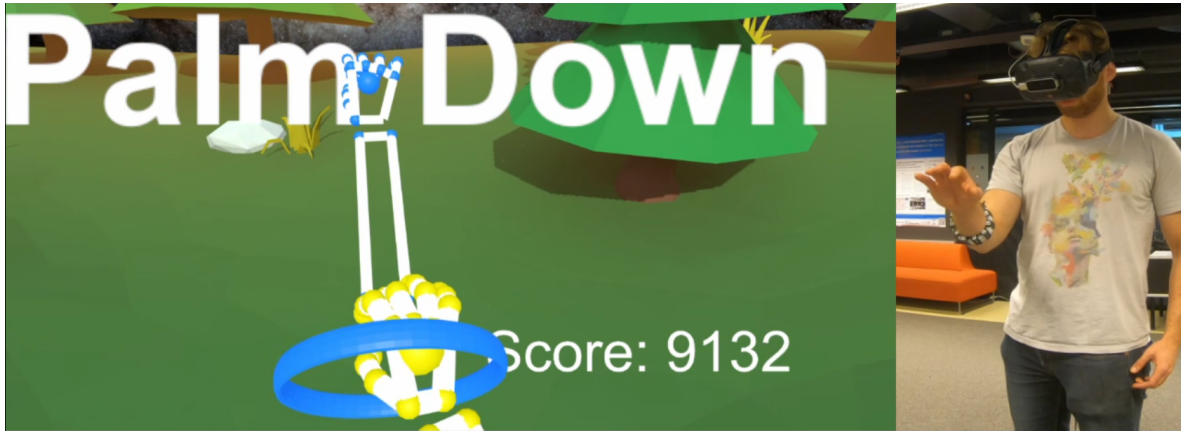


Figure 6.4 – The VR environment during the evaluation session. The scenery (trees, horizon) helps orient the participants. The requested gesture is written on the participant's head-up display and shown as an animation (the blue hand model). The ring indicates the desired hand's position while its color (and the color of the blue hand) indicates the requested gesture's intensity. The yellow hand represents the participant's virtual prosthetic hand and changes color based on the intensity at which the participant is performing the gesture. The score augments if the participant is performing the correct gesture. Bonus points are given if the participant is performing the gesture at the right position and intensity. Note that the software's screenshot only shows the right eye's view and thus does not reflect the depth information seen by the participant.

they were instructed, with an auditory cue, to hold the depicted gesture for five seconds. The cue given to the participants were in the following form: "Gesture X, 3, 2, 1, Go". The data recording began just before the movement was started by the participant as to capture the ramp-up segment of the muscle activity and always started with the neutral gesture. The recording of the eleven gestures for five seconds each was referred to as a *cycle*. A total of four cycles (220s of data) were recorded with no interruption between cycles (unless requested by the participant). When recording the second cycle, the participants were asked to perform each gesture (except the neutral gesture) with maximum intensity. This second cycle serves as a baseline for the maximum intensity of each gesture on a given day, on top of providing labeled data. For the other three cycles, a "normal" level of intensity was requested from the participants ($43.43\% \pm 23.02\%$ of their perceived maximum intensity on average).

Evaluation Session

The evaluation session main purpose was to generate data containing the four main dynamic factors within an online setting. The sessions took the form of a "game", where the participants were randomly requested to hold a gesture at a given intensity and position in 3D. Figure 6.4 provides an overview of the evaluation session.

The evaluation session always took place after a training session within the VR environment, without removing the armband between the two session. The participants were first asked to

stand with their arm stretched forward to calibrate the user’s maximum reach. Then, the user was requested to bend their elbow 90 degrees, with their forearm parallel to the floor (this was the starting position). Once the participant is ready, the researcher starts the experiment which displays a countdown to the participant in the game. When the game starts, a random gesture is requested through text on the participant’s head-up display. Additionally, a floating ring appears at a random position within reach of the participant, with a maximum angle of ± 45 and ± 70 degrees in pitch and yaw respectively. The floating ring’s color (blue, yellow and red) tells the participant at what level of intensity to perform the requested gesture. Three levels of intensity were used: (1) less than 25%, (2) between 25 to 45% and (3) above 45% of the participant’s maximal intensity as determined from the participant’s first training session. A new gesture, position and intensity are randomly requested every five seconds with a total of 42 gestures asked during an evaluation session (210 seconds).

During the experiment and using the leap motion, a *virtual prosthetic arm* is mapped to the participant’s arm, which matches its position and pitch/yaw angles. The participant is thus able to intuitively know where their arm is in the VR environment and how to best reach the floating ring. However, the virtual prosthetic does not match the participant’s hand/wrist movements nor its forearm’s roll angle. Instead, the leap motion’s data is leveraged to predict the subject’s current gesture using a convNet (see Section 6.7.1 for details). The hand of the virtual prosthetic then moves to perform the predicted gesture (including supination/pronation with the roll angle) based on the data recorded during the training session, providing direct feedback to the participant. Note that the sEMG data has no influence on the gesture’s prediction as to not bias the dataset toward a particular EMG classification algorithm. The virtual prosthetic also changes color (blue, yellow, red) based on the currently detected gesture intensity from the armband. Finally, a score is shown to the participant in real-time during the experiment. The score augments when the detected gesture matches the requested gesture. Bonus points are given when the participant correctly matches the requested gesture’s intensity and is performing the gesture at the right position.

6.6.4 Data Pre-processing

This work aims at studying the effect of the four main dynamic factors in myoelectric control systems. Consequently, the input latency is a critical factor to consider. As the optimal guidance latency was found to be between 150 and 250 ms [39], within this work, the data from each participant is segmented into 150 ms frames with an overlap of 100 ms. The raw data is then band-pass filtered between 20-495 Hz using a fourth-order butterworth filter.

6.6.5 Experiments with the Long-term 3DC Dataset

The training sessions will be employed to compare the algorithms described in this work in an offline setting. When using the training sessions for comparison, the classifiers will be trained

on the first and third cycle and tested on the fourth cycle. The second cycle, comprised of the maximal intensity gestures recording, is omitted as to only take into account electrode shift/non-stationarity of the signal and to allow an easier comparison with the literature.

The evaluation session is employed to study the impact of the four main dynamic factors on EMG-based gesture recognition. Classifiers will be trained on cycle 1, 3 and 4 of the training sessions and tested on the two evaluation sessions.

6.6.6 3DC Dataset

A second dataset, referred to as the *3DC Dataset* and featuring 22 able-bodied participants, is used for architecture building, hyperparameters selection and pre-training. This dataset, presented in [11], features the same eleven gestures and is also recorded with the 3DC Armband. Its recording protocol closely matches the training session description (Section 6.6.3), with the difference being that two such sessions were recorded for each participant (one single day recording). This dataset was preprocessed as described in Section 7.6.1. Note that when recording the 3DC Dataset, participants were wearing both the Myo and 3DC Armband, however in this work, only the data from the 3DC Armband is employed.

6.7 Deep Learning Classifiers

The following section presents the deep learning architectures employed in this work for the classification of both EMG data and images from the leap motion camera. The PyTorch [27] implementation of the networks are [readily available here \(https://github.com/UlysseCoteAllard/LongTermEMG\)](https://github.com/UlysseCoteAllard/LongTermEMG).

6.7.1 Leap Motion Convolutional Network

For real-time myoelectric control, visual feedback helps the participant to produce more consistent and discriminative signals [30, 10]. Such feedback is also natural to have as the participant should, in most case, be able to see the effect of its control. To avoid biasing the proposed dataset toward a particular EMG-based classification algorithm, the gesture-feedback was provided using solely the leap motion.

Image classification is arguably the domain in which ConvNet-based architecture had the greatest impact due, in part, to the vast amount of labeled data available [22]. However, within this work and as to provide consistent feedback, training data was limited to the first training session of each participant. Consequently, the network had to be trained with a low amount of data (around 200 examples per gestures). Additionally, while the training session was recorded with a constant point of view of the participant’s hand, the evaluation session generated, by design, widely different point of view that the network had to contend with during inference.

The variable point-of-view problem was addressed using the capability of the leap motion camera to generate a 3D model in the virtual environment of the participant’s hand. Three virtual depth-cameras were then placed around the arm’s 3D representation from three different and fixed point-of-view to capture images of the 3D model (see Figure 6.5 (A) for an example). The three images were then merged together by having each image encoded within one channel of a three-channel color image (see Figure 6.5 (B) for examples). Finally, pixel intensity was inverted (so that a high value corresponds to a part of the hand being close to the camera) before being fed to the ConvNet. Note that one of the main reasons to uses images as input instead of 3D point clouds is to reduce the computational requirement during both training and inference.

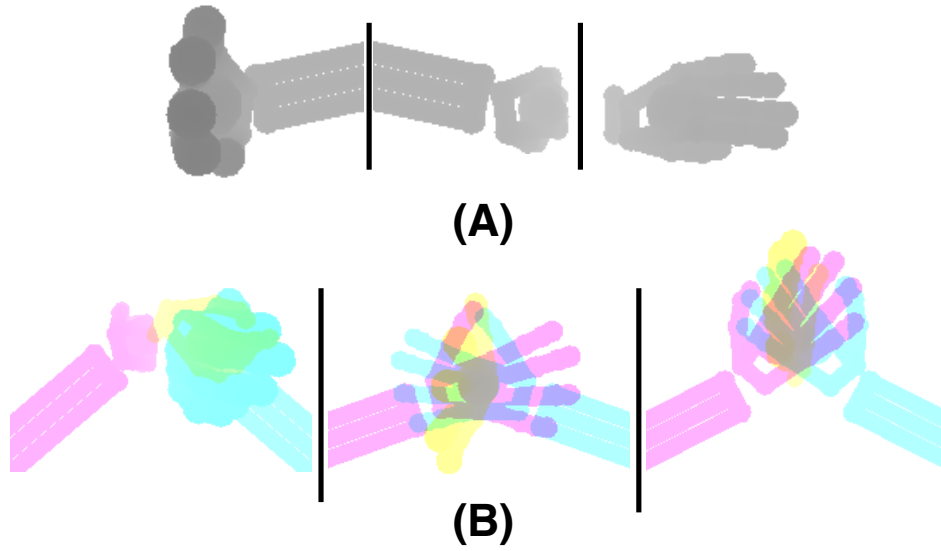


Figure 6.5 – (A) The depth images (darker pixel are closer) of the three virtual cameras taken at the same moment. The gesture captured is Wrist Flexion. Note that, regardless of the participant’s movement, the three cameras are always placed so that they have the same point-of-view in relation to the forearm. (B) Examples of images fed to the ConvNet. The represented gestures from left to right: Wrist Flexion, Open Hand, Radial Deviation.

To address the data sparsity problem, the transfer learning algorithm described in [9, 10] was employed using the data from the 3DC Dataset for pre-training.

The leap motion ConvNet’s architecture is based on EfficientNet-B0 [41] and presented in Table 6.1.

6.7.2 EMG-based Temporal Convolutional Network

TCNs generally differ in two aspects from standard ConvNets. First, TCNs leverage stacked layers of *dilated convolutions* to achieve large receptive fields with a few layers. Dilated convolutions (also known as convolution à trous or convolution with holes) is a convolutional

Table 6.1 – Leap motion ConvNet’s architecture

Level n	Layer Type	Input Dimension Height x Width	#Channels	# Layers Source Network	# Layers Target Network
1	Conv3x3	225×225	33	1	1
2	ConvBlock3x3	113×113	16	2	1
3	ConvBlock5x5	57×57	24	2	1
4	ConvBlock3x3	29×29	32	2	1
5	ConvBlock5x5	15×15	48	2	1
6	ConvBlock5x5	8×8	64	2	1
7	Conv1x1 & Pooling & FC	4×4	64	1	1

Each row describes a level n of the ConvNet. The pooling layer is a global average pooling layer (giving one value per channel), while "FC" refers to a fully connected layer.

layer where the kernel is applied over a longer range by skipping input values by a constant amount [23]. Typically, the dilatation coefficient (d) is defined as $d = 2^i$ where i is the i th layer from the input (starting with $i=0$). The second difference is that TCNs are built with dilated *causal* convolutions where the *causal* part means that the output at time t is convolved only with elements from outputs from time t or earlier. In practice, such a behavior is achieved (in the 1D case with PyTorch) by padding the left side (assuming time flows from left to right) of the vector to be convolved by $(k - 1) * d$, where k is the kernel’s size. This also ensures a constant output size throughout the layers.

The proposed TCN, receives the sEMG data with shape Channel \times Time (10×150). The architecture is based on [8, 5]. The PyTorch implementation is derived from [5].

The TCN’s architecture (see Figure 6.6), contains three *blocks* followed by a global average pooling layer before the output layer. Each block encapsulate a dilated causal convolutional layer [23] followed by batch normalization [20], leaky ReLU [43] and dropout [14].

Adam [21] is employed for the TCN’s optimization with an initial learning rate of 0.0404709 and batch size of 512. 10% of the training data is held out as a validation set which is used for early stopping (with a ten epochs threshold) and learning rate annealing (factor of five and a patience of five). Note that all architecture choices and hyperparameters selection were performed using the *3DC Dataset* or previous works.

6.7.3 Calibration Training Methods

This work considers three calibration methods for long-term classification of sEMG signals: No Calibration, Re-Calibration and Delayed Calibration. In the first case, the network is trained solely from the data of the first session. In the Re-Calibration case, the model is re-trained at each new session with the new labeled data. To leverage previous data, fine-tuning [44] is

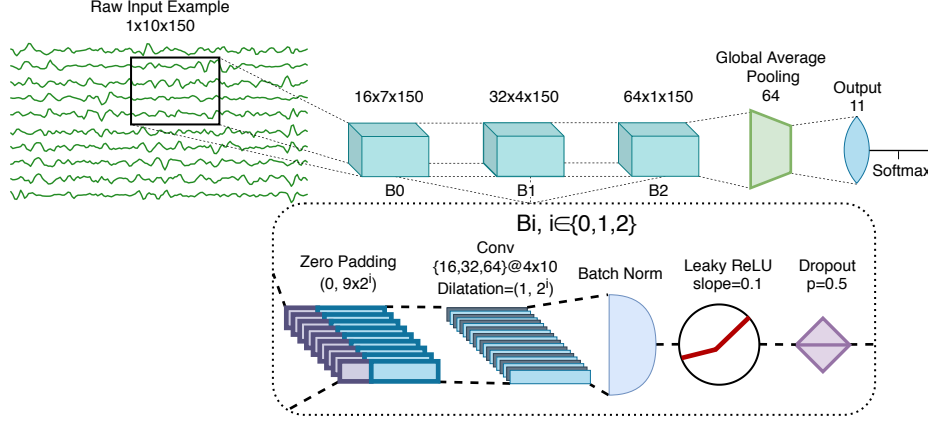


Figure 6.6 – The ConvNet’s architecture employing 104 788 learnable parameters. In this figure, B_i refers to the i th block ($i \in \{0, 1, 2\}$). Conv refers to a convolutional layer while Chomp removes the padding after the convolution.

applied. That is, during re-calibration, the weights of the network are first initialized with the weights found from the previous session. Note that the proposed transfer learning (Section 6.8) will also use the Re-Calibration setting. Delayed Calibration is similar to Re-Calibration, but the network is re-calibrated on the previous session instead of the newest one. The purpose of Delayed Calibration is to see how the classifier’s degradation evolves when there is a similar amount of days since each previous calibration.

6.8 Transfer Learning

Over multiple re-calibration sessions, large amount of labeled data is recorded. However, standard training methods are limited to the data from the most recent session as they cannot take into account the signal drift between each recording. Transfer learning algorithms on the other hand can be developed to account for such signal disparity. Consequently, this work proposes to combine the Adaptive Domain Adversarial Neural Network (ADANN) training presented in [8] and the transfer learning algorithm presented in [10] for inter-session gesture recognition. This new algorithm is referred to as Transferable Adaptive Domain Adversarial Neural Network (TADANN). For simplicity’s sake, the ensemble of calibration sessions prior to the most recent one are referred to as the *pre-calibration sessions* whereas the most recent one is referred to as the *calibration session*.

The proposed algorithm contains a pre-training and a training step. During pre-training, each session within the pre-calibration sessions is considered as a separate labeled domain dataset. At each epoch, pre-training is performed by sharing the weights of a network across all the domains (i.e. pre-calibration sessions), while the Batch-Normalization (BN) statistics are learn independently for each session [8]. The idea behind ADANN is then to extract a general feature representation from this multi-domain setting. To do so, a domain classification head (with

two neurons) is added to the network. At each epoch, a batch is created containing examples from a single, randomly selected, session at a time (referred to as the *source batch*). A second batch (the *target batch*) is then created from a, also randomly, selected session (different than the one used to create the source batch). The examples from the source batch are assigned the domain-label 0, while the domain-label 1 is assigned to the examples from the target batch. Then, a gradient reversal layer [15] is used right after the domain-head during backpropagation to force the network to learn a session-independent feature representation. Note that the BN statistics used by the network correspond to the session from which the source or target batch originate, but that they are updated only with the source batch. Similarly, the classification head is used to back-propagate the loss only with the source batch.

After pre-training is completed, the learned weights are frozen, except for the BN parameters which allow the network to adapt to a new session. Then, a second network is initialized (in this work, the second network is identical to the pre-trained network) and connected with an element-wise summation operation in a layer-by-layer fashion to the pre-trained network (see [10] for details). Additionally, all outputs from the pre-trained network are multiplied by a learnable coefficient (clamped between 0 and 2) before the summation as to provide an easy mechanism to neuter or increase the influence of the pre-trained network at a layer-wise level.

6.9 Results

6.9.1 Training Sessions: Over-time classification accuracy

Figure 6.7 shows the average accuracy over-time across all participants for the three calibration methods and with TADANN (which uses the Re-Calibration method).

Based on Cohen’s d, the effect size of using Re-Calibration vs No-Calibration varies between large to very large [34] (0.95 and 1.32 for session two and three respectively). Overall, TADANN was the best performing method, achieving an average accuracy of $84.44\% \pm 19.15\%$ and $89.04\% \pm 6.49\%$ compared to $79.96\% \pm 18.40\%$ and $80.49\% \pm 21.58\%$ for session two and three respectively. Using the Wilcoxon signed rank test [12] shows that TADANN significantly outperforms Re-Calibration (adjusted p-value = 0.0004 and = 0.002 for session two and three respectively). The effect size was small (0.40) and medium (0.54) using Cohen’s d on session two and three respectively. Note that statistical tests were not performed for session four due to the sample size (n=6).

6.9.2 Evaluation Session

Figure 6.8 shows the scores obtained for all participants on the evaluation sessions in respect to TADANN’s accuracy from the corresponding session. The Pearson r correlation coefficient between the score and accuracy is 0.52. The average scored obtained during the first recording

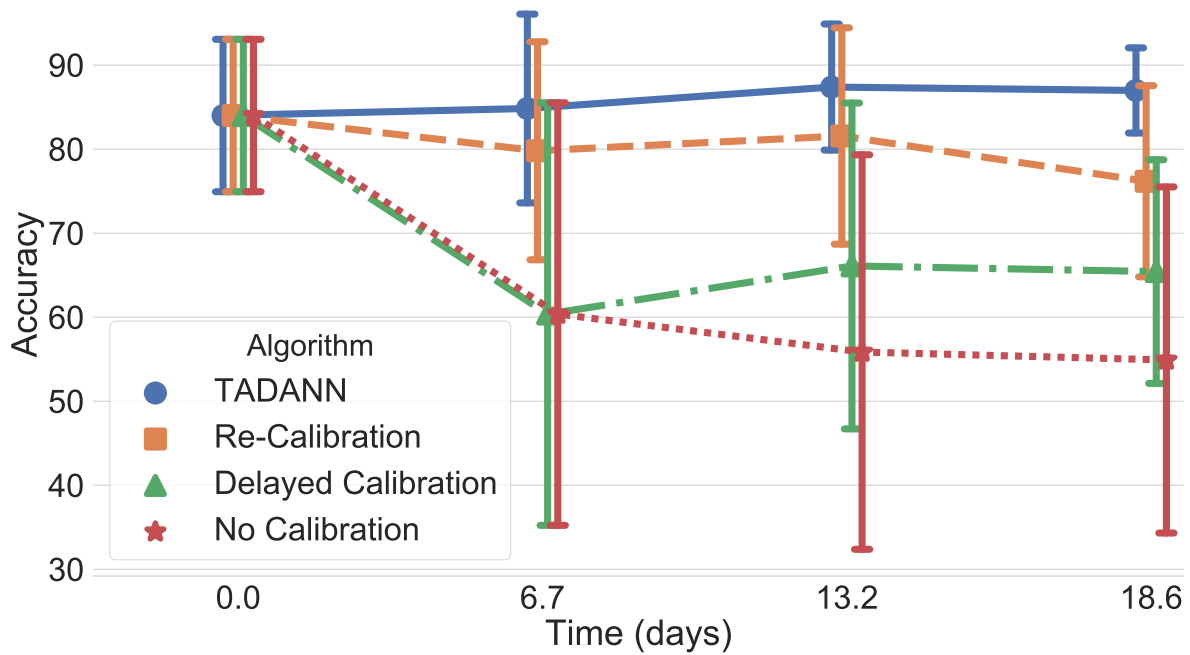


Figure 6.7 – Average accuracy over-time calculated on the last cycle of the training sessions. The values given on the x-axis represent the average time (in days) elapsed between the current session and the first session across all participants.

session was 5634 ± 1521 which increased to 6615 ± 1661 on session three, showing that the participants improved.

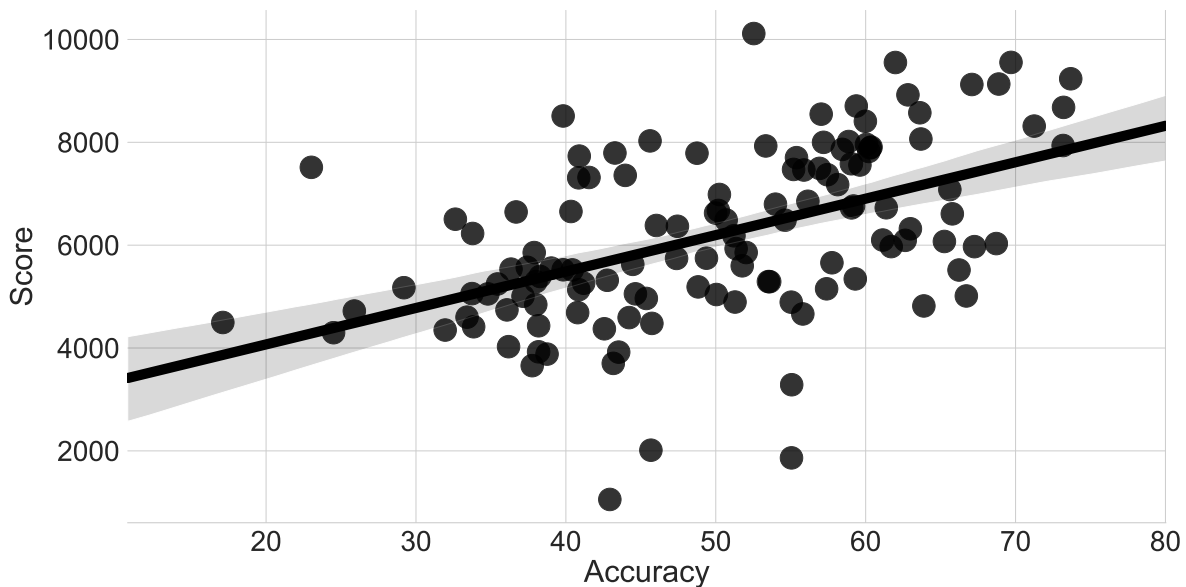


Figure 6.8 – Score obtained by each participant at each evaluation session in respect to TADANN's accuracy on the evaluation sessions. The translucent bar around the regression represents the standard deviation.

Over-time classification accuracy

Figure 6.9 shows the average accuracy over-time on the evaluation sessions across all participants for the three calibration methods and with TADANN (which uses the Re-Calibration method). Re-Calibration again outperforms No-Calibration and the effect was small (0.31 and 0.45) according to cohen's d for session two and three respectively. TADANN again significantly outperformed the Re-Calibration (adjusted p-value = 0.003 and = 0.0005) and the effect size was 0.11 and 0.19 for session two and three respectively.

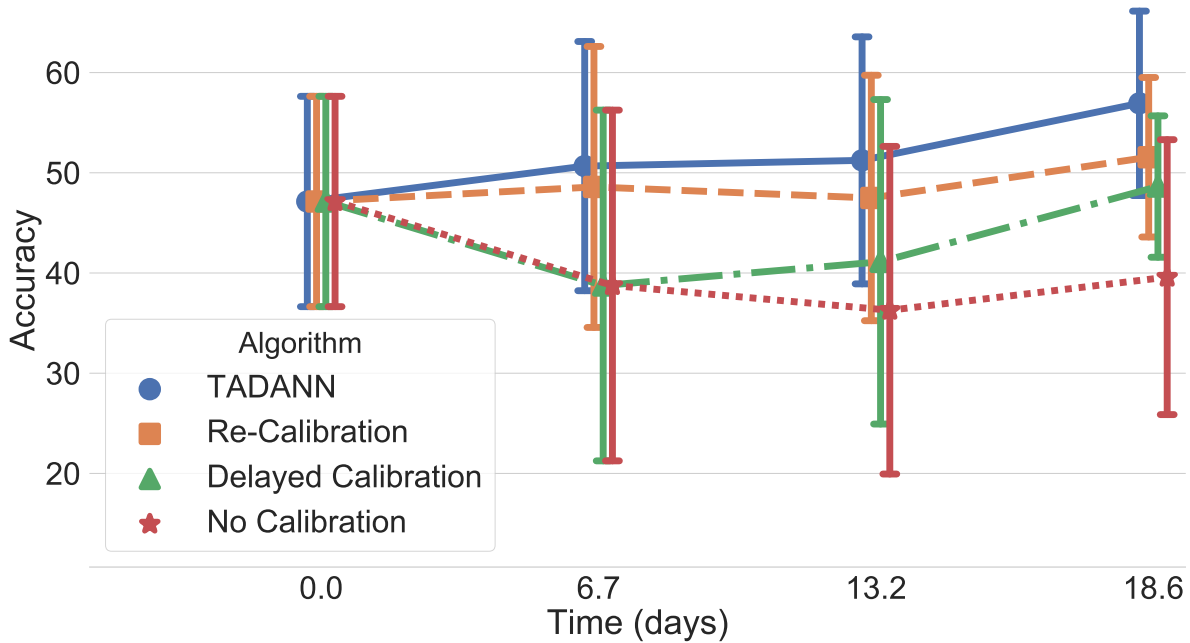


Figure 6.9 – Average accuracy over-time calculated on the evaluation sessions. The values given on the x-axis represent the average time (in days) elapsed between the current session and the first session across all participants.

limb orientation

The impact of limb's position on the Re-Calibrated ConvNet's accuracy is shown in Figure 6.10. Accuracies were computed on the online dataset across all sessions and all participants. The first 1.5s after a new gesture was requested were removed from the data used to generate Figure 6.10, as to reduce the impact of gesture's transition.

Gesture intensity

Figure 6.11 shows the impact of gesture's intensity on the Re-Calibration classifier's accuracy. Accuracies were computed on the online dataset across all sessions and all participants (excluding the neutral gesture). The first 1.5s after a new gesture was requested were again removed from the data used to generate Figure 6.11.

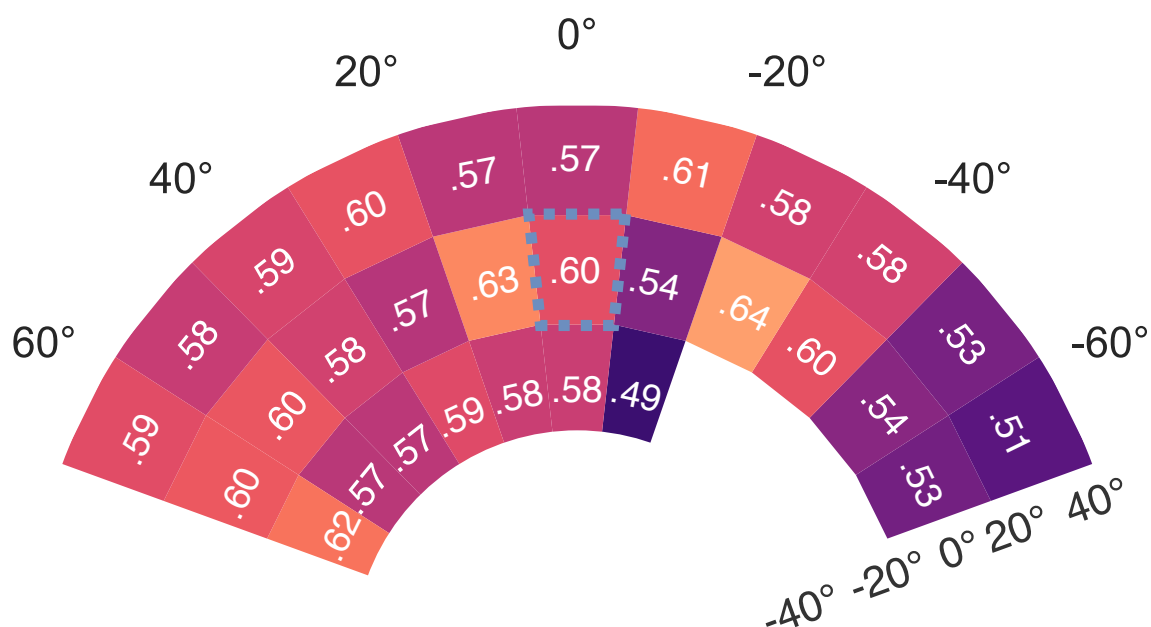


Figure 6.10 – Accuracy in respect to the pitch and yaw angles. The dotted line indicates the neutral orientation. Note that, a minimum threshold of 600 examples per pitch/yaw combination was set to show the accuracy.

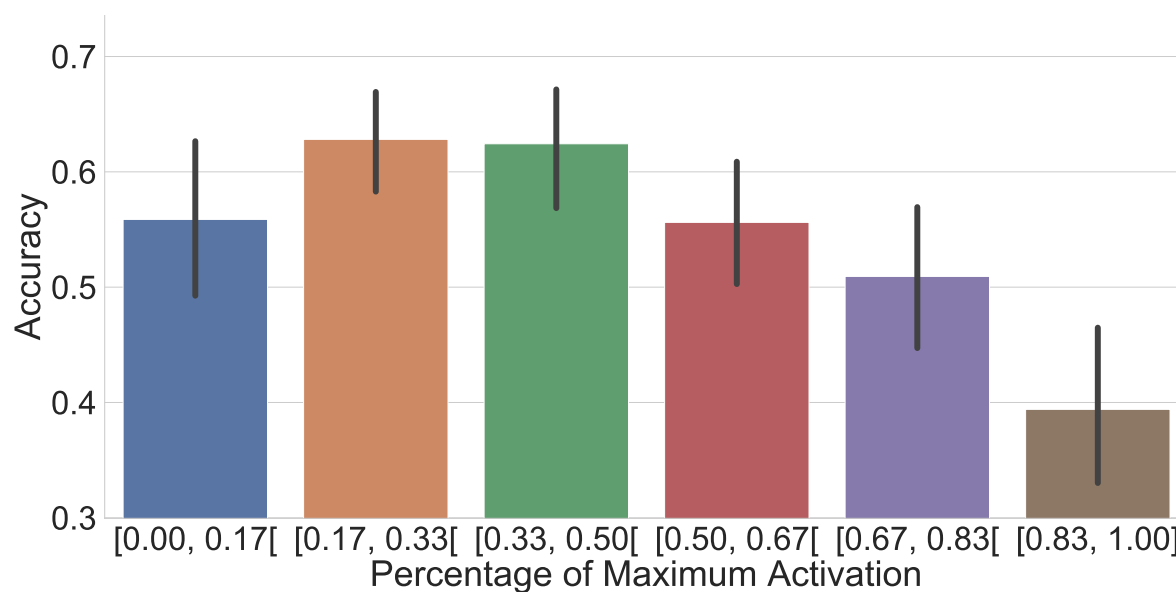


Figure 6.11 – Average accuracy obtained from the re-calibrated ConvNet in respect to the percentage of the maximum activation when performing the gestures over all evaluation sessions across all participants. Note that the data from the gesture transition period were ignored when computing the accuracy (by removing one second of data whenever a new gesture was requested).

6.10 Discussion

This paper leverages the leap motion for gesture recognition, to avoid biasing the real-time dataset toward a particular sEMG-based algorithm. Figure 6.8 shows that the score obtained from a session correlates with the accuracy obtained from the same session. Note that the three lowest scores come from sessions where the leap motion lost tracking of the hand particularly often. Comparing the Delayed Calibration with the No Calibration from Figure 6.7 and 6.9 shows that participant where able to learn to produce more consistent gestures across sessions (from a sEMG-based classifier perspective). Thus, feedback provided by the leap motion seems to act as a good proxy, while also removing the bias normally present in online datasets. Qualitatively, the participants enjoyed the experiment gamification as almost all of them were trying to beat their own high-score and to claim to the top of the leader-board. Additionally, several participants requested to do "one more try" to try to achieve a high-score (only allowed after their last session). As such, virtual reality can provide an entertaining environment from which to perform complex 3D tasks [31, 2, 6] at an affordable cost when compared to using robotic arms or myoelectric prosthesis.

Inter-day classification was shown to have a significant impact both offline and online. With standard classification algorithms, the need for periodic re-calibration is thus apparent. The proposed TADANN algorithm was shown to consistently achieve higher accuracy than simple fine-tuning re-calibration. In this particular dataset, on a per-subject basis, TADANN routinely outperformed fine-tuning by more than 5%, whereas for the opposite 1% or less was the most common. The difference between the two also grew as TADANN could pre-train on more sessions. Thus future work will consider even more sessions per participant to evaluate TADANN.

Figure 6.10 shows that gestures which were performed while the participant's arm was externally rotated were the hardest in general for the classifier to correctly predict. This is likely due to the fact that the origin of the brachioradialis muscle (which is under the area of recording) is the lateral supracondylar ridge of the humerus. It is possible, therefore, that as the humerus becomes more externally rotated that it changes the geometry of the brachioradialis, affecting the observed signals. In addition, the arm may tend to supinate slightly for higher levels of external humeral rotation, which is known to create worse limb position effect than the overall arm position. In contrast, when the participant's arm was internally rotated, no such drastic drop in performance was noted. As shown in [36], training a classifier by including multiple limb-positions can improve inter-position performances. Consequently, it might be beneficial for future studies to focus on including externally rotated forearm positions within the training dataset. Note however, that while the participants were instructed to limit as most as possible any torso rotation, they were not restrained and consequently such rotation are likely present within the dataset. This might explain the decrease→increase→decrease in accuracy observed for the external rotation. Participants accepted an external rotation up to when they felt

uncomfortable and then rotated their torso. This also explains the lower number of examples with an external yaw and a downward pitch as such combinations tend to be uncomfortable (the software considered all angle combination with equal probability).

The impact of gesture's intensity obtained within this study corroborate past findings in the literature [36]. The classifier is relatively unaffected by different levels of gesture intensity between 17 and 50%. Additionally, at lower intensity, the main error factor comes from classifying the neutral gesture. However it has been shown that rejection-based classifiers can improve classifier's usability [37]. The problematic intensities are thus all above 50% of maximal gesture intensity.

The main limitation of this study is the relatively important gap between sessions. While such a scenario is realistic (e.g. for consumer grade armband used to play video games or make a presentation) it does not allow to smoothly see the change in signals within day. As such, future works will expend upon the current dataset to include more frequent evaluation sessions for each participant (and multiple within the same day).

6.11 Conclusion

This paper presented a new VR experimental protocol for sEMG-based gesture recognition leveraging the leap motion camera as to not bias the online dataset. Quantitatively and qualitatively, the participants were shown to improve over time and were motivated in taking part in the experiment. Overall, TADANN was shown to significantly outperform fine-tuning. The VR environment in conjunction with the leap motion allowed to quantify the impact of limb position with, to the best of the authors knowledge, the highest resolution yet.

Future work will use self-calibrating algorithms based on domain adversarial training [1] to hopefully reduce the impact of transient change in sEMG signals.

Acknowledgment

The authors would like to thank Alexandre Campeau-Lecours for his support without which this manuscript would not have been possible.

Funding

This research was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) [funding reference numbers 401220434, 376091307, 114090], the Institut de recherche Robert-Sauvé en sante et en sécurité du travail (IRSST). Cette recherche a été financée par le Conseil de recherches en sciences naturelles et en genie du Canada (CRSNG) [numéros de référence 401220434, 376091307, 114090].

6.12 Bibliography

- [1] Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. Domain-adversarial neural networks. *arXiv preprint arXiv:1412.4446*, 2014.
- [2] Adel Al-Jumaily and Ricardo A Olivares. Electromyogram (emg) driven system based virtual reality for prosthetic and rehabilitation devices. In *Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services*, pages 582–586. ACM, 2009.
- [3] Ulysse Côté Allard, François Nougrou, Cheikh Latyr Fall, Philippe Giguère, Clément Gosselin, François Laviolette, and Benoit Gosselin. A convolutional neural network for robotic arm guidance using semg based frequency-features. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2464–2470. IEEE, 2016.
- [4] Ali Ameri, Mohammad Ali Akhaee, Erik Scheme, and Kevin Englehart. Real-time, simultaneous myoelectric control using a convolutional neural network. *PloS one*, 13(9):e0203835, 2018.
- [5] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.
- [6] Dimitra Blana, Theocharis Kyriacou, Joris M Lambrecht, and Edward K Chadwick. Feasibility of using combined emg and kinematic signals for prosthesis control: A simulation study using a virtual reality environment. *Journal of Electromyography and Kinesiology*, 29:21–27, 2016.
- [7] Claudio Castellini, Angelo Emanuele Fiorilla, and Giulio Sandini. Multi-subject/daily-life activity emg-based control of mechanical hands. *Journal of neuroengineering and rehabilitation*, 6(1):41, 2009.
- [8] Ulysse Côté-Allard, Evan Campbell, Angkoon Phinyomark, François Laviolette, Benoit Gosselin, and Erik Scheme. Interpreting deep learning features for myoelectric control: A comparison with handcrafted features. *arXiv preprint arXiv:1912.00283*, 2019.
- [9] Ulysse Cote-Allard, Cheikh Latyr Fall, Alexandre Campeau-Lecours, Clément Gosselin, François Laviolette, and Benoit Gosselin. Transfer learning for semg hand gestures recognition using convolutional neural networks. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1663–1668. IEEE, 2017.
- [10] Ulysse Côté-Allard, Cheikh Latyr Fall, Alexandre Drouin, Alexandre Campeau-Lecours, Clément Gosselin, Kyrre Glette, François Laviolette, and Benoit Gosselin. Deep learning

- for electromyographic hand gesture signal classification using transfer learning. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(4):760–771, 2019.
- [11] Ulysse Côté-Allard, Gabriel Gagnon-Turcotte, François Laviolette, and Benoit Gosselin. A low-cost, wireless, 3-d-printed custom armband for semg hand gesture recognition. *Sensors*, 19(12):2811, 2019.
 - [12] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006.
 - [13] Yin Fan, Xiangju Lu, Dian Li, and Yuanliu Liu. Video-based emotion recognition using cnn-rnn and c3d hybrid networks. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, pages 445–450. ACM, 2016.
 - [14] Yarín Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.
 - [15] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.
 - [16] Yanming Guo, Yu Liu, Erwin M Bakker, Yuanhao Guo, and Michael S Lew. Cnn-rnn: a large-scale hierarchical image classification framework. *Multimedia Tools and Applications*, 77(8):10251–10271, 2018.
 - [17] Maria Hakonen, Harri Piitulainen, and Arto Visala. Current state of digital signal processing in myoelectric interfaces and related applications. *Biomedical Signal Processing and Control*, 18:334–359, 2015.
 - [18] David Holz, Kyle Hay, and Michael Buckwald. Electronic sensor, April 14 2015. US Patent App. 29/428,763.
 - [19] Yu Hu, Yongkang Wong, Wentao Wei, Yu Du, Mohan Kankanhalli, and Weidong Geng. A novel attention-based hybrid cnn-rnn architecture for semg-based gesture recognition. *PloS one*, 13(10):e0206049, 2018.
 - [20] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
 - [21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
 - [22] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

- [23] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.
- [24] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.
- [25] Mohammadreza Asghari Oskoei and Huosheng Hu. Myoelectric control systems—a survey. *Biomedical signal processing and control*, 2(4):275–294, 2007.
- [26] P Parker, K Englehart, and Bernard Hudgins. Myoelectric signal processing for control of powered limb prostheses. *Journal of electromyography and kinesiology*, 16(6):541–548, 2006.
- [27] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.
- [28] Angkoon Phinyomark, Franck Quaine, Sylvie Charbonnier, Christine Serviere, Franck Tarpin-Bernard, and Yann Laurillau. Emg feature evaluation for improving myoelectric pattern recognition robustness. *Expert Systems with applications*, 40(12):4832–4840, 2013.
- [29] Angkoon Phinyomark and Erik Scheme. A feature extraction issue for myoelectric control based on wearable emg sensors. In *2018 IEEE Sensors Applications Symposium (SAS)*, pages 1–6. IEEE, 2018.
- [30] Tobias Pistohl, Deepak Joshi, Gowrishankar Ganesh, Andrew Jackson, and Kianoush Nazarpour. Artificial proprioceptive feedback for myoelectric control. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 23(3):498–507, 2014.
- [31] Jose L Pons, R Ceres, E Rocon, S Levin, I Markovitz, B Saro, Dominiek Reynaerts, W Van Moorleghe, and Leandro Bueno. Virtual reality training and emg control of the manus hand prosthesis. *Robotica*, 23(3):311–317, 2005.
- [32] Michael A Powell, Rahul R Kaliki, and Nitish V Thakor. User training for pattern recognition-based myoelectric prostheses: Improving phantom limb movement consistency and distinguishability. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(3):522–532, 2013.
- [33] Saritha M Radhakrishnan, Stuart N Baker, and Andrew Jackson. Learning a novel myoelectric-controlled interface task. *Journal of neurophysiology*, 100(4):2397–2408, 2008.
- [34] Shlomo S Sawilowsky. New effect size rules of thumb. *Journal of Modern Applied Statistical Methods*, 8(2):26, 2009.

- [35] Erik Scheme and Kevin Englehart. Electromyogram pattern recognition for control of powered upper-limb prostheses: state of the art and challenges for clinical use. *Journal of Rehabilitation Research & Development*, 48(6), 2011.
- [36] Erik Scheme, A Fougner, Øyvind Stavadahl, Adrian DC Chan, and Kevin Englehart. Examining the adverse effects of limb position on pattern recognition based myoelectric control. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pages 6337–6340. IEEE, 2010.
- [37] Erik J Scheme, Bernard S Hudgins, and Kevin B Englehart. Confidence-based rejection for improved pattern recognition myoelectric control. *IEEE Transactions on Biomedical Engineering*, 60(6):1563–1570, 2013.
- [38] Xing Shusong and Zhang Xia. Emg-driven computer game for post-stroke rehabilitation. In *2010 IEEE Conference on Robotics, Automation and Mechatronics*, pages 32–36. IEEE, 2010.
- [39] Lauren H Smith, Levi J Hargrove, Blair A Lock, and Todd A Kuiken. Determining the optimal window length for pattern recognition-based myoelectric control: balancing the competing effects of classification error and controller delay. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 19(2):186–192, 2010.
- [40] David St-Onge, Ulysse Côté-Allard, Kyrre Glette, Benoit Gosselin, and Giovanni Beltrame. Engaging with robotic swarms: Commands from expressive motion. *ACM Transactions on Human-Robot Interaction (THRI)*, 8(2):11, 2019.
- [41] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019.
- [42] Frank Weichert, Daniel Bachmann, Bartholomäus Rudak, and Denis Fisseler. Analysis of the accuracy and robustness of the leap motion controller. *Sensors*, 13(5):6380–6393, 2013.
- [43] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.
- [44] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.
- [45] Muhammad Zia ur Rehman, Asim Waris, Syed Gilani, Mads Jochumsen, Imran Niazi, Mohsin Jamil, Dario Farina, and Ernest Kamavuako. Multiday emg-based classification of hand motions with deep learning techniques. *Sensors*, 18(8):2497, 2018.

Chapter 7

Unsupervised Domain Adversarial Self-Calibration for Electromyographic-based Gesture Recognition

7.1 Reference

Ulysse Côté-Allard, Gabriel Gagnon-Turcotte, Angkoon Phinyomark, Kyrre Glette, Erik Scheme[‡], François Laviolette[‡] and Benoit Gosselin[‡]. "Unsupervised Domain Adversarial Self-Calibration for Electromyographic-based Gesture Recognition" arXiv preprint (2019).

[‡]These authors share senior authorship

7.2 Context

This article is a natural progression from the paper presenting the four main dynamic factors within a VR environment. In that last paper, a clear degradation between each recording session was observed. This degradation required the participant to perform periodic recalibration to maintain classifier's performances. Consequently, this article proposes to try to address this issue by using unlabeled data from new session to automatically adapt the classifier and hopefully enhance classification accuracies between sessions.

7.3 Résumé

Cet article présente un nouvel algorithme d'apprentissage profond adaptatif, nommé SCADANN, pour la reconnaissance de gestes à long terme basé sur l'EMG de surface. SCADANN est

comparé à trois algorithmes d’adaptation par domaine contradictoire qui sont l’état de l’art. Les algorithmes sont évalués sur la tâche particulièrement difficile de s’adapter au signal EMG lorsque plusieurs jours se sont écoulés entre chaque adaptation. Globalement, SCADANN améliore systématiquement les performances du classificateur sans adaptation et se classe premier parmi les algorithmes testés dans presque tous les cas.

7.4 Abstract

Surface electromyography (sEMG) provides an intuitive and non-invasive interface from which to control machines. However, preserving the myoelectric control system’s performance over multiple days is challenging, due to the transient nature of the signals obtained with this recording technique. In practice, if the system is to remain usable, a time-consuming and periodic recalibration is necessary. In the case where the sEMG interface is employed every few days, the user might need to do this recalibration before every use. Thus, severely limiting the practicality of such a control method.

Consequently, this paper proposes tackling the especially challenging task of unsupervised adaptation of sEMG signals when multiple days have elapsed between each recording by introducing Self-Calibrating Asynchronous Domain Adversarial Neural Network (SCADANN). SCADANN is compared with two state-of-the-art self-calibrating algorithms developed specifically for deep learning within the context of EMG-based gesture recognition and three state-of-the-art domain adversarial algorithms. The comparison is made both on offline and a dynamic datasets (20 participants per dataset), using two different deep network architectures with two different input modalities (temporal spatial descriptors and spectrograms). Overall, SCADANN is shown to substantially and systematically improves classifier’s performances over no recalibration and obtains the highest average accuracy for all tested cases across all methods.

7.5 Introduction

Robots have become increasingly prominent in the lives of human beings. As a result, the way in which people interact with machines is constantly evolving towards better synergies between human intention and machine action. The ease of transcribing intention into commands is highly dependent on the type of interface and its implementations [11]. Within this context, muscle activity offers an attractive and intuitive way to perform gesture recognition as a guidance method [4, 39]. Such activity can be recorded from surface electromyography (sEMG), a non-invasive technique widely adopted both for prosthetic control and in research as a way to seamlessly interact with machines [50, 55]. Artificial intelligence can then be leveraged as the bridge between these biological signals and a robot input guidance.

Current state-of-the-art algorithms in gesture recognition routinely achieve accuracies above

95% for the classification of offline, within-day datasets [15, 24]. However, many practical issues still need to be solved before implementing these type of algorithms into functional applications [50, 28]. Electrode shift and the transient nature of the sEMG signals are among the main obstacles to a robust and widespread implementation of real-time sEMG-based gesture recognition [50]. In practice, this means that users of current myoelectric systems need to perform periodic recalibration of their device so as to retain their usability. To address the issue of real-time myoelectric control, researchers have proposed rejection-based methods where a gesture is predicted only when a sufficient level of certainty is achieved [52, 2]. While these types of methods have been shown to increase online usability, they do not directly address the inherent decline in performance of the classifier over time. One way to address this challenge is to leverage transfer learning algorithms to periodically recalibrate the system with less data than normally required [49, 17]. While these types of methods reduce the burden on the user, they still require said user to periodically record labeled data.

This work focuses on the problem of across-day sEMG-based gesture recognition both within an offline and dynamic setting. In particular, this work considers the situation where several days are elapsed between each recording session. Such a setting naturally arises when sEMG-based gesture recognition is used for video games, artistic performances or, simply, to control non-essential devices [58, 55, 5]. In contrast to within-day or even day-to-day adaptation, this work’s setting is especially challenging as the change in the signal between two sessions is expected to be substantially greater and no intermediary data is available to bridge this gap. The goal is then for the classifier to be able to adapt over-time using the unlabeled data obtained from the myoelectric system. Such a problem can be framed within an unsupervised domain adaptation setting [1] where there exists an initial labeled dataset on which to train, but the classifier then has to adapt to unlabeled data from a different, but similar distribution. Huang et al. [30] proposes to use this setting to update a support vector machine by replacing old examples forming the support vectors with new unlabeled examples which are close to the old ones (and assigning the same label as the example that is replaced). Other authors [27] propose instead to periodically retrain an LDA by updating the training dataset itself. The idea is to replace old examples with new, near (i.e. small distance within the feature space) ones. Such methods, however, are inherently restricted to single-day use as they rely on smooth and small signal drift to update the classifier. Additionally, these types of methods do not leverage the potentially large quantity of unlabeled data generated. In contrast, deep learning algorithms are well suited to scale to large amounts of data and were shown to be more robust to between-day signal drift than LDA, especially as the amount of training data increases [63]. Within the field of image recognition, deep learning-based unsupervised domain adaptation has been extensively studied. A popular approach to this problem is domain adversarial training popularized by DANN [1, 23]. The idea behind DANN is to learn a feature representation which favors class separability of the labeled dataset, while simultaneously hindering domain separability (i.e. differentiation between the labeled and unlabeled examples).

See Section 7.7 for details. Building on DANN, the VADA (for Virtual Adversarial Domain Adaptation) algorithm [53] proposes to also minimize the cluster assumption violations on the unlabeled dataset [62] (i.e. decision boundary should avoid area of high data density). Another state-of-the-art algorithm, but this time for non-conservative unsupervised domain adaptation (i.e. the final model might not be good at classifying the original data), is DIRT-T (for Decision-boundary Iterative Refinement Training with a Teacher), which starting from the output of VADA, removes the labeled data and iteratively tries to continue minimizing the cluster assumption. A detailed explanation of DANN, VADA and DIRT-T is given in Section 7.7. These three state-of-the-art domain adversarial algorithms achieve a two-digit accuracy increase on several difficult image recognition benchmarks [53] compared to the non-adapted deep network. This work thus proposes to test these algorithms on the challenging problem of multiple-day sEMG-based gesture recognition both within an offline and dynamic setting.

An additional difficulty of the setting considered in this work is that real-time myoelectric control imposes strict limitations in relation to the amount of temporal data which can be accumulated before each new prediction. The window’s length requirement has a direct negative impact on the performance of classifiers [54, 2]. This is most likely due to the fact that temporally neighboring segments most likely belong to the same class [6, 61]. In other words, provided that predictions can be deferred, it should be possible to generate a classification algorithm with improved accuracy (compared to the real-time classifier) by looking at a wider temporal context of the data [2]. Consequently, one potential way of coping with electrode shift and the non-stationary nature of EMG signals for gesture recognition is for the classifier to self-calibrate using pseudo-labels generated from this improved classification scheme. The most natural way of performing this relabeling is using a majority vote around each classifier’s prediction. Xiaolong et al. [61] have shown that such a recalibration strategy significantly improves intra-day accuracy on an offline dataset for both amputees and able-bodied subjects (tested on the NinaPro DB2 and DB3 datasets [7]). However for real-time control, such a majority vote strategy will increase latency, as transitions between gestures inevitably take longer to be detected. Additionally, as the domain divergence over multiple days is expected to be substantially greater than within a single day, ignoring this gap before generating the pseudo-labels might negatively impact the self-recalibrated classifier. Finally, trying to re-label every segment, even when there is no clear gesture detected by the classifier, will necessarily introduce undesirable noise in the pseudo-labels. To address these issues, the main contribution of this paper is the introduction of SCADANN (for Self-Calibrating Asynchronous Domain Adversarial Neural Network), a deep learning-based algorithm, which leverages domain adversarial training and the unique properties of real-time myoelectric control for inter-day self-recalibration.

This paper is organized as follows. An overview of the datasets and the deep network architecture

employed in this work is provided in Section 7.6. Section 7.7 presents the domain adaptation algorithm considered in this work, while Section 7.8 thoroughly describes SCADANN alongside the two most popular sEMG-based unsupervised adaptation algorithms. Finally, these three algorithms are compared alongside DANN, VADA and DIRT-T with the non-adaptive network in Section 7.9 and their associated discussions are shown in Section 7.10.

7.6 Datasets and Network’s Architecture

This work employs the *3DC Dataset* [16] for architecture building and hyperparameter optimization and the *Long-term 3DC Dataset* [17] for training and testing the different algorithms considered. Both datasets were recorded using the *3DC Armband* [16]; a wireless, 10-channel, dry-electrode, 3D printed sEMG armband. The device samples data at 1000 Hz per channel, allowing to take advantage of the full spectra of sEMG signals [46].

As stated in [16, 17], the data acquisition protocol of the 3DC Dataset and Long-term 3DC Dataset were approved by the Comités d’Éthique de la Recherche avec des êtres humains de l’Université Laval (approval number: 2017-0256 A-1/10-09-2018 and 2017-026 A2-R2/26-06-2019 respectively), and informed consent was obtained from all participants.

7.6.1 Long-term 3DC Dataset

The Long-term 3DC Dataset features 20 able-bodied participants (5F/15M) aged between 18 and 34 years old (average 26 ± 4 years old) performing eleven gestures (shown in Figure 7.1). Each participant performed three recording sessions over a period of fourteen days (in seven-day increments). Each recording session is divided into a *Training Recording* and two *Evaluation Recordings*. For each new session, the participants were the ones placing the armband on their forearm at the beginning of each session (introducing small electrode shift between each session).

The Long-term 3DC Dataset was recorded within a virtual reality environment in conjunction with the leap motion camera. The usefulness of the VR environment was three fold. First, it allowed to more intuitively communicate requested gesture intensity and position to the participant. Second, it allowed to replace the arm of the participant with a *virtual prosthetic*, which provided direct and intuitive feedback (gesture held, intensity and position) to the participant. Third, it allowed the gamification of the experimental protocol, which greatly facilitated both recruitment and participant retention. During recording, the leap motion, in conjunction with an image-based convolutional network, served as the real-time controller and as a way to provide feedback without biasing the dataset to a particular EMG-based classifier.

The dataset is thoroughly described alongside a detailed explanation of the VR system and the contributions of the leap motion camera in [17]. A brief overview of the dataset is provided in

the following subsections. A video showing the recording protocol in action is also available at the following link: <https://www.youtube.com/watch?v=BnDwcw8ol6U>.

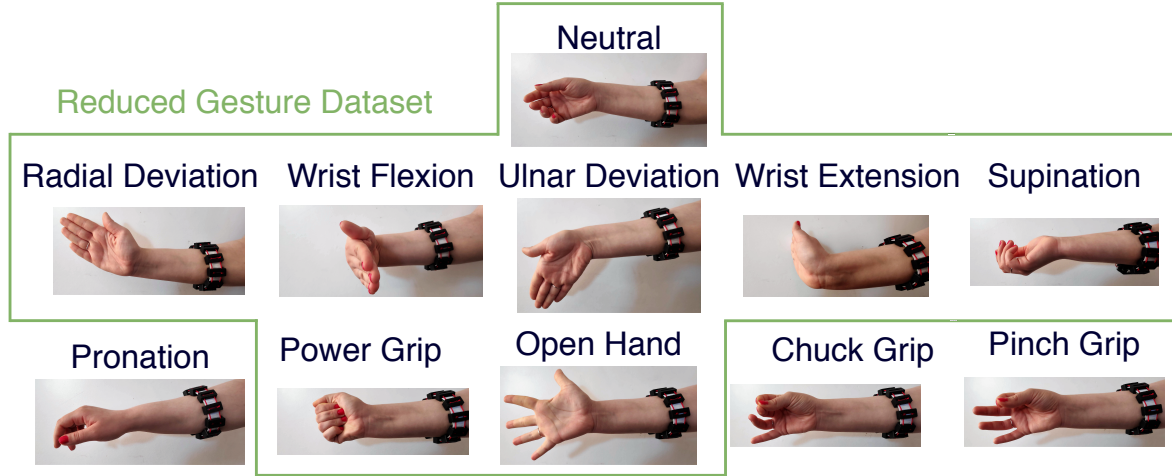


Figure 7.1 – The eleven hand/wrist gestures recorded in the *Long-term 3DC dataset* and the *3DC Dataset*. The gestures included within the *Reduced Long-term 3DC Dataset* are encompassed within the green line (7 gestures totals).

Training Recording

During the Training Recording, each participant was standing and held their forearm, unsupported, parallel to the floor, with their hand relaxed (neutral position). Starting from this neutral position, each participant was asked to perform and hold each gesture for a period of five seconds. This was referred to as a *cycle*. Two more such cycles were recorded. In this work, the first two cycles are used for training, while the last one is used for testing (unless specified otherwise). Note that in the original dataset, four cycles are recorded for each participant, with the second one recording the participant performing each gesture with maximal intensity. This second cycle was removed for this work to reduce confounding factors. In other words, cycle two and three in this work correspond to cycle three and four in the original dataset.

In addition to the eleven gestures considered in the Long-term 3DC Dataset, a reduced dataset from the original Long-term Dataset containing seven gestures is also employed. This *Reduced Long-term 3DC Dataset* is considered as it could more realistically be implemented on a real-world system given the current state of the art of EMG-based hand gesture recognition. The following gestures are selected to form the reduced dataset: neutral, open hand, power grip, radial/ulnar deviation and wrist flexion/extension. They were selected as they were shown to be sufficient in conjunction with orientation data to control a 6 degree-of-freedom robotic arm in real-time [5].

Evaluation Recording

During the Evaluation Recordings, the participants were asked to perform a specific gesture at a specific intensity (low, medium and high intensity based on their corresponding maximal gesture intensity) and at a random position (a point within reach of the participant’s extended arm at a maximum angle of ± 45 and ± 70 degrees in pitch and yaw respectively). A new gesture, intensity and position were randomly asked every five seconds. Each Evaluation Recording lasted three and a half minutes and two such recordings were performed by each participant for each recording session (total of six Evaluation Recordings per participant). The Evaluation Recordings provide a dynamic dataset which includes the transitions between the different gestures and the four main dynamic factors [50] (i.e. contraction intensity, inter-day recording, electrode shifts and limb position) in sEMG-based gesture recognition. Note that while the participants received visual feedback within the VR environment in relation to the held gesture, limb position and gesture intensity, the performed gestures were classified using the leap motion camera [29] in order to avoid bias in the dataset towards a particular EMG-based classifier. In other words, the controller used by the participants during the Evaluation Recordings is distinct and independent from the sEMG-based gesture recognition algorithms considered in this manuscript, which is the main difference between the dynamic dataset considered and a real-time dataset. In this work, the first evaluation recording of a given session was employed as the unlabeled training dataset for the algorithms presented in Section 7.7 and 7.8, while the second evaluation recording was used for testing.

Data Pre-processing

This work aims at studying unsupervised recalibration of myoelectric control systems. Consequently, the input latency is a critical factor to consider. The optimal guidance latency was found to be between 150 and 250 ms [54]. As such, the data from each participant is segmented into 150 ms frames with an overlap of 100 ms. Each segment thus contains 10×150 (*channel \times time*) data points. The segmented data is then band-pass filtered between 20-495 Hz using a fourth-order butterworth filter.

Given a segment, the spectrogram for each sEMG channel are then computed using a 48 points Hann window with an overlap of 14 yielding a matrix of 4×25 (*time \times frequency*). The first frequency band is then removed in an effort to reduce baseline drift and motion artifacts. Finally, following [14], the time and channel axis are swapped such that an example is of the shape $4 \times 10 \times 24$ (*time \times channel \times frequency*). Spectrograms were selected as inputs for the ConvNet presented in Section 7.6.3, as they have been shown to obtain competitive performance on a wide variety of datasets [15, 61, 16] and in the control of a robotic arm in real-time [5]. In addition, they are relatively inexpensive to compute and allow for faster training of a ConvNet when compared to the raw sEMG signal due to the relatively low dimensionality of the obtained input images from the spectrograms.

7.6.2 3DC Dataset

The 3DC Dataset features 22 able-bodied participants and is employed for architecture building and hyperparameter selection. This dataset, presented in [16], includes the same eleven gestures as the Long-term 3DC Dataset. Its recording protocol closely matches the Training Recording description (Section 7.6.1), with the difference being that two such recordings were recorded for each participant (within the same day). This dataset was preprocessed as described in Section 7.6.1.

7.6.3 Convolutional Network’s Architecture

A small and simple ConvNet’s architecture inspired from [13] and presented in Figure 7.2 was selected to reduce potential confounding factors. The ConvNet’s architecture contains four *blocks* followed by a global average pooling and two heads. The first head is used to predict the gesture held by the participant. The second head is only activated when employing domain adversarial algorithms (see Section 7.7 and 7.8 for details). Each block encapsulates a convolutional layer [36], followed by batch normalization [31], leaky ReLU [60] and dropout [31] (set to $p=0.5$) [22].

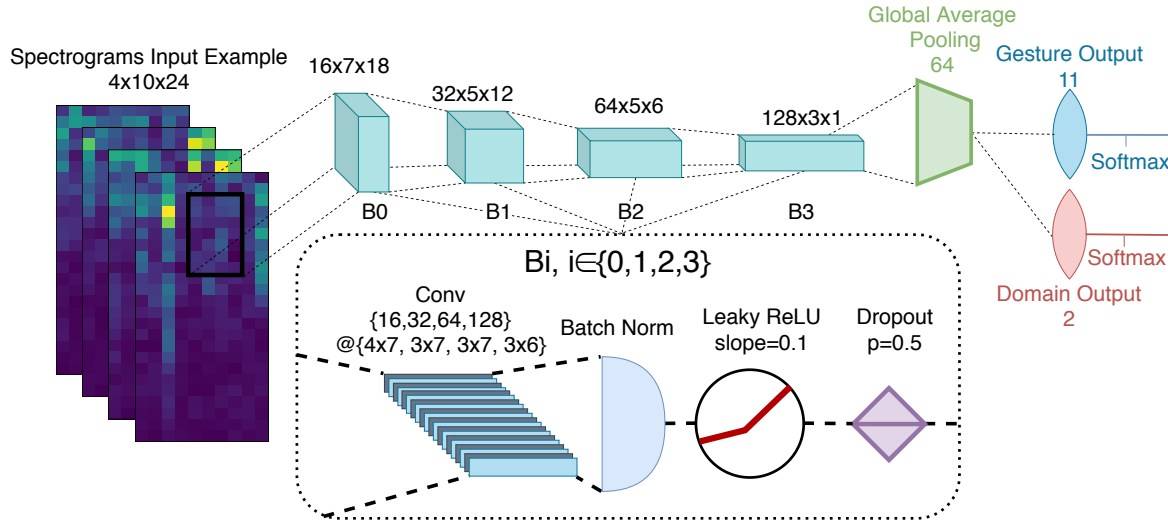


Figure 7.2 – The ConvNet’s architecture employing 206 548 learnable parameters. In this figure, B_i refers to the i th block ($i \in \{0, 1, 2, 3\}$). Conv refers to a convolutional layer. When working with the reduced dataset, the number of output neurons from the gesture-head are reduced to seven.

ADAM [34] is employed for the ConvNet’s optimization with batch size of 512. The learning rate ($lr=0.001316$) was selected with the 3DC Dataset by random search [8] using a uniform random distribution on a logarithm scale between 10^{-5} and 10^1 and 100 candidates (each candidate was evaluated 5 times). Early stopping, with a patience of 10 epochs, is also applied by using 10% of the training dataset as a validation set. Additionally, learning rate annealing,

with a factor of five and a patience of five, was also used. Within this paper, this classifier will be referred to as Spectrogram ConvNet.

Note that the ConvNet’s architecture implementation, written with PyTorch [41], is made [readily available here](https://github.com/UlysseCoteAllard/LongTermEMG) (<https://github.com/UlysseCoteAllard/LongTermEMG>).

7.6.4 Temporal Spatial Descriptors Deep Network

Due to the ubiquity of handcrafted feature sets within the field of EMG-based gesture recognition, a deep network taking Temporal Spatial Descriptors (TSD) as input is also considered. TSD is a handcrafted feature set proposed by Khushaba et al. [33] which achieved state-of-the-art results in EMG-based gesture classification. A short overview of this feature set is given in Appendix 7.A and the interested reader is encouraged to consult [33] for a detailed description. Note that before computing the gesture, the data is preprocessed as described in Section 7.6.1 (without the spectrogram part).

The deep network architecture was again selected to be as simple as possible and is comprised of 3 fully connected layers each 200 neurons wide. Each layer also applies batch normalization, leaky ReLU (slope 0.1) as the activation function and dropout ($p=0.5$). The training procedure is the same as for the Spectrogram ConvNet. ADAM is also employed with a learning rate of 0.002515 (found by cross-validation on the 3DC Dataset using the same hyperparameter as the Spectrogram ConvNet). The PyTorch implementation of the Deep Network, which will be referred to as TSD DNN for the remainder of this paper, is also made [readily available here](https://github.com/UlysseCoteAllard/LongTermEMG) (<https://github.com/UlysseCoteAllard/LongTermEMG>).

7.6.5 Calibration Methods

This work considers three types of calibration for long-term classification of sEMG signals: No Calibration, Recalibration and Unsupervised Calibration. In the first case, the network is trained solely from the data of the first session. In the Recalibration case, the model is re-trained at each new session with the new labeled data. Unsupervised Calibration is similar to Recalibration, but the dataset used for recalibration is unlabeled. Section 7.7 and 7.8 presents the unsupervised calibration algorithms considered in this work.

7.7 Unsupervised Domain Adaptation

Domain adaptation is an area in machine learning which aims at learning a discriminative predictor from two datasets (source and target datasets) coming from two different, but related, distributions [23] (referred to as \mathcal{D}_s and \mathcal{D}_t). In the unsupervised case, one of the datasets is labeled (and comes from \mathcal{D}_s), while the second is unlabeled (and comes from \mathcal{D}_t).

Within the context of myoelectric control systems, labeled data is obtained through a user’s conscious calibration session. However, due to the transient nature of sEMG signals [50, 37], classification performance tends to degrade over time. This naturally creates a burden for the user who needs to periodically recalibrate the system to maintain its usability [37, 19]. During normal usage, however, unlabeled data is constantly generated. Consequently, the unsupervised domain adaptation setting naturally arises by defining the *source dataset* as the labeled data of the calibration session and the *target dataset* as the unlabeled data generated by the user during control.

The PyTorch implementation of the domain adversarial algorithms is mainly based on [40].

7.7.1 Domain-Adversarial Training of Neural Networks

The Domain-Adversarial Neural Network (DANN) algorithm proposes to predict on the target dataset by learning a representation from the source dataset that makes it hard to distinguish examples from either distribution [1, 23]. To achieve this objective, DANN adds a second head (which may be comprised of one or more layers) to the network. This head, referred to as the *domain classification head*, receives the features from the last feature extraction layer of the network (in this work case; from the global average pooling layer). The goal of this second head is to learn to discriminate between the two domains (source and target). However, during backpropagation, the gradient computed from the domain loss is multiplied by a negative constant (-1 in this work). This gradient reversal explicitly forces the feature distribution of the domains to be similar. The backpropagation algorithm proceeds normally for the original head (classification head). The two losses are combined as follows: $\mathcal{L}_y(\theta; \mathcal{D}_s) + \lambda_d \mathcal{L}_d(\theta; \mathcal{D}_s, \mathcal{D}_t)$, where θ is the classifier’s parametrization, \mathcal{L}_y and \mathcal{L}_d are the prediction and domain loss respectively. λ_d is a scalar that weights the domain loss (set to 0.1 in this work).

7.7.2 Decision-boundary Iterative Refinement Training with a Teacher

Decision-boundary Iterative Refinement Training with a Teacher (DIRT-T) is a two-step domain-adversarial training algorithm which achieves state-of-the-art results on a variety of domain adaptation benchmarks [53].

First step

During the first step, referred to as VADA (for Virtual Adversarial Domain Adaptation) [53]), training is done using DANN as described previously (i.e. using a second head to discriminate between domains). However, with VADA, the network is also penalized when it violates the cluster assumption on the target. This assumption states that data belonging to the same cluster in the feature space share the same class. Consequently, decision boundaries should

avoid crossing dense regions. As shown in [26], this behavior can be achieved by minimizing the conditional entropy with respect to the target distribution:

$$\mathcal{L}_c(\theta; \mathcal{D}_t) = \mathbb{E}_{x \sim \mathcal{D}_t} [h_\theta(x)^T \ln(h_\theta(x))] \quad (7.1)$$

Where θ is the parametrization of a classifier h .

In practice, \mathcal{L}_c must be estimated from the available data. However, as noted by [26], such an approximation breaks if the classifier h is not locally-Lipschitz (i.e. an arbitrary small change in the classifier's input produces an arbitrarily large change in the classifier's output). To remedy this, VADA proposes to explicitly incorporate the locally-Lipschitz constraint during training via Virtual Adversarial Training (VAT) [38]. VAT generates new "virtual" examples at each training batch by applying small perturbation to the original data. The average maximal Kullback-Leibler divergence ($\mathbf{D}_{\mathbf{KL}}$) [35] is then minimized between the real and virtual examples to enforce the locally-Lipschitz constraint. In other words, VAT adds the following function to minimize during training:

$$\mathcal{L}_v(\theta; \mathcal{D}) = \mathbb{E}_{x \sim \mathcal{D}} \left[\max_{\|r\| \leq \epsilon} \mathbf{D}_{\mathbf{KL}}(h_\theta(x) || h_\theta(x+r)) \right] \quad (7.2)$$

As VAT can be seen as a form of regularization, it is also applied for the source data. In summary, the combined loss function to minimize during VADA training is:

$$\begin{aligned} \min_{\theta} \mathcal{L}_y(\theta; \mathcal{D}_s) + \lambda_d \mathcal{L}_d(\theta; \mathcal{D}_s, \mathcal{D}_t) + \lambda_{vs} \mathcal{L}_v(\theta; \mathcal{D}_s) + \\ \lambda_{vt} \mathcal{L}_v(\theta; \mathcal{D}_t) + \lambda_c \mathcal{L}_c(\theta; \mathcal{D}_t) \end{aligned} \quad (7.3)$$

Where the importance of each additional loss function is weighted with a hyperparameter ($\lambda_d, \lambda_{vs}, \lambda_{vt}, \lambda_c$) . A diagram of VADA is provided in Figure 7.3.

Second Step

During the second step, the signal from the source is removed. The idea is then to find a new parametrization that further minimizes the target cluster assumption violation while remaining close to the classifier found during the first step. This process can then be repeated by updating the original classifier with the classifier's parametrization found at each iteration. The combined loss function to minimize during the n th iteration thus becomes:

$$\begin{aligned} \min_{\theta_n} \beta \mathbb{E} [\mathbf{D}_{\mathbf{KL}}(h_{\theta_{n-1}}(x) || h_{\theta_n}(x))] + \\ \lambda_{vt} \mathcal{L}_v(\theta; \mathcal{D}_t) + \lambda_c \mathcal{L}_c(\theta; \mathcal{D}_t) \end{aligned} \quad (7.4)$$

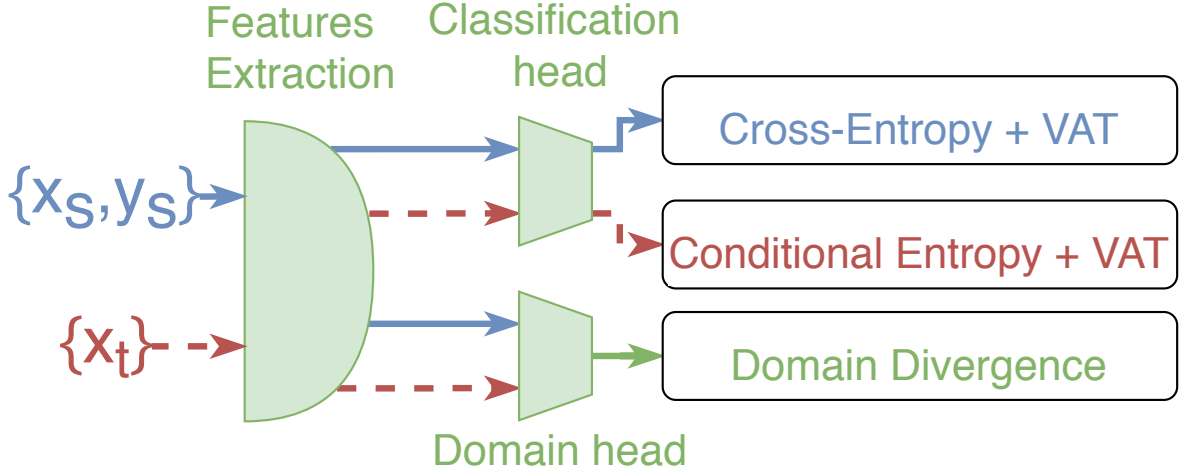


Figure 7.3 – The VADA algorithm which simultaneously tries to reduce the divergence between the labeled source ($\{x_s, y_s\}$) and unlabeled target ($\{x_t\}$) dataset while also penalizing violation of the cluster assumption on the target dataset.

Where β is a hyperparameter which weighs the importance of remaining close to $h_{\theta_{n-1}}$. In practice, the optimization problem of Eq. 7.4 can be approximately solved with a finite number of stochastic gradient descent steps [53]. Following [53], the hyperparameters values are set to $\lambda_d = 10^{-2}$, $\lambda_{vs} = 1$, $\lambda_{vt} = 10^{-2}$, $\lambda_c = 10^{-2}$, $\beta = 10^{-2}$.

Note that, both DANN and VADA were conservative domain adaptation algorithms (i.e. the training algorithms try to generate a classifier that is able to discriminate between classes from both the source and target simultaneously). In contrast, DIRT-T is non-conservative as it ignores the source’s signal during training. In the case where the gap between the source and the target is important, this type of non-conservative algorithm is expected to perform better than its conservative counterparts [53].

7.7.3 Unsupervised Adaptation - Hyperparameters Selection

One challenge in applying unsupervised domain adaptation algorithms is the selection of the hyperparameters associated with the loss functions weights. This is due to the absence of labeled data on the target dataset, which in practice prohibits performing standard hyperparameter selection. One possible solution is to perform the adaptation without explicitly minimizing the distance between the source and target, so that this distance can be used as a measure of adaptation performance [56]. However, such a solution precludes algorithms like the ones considered in this work and so the question of how to best perform hyperparameters selection remains a difficult and open question.

In their work introducing VADA and DIRT-T [53], Shu et al. observed that extensive hyperparameter tuning was not necessary to achieve state-of-the-art performance on the

datasets they were using. Consequently, following this recommendation, the hyperparameters associated with the unsupervised domain adversarial algorithms described in this section used the defaults weights recommended in their respective paper.

7.8 Unsupervised Self-Calibration

Within an unsupervised domain adaptation setting, the classifier’s performance is limited by the unavailability of labeled data from the target domain. However, real-time EMG-based gesture recognition offers a particular situation from which pseudo-labels can be generated from the recorded data by looking at the prediction’s context. These pseudo-labels can then be used as a way for the classifier to perform self-recalibration. Zhai et al. [61] proposed to leverage this context by relabeling the network’s predictions. Let $P(i, j)$ be the softmax value of the network’s output for the j th gesture (associated with the j th output neuron) of the i th example of a sequence. The heuristic considers an array composed of the t segments surrounding example i (included). For each j , the median softmax value over this array is computed:

$$\tilde{P}(i, j) = \text{median}(P(i - t, j), P(i - t + 1, j), \dots, P(i, j), \dots, P(i + t, j)) \quad (7.5)$$

The pseudo-label of i then becomes the gesture j associated with the maximal $\tilde{P}(i, j)$. The median of the softmax’s outputs is used instead of the prediction’s mean to reduce the impact of outliers [61]. This self-calibrating heuristic will be referred to as MV (for Multiple Votes) from now on. As it was the best performing setting, the All-Session recalibration setting (i.e. using all available unlabeled data across sessions) [61] is employed for MV. The hyperparameter t was set to 1 second, as recommended in [61].

This work proposes to improve on MV with a new self-calibrating algorithm, named SCADANN, which can be divided into three steps:

1. Apply DANN to the network using the labeled and newly acquired unlabeled data.
2. Using the adapted network, perform the relabeling scheme described in Section 7.8.1.
3. Starting from the adapted network, train the network with the pseudo-labeled data and labeled data while continuing to apply DANN to minimize domain divergence.

The first step aims at reducing the domain divergence between the labeled recording session and the unlabeled recording to improve classification performance of the network.

The second step uses the pseudo-labeling heuristic described in Section 7.8.1. In addition to using the prediction’s context to enhance the relabeling process, the proposed heuristic introduces two improvements compared to [61]:

First, the heuristic tries to detect transition from one gesture to another. Then, already relabeled predictions falling within the transition period are vetted and possibly relabeled to better reflect when the actual transition occurred. This improvement aims at addressing two problems. First, the added latency introduced by majority-voting pseudo-labeling is removed. Second, this relabeling can provide the training algorithm with gesture transition examples. This is of particular interest as labeled transition examples are simply too time consuming to produce, especially considering the current need for periodic recalibration (g gestures create $g \times (g - 1)$ transitions to record). Introducing pseudo-labeled transition examples within the target dataset, could allow the network to detect transitions more rapidly and thus reduce the system latency. In turn, due to this latency’s reduction, window’s length could be increases to improve the overall system’s performance.

The second improvement, introduces the notion of *stability* to the network’s predictions. Using this notion, the heuristic removes examples that are more likely to be relabeled falsely from the pseudo-labeled dataset. This second improvement is essential for a realistic implementation of self-calibrating algorithms, as otherwise the pseudo-labeled dataset would rapidly be filled with an important quantity of noise. This would result in a rapidly degenerating network as self-calibration is performed iteratively.

The third step re-calibrates the network using the labeled and pseudo-labeled dataset in conjunction. DANN is again employed to try to obtain a similar feature representation between the source and target datasets. The source dataset contains the labeled dataset alongside all the pseudo-labeled data from prior sessions, while the target dataset contains the pseudo-labeled data from the current session. The difference with SCADANN’s first step is that the network’s weights are also optimized in relation to the cross-entropy loss calculated from the newly generated pseudo-labels. If only the pseudo-labeled dataset was employed for recalibration, the network performance would rapidly degrade from being trained only with noisy labels and possibly without certain gestures (i.e. nothing ensure that the pseudo-labeled dataset is balanced or even contains all the gestures). Early stopping is performed using part of the newly generated pseudo-labels.

7.8.1 Proposed Pseudo-labels Generating Heuristic

For concision’s sake, the pseudo-code for the proposed relabeling heuristic is presented in Appendix 7.B-Algorithm 2. Note also that a python implementation of SCADANN (alongside the pseudo-labeling heuristic) is available in the previously mentioned online [repository](#).

The main idea behind the heuristic is that if the new prediction is different than the previous

one, the state goes from *stable* to *unstable*. During the stable state, the prediction of the considered segment is added to the pseudo-label array. During the unstable state, all the network’s output (after the softmax layer) are instead accumulated in a second array. When this second array contains enough segments (hyperparameter sets to 1.5s), the class associated with the output neuron with the highest median value is defined as the new possible stable class. The new possible stable class is confirmed if the median percentage of this class (compared with the other classes) is above a certain threshold (85% and 65% for the seven and eleven gestures dataset respectively (selected using the 3DC dataset)). If this threshold is not achieved, the oldest element in the second array is removed and replaced with the next element. Note that the computation of the new possible stable class using the median is identical to MV.

When the new possible class is confirmed, the heuristic first verify if it was in the unstable state for too long (2s in this work). If it was, all the predictions accumulated during the unstable state are removed. Otherwise, if the new stable state class is different than before it means that a gesture’s transition probably occurred. Consequently, the heuristic goes back in time before the instability began (maximum of 0.5s in this work) and looks at the derivative of the entropy calculated from the network’s softmax output to determine when the network started to be affected by the gesture’s transition. All the segments from this instability period (and adding the relevant segments from the look-back step) are then relabeled as the new stable state class found. If instead the new stable state class is identical to the previous one, only the segments from the instability period are relabeled. The heuristic then returns to its stable state.

7.8.2 SCADANN - Hyperparameters Selection

On the surface, SCADANN introduces several hyperparameters whose selection, within an unsupervised domain adaptation paradigm, is not straightforward. The majority of the introduced hyperparameters, however, have a meaningful interpretation within the context of EMG-based gesture recognition. In other words, reasonable values can be assigned to them without performing detailed data-driven hyperparameter selection. In addition, because these newly introduced hyperparameters are solely related to the pseudo-labeling aspect of the work, a labeled dataset (in this work case the 3DC Dataset) can be leveraged to perform hyperparameter selection.

7.8.3 Adaptive Batch Normalization

For the sake of completeness, in addition to the five previously mentioned adaptation algorithms, this work also considers Adaptive Batch Normalization (AdaBN) [20, 19]. AdaBN is an unsupervised domain adaptation algorithm which was successfully applied to EMG-based gesture recognition in [19]. The hypothesis behind AdaBN is that the label-related information (the difference between gestures) can be encapsulated within the weights of the network, while

the domain-related information (the difference between sessions) are contained within the batch normalization (BN) statistics. In practice, this means that the adaptation is done by feeding the unlabeled examples from the target dataset to the network to update the BN statistics. Note that within this work’s setting, as only one session is contained within the source dataset and inter-user classification is not considered, the multi-stream aspect proposed in [19] cannot be applied.

7.9 Experiments and results

As suggested in [18], a two-step statistical procedure is used whenever multiple algorithms are compared against each other. First, Friedman’s test ranks the algorithms amongst each other. Then, Holm’s post-hoc test is applied ($n = 20$) using the No Calibration setting as a comparison basis. Additionally, Cohen’s d_z [12] is employed to determine the effect size of using one of the self-supervised algorithm over the No Calibration setting. To better contextualize the performance of the basic ConvNet used in this work, a comparison between the ConvNet and 6 widely used features ensembles within the field of sEMG-based gesture recognition is performed. For the sake of concision, this comparison is given in Appendix 7.A.

7.9.1 Training Recording

In this subsection, all training were performed using the first and second cycles of the relevant Training Recording, while the third cycle was employed for testing. All 20 participants completed three Training Recordings and only the labels from the first Training Recording are used (the data from the other Training Recordings are used without labels for the unsupervised recalibrations algorithms when relevant). The time-gap between each Training Recording was around seven days (14-day gap between session 1 and 3). Note that for the first session, all algorithms are equivalent to the No Calibration scheme and consequently perform the same.

Offline Seven Gestures Reduced Dataset

Table 7.1 shows a comparison of the No Calibration setting alongside the three DA algorithms, AdaBN, MV and SCADANN for both the Spectrogram ConvNet and the TSD DNN.

Offline Eleven Gestures Dataset

Table 7.2 compares the No Calibration setting with the three DA algorithms, AdaBN, MV and SCADANN for both networks. Figure 7.4 shows a histogram of the accuracy obtained by the TSD DNN for the No Calibration, SCADANN and the Recalibration methods.

Table 7.1 – Offline accuracy for seven gestures

Spectrogram ConvNet							
	No Cal	DANN	VADA	Dirt-T	AdaBN	MV	SCADANN
Session 0	93.58%	N\A	N\A	N\A	N\A	N\A	N\A
STD	4.58%	N\A	N\A	N\A	N\A	N\A	N\A
Session 1	71.10%	72.76%	73.35%	74.28%	72.61%	74.45%	75.50%
STD	22.90%	26.00%	25.48%	24.42%	25.95%	24.03%	25.41%
Friedman Rank	4.85	4.73	4.25	3.70	5.23	2.78	2.48
H0	N\A	1	1	1	1	0 (0.01193)	0 (0.00305)
Cohen’s Dz	N\A	0.19	0.24	0.36	0.16	0.62	0.52
Session 2	68.75%	74.49%	75.55%	75.52%	76.02%	70.01%	77.22%
STD	22.58%	22.73%	22.76%	23.55%	23.10%	24.82%	22.50%
Friedman Rank	5.60	4.40	4.03	3.40	2.95	4.68	2.95
H0	N\A	1	1	0 (0.00512)	0 (0.00063)	1	0 (0.00063)
Cohen’s Dz	N\A	0.73	0.77	0.70	0.79	0.22	0.92
TSD DNN							
	No Cal	DANN	VADA	Dirt-T	AdaBN	MV	SCADANN
Session 0	96.39%	N\A	N\A	N\A	N\A	N\A	N\A
STD	3.20%	N\A	N\A	N\A	N\A	N\A	N\A
Session 1	78.14%	83.15%	80.90%	80.94%	84.37%	83.01%	84.91%
STD	18.49%	15.47%	15.46%	14.06%	14.64%	19.43%	16.09%
Friedman Rank	5.45	4.03	4.95	4.68	3.00	3.48	2.43
H0	N\A	1	1	1	0 (0.00168)	0 (0.01536)	0 (0.00006)
Cohen’s Dz	N\A	0.90	0.37	0.22	0.88	0.86	0.84
Session 2	79.78%	84.73%	84.50%	82.16%	85.91%	81.47%	88.20%
STD	19.06%	19.38%	17.37%	17.68%	19.06%	19.23%	17.55%
Friedman Rank	5.20	3.93	4.20	5.18	3.23	4.15	2.13
H0	N\A	1	1	1	0 (0.01919)	1	0 (0.00004)
Cohen’s Dz	N\A	0.55	0.52	0.28	0.61	0.48	0.81

7.9.2 Evaluation Recording

Eleven Gestures - Dynamic Dataset, offline adaptation

Table 7.3 compares the No Calibration setting with the three DA algorithms, AdaBN, MV and SCADANN for both networks on the second Evaluation Recording of each session, when the labeled and unlabeled data leveraged for training comes from the Training Recordings (as in Section 7.9.1).

Eleven Gestures - Adaptation on the Dynamic Dataset

Table 7.4 presents the comparison between the No Calibration setting and using the first Evaluation Recording of each experiment’s session as the unlabeled dataset for the three DA algorithms, AdaBN, MV and SCADANN.

A histogram of the dynamic dataset’s accuracy of the No Calibration, Recalibrated, SCADANN and Recalibrated SCADANN methods, trained on the TSD DNN, using the first Evaluation Recording of each experimental session as unlabeled data is shown in Figure 7.5. The

Table 7.2 – Offline accuracy for eleven gestures

Spectrogram ConvNet							
	No Cal	DANN	VADA	Dirt-T	AdaBN	MV	SCADANN
Session 0	84.19%	N\A	N\A	N\A	N\A	N\A	N\A
STD	9.12%	N\A	N\A	N\A	N\A	N\A	N\A
Session 1	58.29%	62.27%	62.45%	62.35%	61.83%	60.75%	63.00%
STD	25.33%	24.86%	25.00%	24.99%	25.42%	26.38%	24.84%
Friedman Rank	5.50	3.85	3.83	3.78	4.05	3.55	3.45
H0	N\A	0 (0.04626)	0 (0.04626)	0 (0.04626)	0 (0.04626)	0 (0.02155)	0 (0.01615)
Cohen’s Dz	N\A	0.63	0.63	0.57	0.49	0.93	0.71
Session 2	56.69%	62.08%	62.40%	62.43%	62.49%	58.27%	63.43%
STD	23.04%	22.84%	22.77%	22.69%	22.98%	23.26%	23.03%
Friedman Rank	5.43	3.95	3.65	3.68	3.80	4.45	3.05
H0	N\A	1	0 (0.04684)	0 (0.04684)	1	1	0 (0.00305)
Cohen’s Dz	N\A	0.75	0.77	0.75	0.78	0.53	0.68
TSD DNN							
	No Cal	DANN	VADA	Dirt-T	AdaBN	MV	SCADANN
Session 0	89.95%	N\A	N\A	N\A	N\A	N\A	N\A
STD	8.37%	N\A	N\A	N\A	N\A	N\A	N\A
Session 1	66.16%	72.44%	69.25%	69.14%	73.63%	71.34%	75.40%
STD	22.66%	20.58%	19.51%	16.64%	19.79%	23.41%	20.06%
Friedman Rank	5.65	3.83	4.75	4.88	3.18	3.33	2.40
H0	N\A	0 (0.02265)	1	1	0 (0.00146)	0 (0.00266)	0 (0.00001)
Cohen’s Dz	N\A	0.76	0.36	0.26	0.87	0.92	1.10
Session 2	66.84%	74.30%	73.61%	73.71%	74.99%	69.94%	77.65%
STD	20.53%	20.57%	18.65%	17.26%	21.97%	20.19%	19.52%
Friedman Rank	6.15	4.13	3.75	3.95	2.98	4.70	2.35
H0	N\A	0 (0.00607)	0 (0.00177)	0 (0.00384)	0 (0.00002)	0 (0.03379)	0 (<0.00001)
Cohen’s Dz	N\A	0.82	0.71	0.63	0.80	1.02	1.12

Recalibration SCADANN scheme systematically and significantly ($p < 0.05$) outperforms the Recalibration scheme for all three sessions for both networks, using the Wilcoxon signed rank-test [59, 18], as can be seen from Table 7.5.

7.10 Discussion

The task of performing adaptation when multiple days have elapsed is especially challenging. As a comparison, on the within-day adaptation task presented in [61], MV was able to enhance classification accuracy by 10% on average compared to the No Calibration scheme. Within this work however, the greatest improvement achieved by MV was 3.35% for the Spectrogram ConvNet and 5.18% for the TSD DNN. Overall, the best improvement in this paper was 8.47% and 10.81% both achieved by SCADANN with the Spectrogram ConvNet and TSD DNN respectively. All three tested domain adversarial algorithms were also able to consistently improve the network’s accuracy compared to the No Calibration scheme (the only exception being VADA and Dirt-T for the TSD DNN in Table 7.3). When used to adapt to dynamic unsupervised data, some were even able to achieve a higher overall ranking than SCADANN using the Spectrogram ConvNet. Note however, that the improvements they seem to allow is overall lower than when they are applied on image-based dataset such as MNIST and

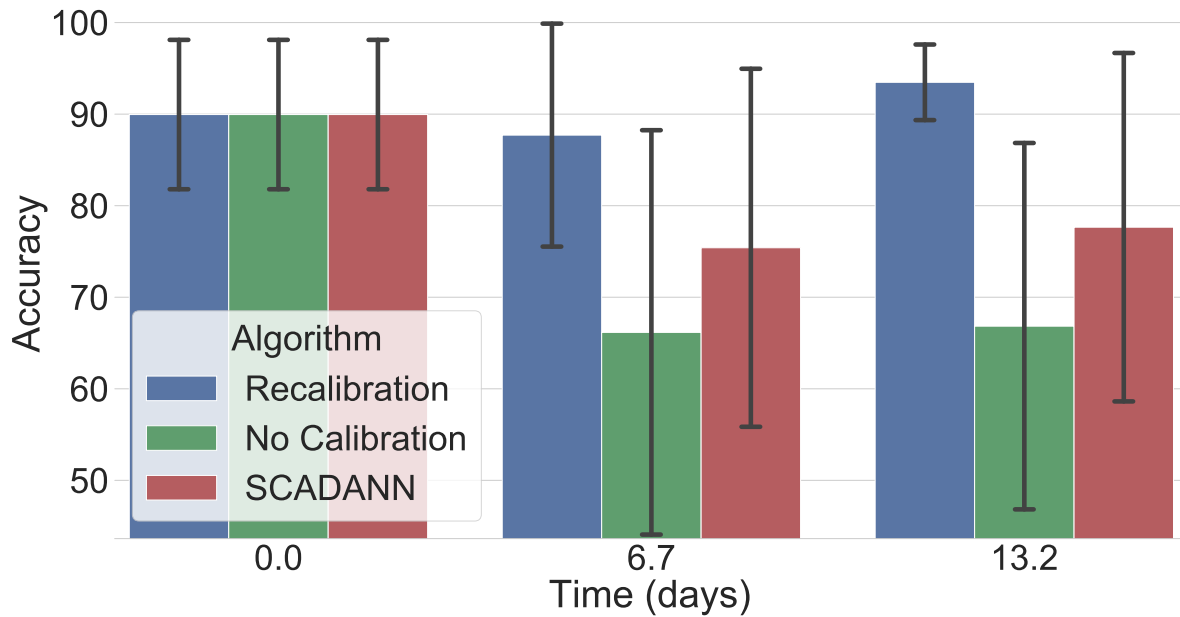


Figure 7.4 – Offline accuracy using the TSD DNN for the eleven gestures in respect to time. The values on the x-axis represent the average number of days elapsed across participants since the first session.

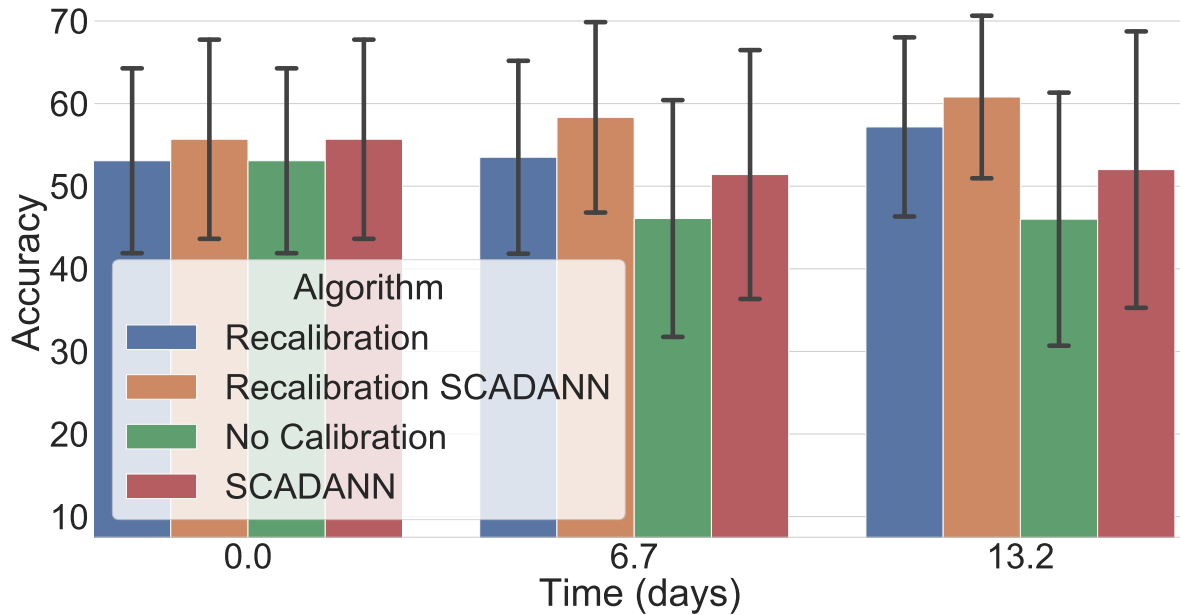


Figure 7.5 – TSD DNN dynamic dataset's accuracy for eleven gestures in respect to time. Training is performed offline with the first Training Recording session. Adaptation takes place on the first Evaluation Recording of the corresponding tested session, while the test set comes from the second Evaluation Recording of the same tested session. The values on the x-axis represent the average number of days elapsed across participants since the first session.

Table 7.3 – Dynamic dataset’s accuracy for eleven gestures using Training Recordings as unlabeled data

Spectrogram ConvNet							
	No Cal	DANN	VADA	Dirt-T	AdaBN	MV	SCADANN
Session 0	47.81%	N\A	N\A	N\A	N\A	N\A	N\A
STD	10.94%	N\A	N\A	N\A	N\A	N\A	N\A
Session 1	38.39%	39.64%	39.52%	39.07%	38.99	39.70%	40.80%
STD	16.65%	17.37%	17.66%	17.56%	17.16%	17.75%	17.77%
Friedman Rank	4.80	3.78	4.10	4.70	4.33	3.30	3.00
H0	N\A	1	1	1	1	1	1
Cohen’s Dz	N\A	0.45	0.32	0.16	0.17	0.54	0.63
Session 2	38.54%	39.87%	40.07%	39.59%	39.53%	40.98%	42.26%
STD	14.65%	15.32%	15.81%	15.43%	15.59%	15.18%	16.34%
Friedman Rank	5.50	4.20	3.60	4.45	4.30	3.35	2.60
H0	N\A	1	0 (0.02166)	1	1	0 (0.00824)	0 (0.00013)
Cohen’s Dz	N\A	0.33	0.35	0.25	0.22	0.84	0.65
TSD DNN							
	No Cal	DANN	VADA	Dirt-T	AdaBN	MV	SCADANN
Session 0	53.08%	N\A	N\A	N\A	N\A	N\A	N\A
STD	11.48%	N\A	N\A	N\A	N\A	N\A	N\A
Session 1	46.09%	48.07%	43.92%	42.75%	47.11%	48.36%	49.09%
STD	14.70%	14.59%	13.45%	12.65%	14.11%	14.30%	14.68%
Friedman Rank	4.50	3.00	5.40	5.80	3.75	2.85	2.70
H0	N\A	1	1	1	1	1	1
Cohen’s Dz	N\A	0.55	-0.37	-0.49	0.21	1.08	0.67
Session 2	46.01%	48.50%	45.69%	45.48%	48.35%	48.17%	50.90%
STD	15.72%	15.80%	14.21%	13.26%	16.17%	17.06%	16.64%
Friedman Rank	4.90	3.85	4.80	5.00	3.73	3.78	1.95
H0	N\A	1	1	1	1	1	0 (0.00009)
Cohen’s Dz	N\A	0.50	-0.05	-0.08	0.42	0.60	0.91

CIFAR [53]. Deep domain adversarial algorithms thus seems to be a promising avenue to explore further, by developing adversarial algorithms specifically for the field of sEMG-based gesture recognition. SCADANN could then easily be augmented by these new algorithms to improve performance further.

In Table 7.4, it can be seen that the performances of MV dropped substantially compared to the other experiments conducted within this paper. A possible explanation is that this was the first time that MV had to adapt using the Dynamic Dataset data. In other words, instead of adapting to a well defined series of examples grouped by gesture, MV had to contend with a continuous data stream including gesture transitions. In contrast, SCADANN actually performed generally better in Table 7.4 than in Table 7.3, which is encouraging as Table 7.4 showcased a more realistic setting for unsupervised recalibration.

It is also important to note that both the general performance and the type of error that the classifier makes can greatly affect the self-calibrating algorithms. While SCADANN partially

Table 7.4 – Dynamic dataset’s accuracy for eleven gestures using the first Evaluation Recording as unlabeled data

Spectrogram ConvNet							
	No Cal	DANN	VADA	Dirt-T	AdaBN	MV	SCADANN
Session 0	47.81%	49.37%	49.36%	49.48%	47.33%	47.68%	49.89%
STD	10.94%	11.24%	11.04%	11.21%	10.45%	11.27%	11.25%
Friedman Rank	4.75	3.80	3.78	3.38	4.95	4.80	2.55
H0	N\A	1	1	1	1	1	0 (0.00490)
Cohen’s Dz	N\A	0.64	0.60	0.53	-0.11	-0.07	0.73
Session 1	38.39%	40.92%	40.73%	40.66%	40.36%	38.60%	41.07%
STD	16.65%	18.51%	18.55%	18.38%	17.77%	17.13%	19.11%
Friedman Rank	5.15	3.10	3.63	3.85	4.25	4.83	3.20
H0	N\A	0 (0.02643)	1	1	1	1	0 (0.02643)
Cohen’s Dz	N\A	0.56	0.53	0.49	0.41	0.13	0.52
Session 2	38.54%	40.78%	40.82%	41.01%	38.15%	40.02%	41.41%
STD	14.65%	16.05%	16.05%	16.29%	15.36%	15.42%	16.45%
Friedman Rank	5.10	2.78	3.28	2.95	5.60	4.10	3.50
H0	N\A	0 (0.00063)	0 (0.00266)	0 (0.00063)	1	1	0 (0.00633)
Cohen’s Dz	N\A	0.50	0.48	0.51	-0.07	0.79	0.48
TSD DNN							
	No Cal	DANN	VADA	Dirt-T	AdaBN	MV	SCADANN
Session 0	53.08%	55.29%	50.42%	53.59%	49.98%	53.67%	55.69%
STD	11.48%	12.09%	10.67%	11.51%	10.90%	11.51%	12.37%
Friedman Rank	4.10	2.60	5.80	4.00	5.50	3.70	2.30
H0	N\A	1	1	1	1	1	1
Cohen’s Dz	N\A	0.71	-0.81	0.13	-0.60	0.30	0.67
Session 1	46.09%	50.65%	46.10%	49.30%	48.12%	47.34%	51.41%
STD	14.70%	14.55%	13.66%	13.81%	14.14%	16.16%	15.46%
Friedman Rank	5.50	2.30	5.55	3.50	4.60	4.40	2.15
H0	N\A	0 (0.00001)	1	0 (0.01366)	1	1	0 (0.00001)
Cohen’s Dz	N\A	1.35	<0.01	0.73	0.36	0.46	1.17
Session 2	46.01%	50.91%	48.33%	50.27%	44.22%	46.90%	52.01%
STD	15.72%	15.88%	14.12%	14.60%	14.58%	16.31%	17.17%
Friedman Rank	5.40	2.65	4.20	3.15	5.90	4.50	2.20
H0	N\A	0 (0.00028)	1	0 (0.00396)	1	1	0 (0.00002)
Cohen’s Dz	N\A	0.98	0.46	0.84	-0.28	0.37	1.32

address the first consideration (by ignoring data that are more likely to be misclassified), the second consideration is harder to address. That is, when the classifier is not only wrong, but is confident in its error and that error spans over a large amount of time, the pseudo-labeling heuristic cannot hope to re-label the segments correctly or even identify this segment of data as problematic. In an effort to address this issue, future works could explore the use of a hybrid IMU/EMG system, as they have been shown to improve gesture recognition accuracy [25, 32]. The use of accelerometer data within the field is generally linked with mechanomyogram

Table 7.5 – Accuracy for the Recalibration and Recalibration SCADANN with eleven gestures on the dynamic dataset using the first Evaluation Recording as unlabeled data

	Spectrogram ConvNet		TSD DNN	
	Recalibration	SCADANN Recalibration	Recalibration	SCADANN Recalibration
Session 0	47.81%	49.89%	53.08%	55.69%
STD	10.94%	11.25%	11.48%	12.37%
H0	0 (0.00642)	0 (0.00642)	0 (0.01000)	0 (0.01000)
Cohen’s Dz	N\A	0.73	N\A	0.67
Session 1	49.54%	53.02%	53.51%	58.34%
STD	11.28%	11.18%	11.98%	11.82%
H0	0 (0.00455)	0 (0.00455)	0 (0.00014)	0 (0.00014)
Cohen’s Dz	N\A	0.88	N\A	1.25
Session 2	52.18%	55.19%	57.18%	60.81%
STD	10.66%	10.15%	11.12%	10.10%
H0	0 (0.00059)	0 (0.00059)	0 (0.00012)	0 (0.00012)
Cohen’s Dz	N\A	1.11	N\A	0.75

*Wilcoxon signed rank test. Null hypothesis rejected when $H_0=0$ ($p<0.05$).

(MMG), which is strongly associated with EMG signals. Recent works [10] however, have shown that, within a human-computer interaction context, accelerometer data can also help recognize different gestures with high accuracy using the positional variance of the different gestures, which is uncharacteristic of MMG. The fusion of these two different modalities could reduce the likelihood of concurrent errors, enabling SCADANN’s relabeling heuristic to generate the pseudo-labels more accurately. Note that, using EMG signals alone, SCADANN’s relabeling heuristic substantially enhanced the pseudo-labels accuracy compared to the one used with MV. As an example, consider the supervised Recalibrating classifier (with the Spectrogram ConvNet) trained on all the training cycles of the relevant Training Recording and tested on the Evaluation Recording. This classifier achieves an average accuracy of 49.84% over 544 263 examples. In comparison, the MV relabeling heuristic achieves 54.28% accuracy over the same number of examples, while the SCADANN relabeling heuristic obtains 61.89% and keeps 478 958 examples using the 65% threshold. When using a threshold of 85%, the accuracy reaches 68.21% and retains 372 567 examples. SCADANN’s improved relabeling accuracy compared to MV is in part due to the look-back feature of the heuristic (when de-activated, SCADANN’s relabeling accuracy drops to 65.23% for the 85% threshold) and its ability to remove highly uncertain sub-sequences of predictions.

The results presented in Table 7.5 are of particular interest as they show that SCADANN actually consistently and significantly improves the classifier’s performance over the recalibration scheme. In other words, SCADANN enhance classifier’s performance without increasing the training time for the participant. In addition, as SCADANN does not impact the classifier’s inference time, SCADANN seems to be an overall net benefit for the classifier’s usability.

7.10.1 Limitations of the study

One major limitation of this work is that the participants were not reacting to the different classifiers being tested (instead using the leap-motion based controller) while performing the task from the Evaluation Recording. This limitation is the only difference between the Dynamic dataset and an online dataset. It is important to note that the participants generally became better at performing the requested task over time (see [17] and Table 7.5). The extent to which this improvement can be attributed to the user’s adaptation to the leap-motion based controller and how much should be attributed to the participants learning how to complete the task better remains unclear. What is known is that the user’s adaptation to the controller substantially affects the real-time control performance of the system [15, 48]. If and how much this adaptation changes in relation to the controller use, however, remains an open question to the best of the authors’ knowledge. Furthermore, this user adaptation would substantially alter the optimal rate of unsupervised calibration and the acceptable extent of said calibration. These new parameters might be better explored within a reinforcement learning [57] framework.

As a direct consequence of not having the adaptation algorithms tested in real-time, another limitation of this work is that the adaptation algorithms were not evaluated using online metrics (e.g. throughput, completion rate, overshoot) [51]. To do so would require recording a separate long-term dataset, as extensive as the one used in this work, for each compared technique so that the different adaptive classifier could be used by the participants in real-time. The difficulty of comparing different adaptation algorithms using online metrics was, in fact, the motivation behind the use of the *Long-term 3DC Dataset* [17] which allows for recording closer to an online setting (compared to offline datasets) without biasing the dataset to a particular EMG-based gesture classification algorithms. Thus, allowing comparison between multiple techniques on a single dataset.

7.11 Conclusion

This paper presents SCADANN, a self-calibrating domain adversarial algorithm for myoelectric control systems. Overall, SCADANN was shown to improve the network’s performance compared to the No Calibration setting in all the tested cases and the difference was significant across all experiments except for one single session. In addition, this work tested three widely used, state-of-the-art, unsupervised domain adversarial algorithms on the challenging task of EMG-based self-calibration. These three algorithms were also found to consistently improve the classifier’s performance compared to the No Calibration setting. MV [61] and AdaBatch [19], two self-calibrating algorithms designed for EMG-based gesture recognition, were also compared to the three DA algorithms and SCADANN. Overall, SCADANN was shown to consistently obtain the best average accuracy amongst the six unsupervised adaptation methods considered in this work both using offline and dynamic datasets. Given the results shown in this paper

and considering that SCADANN has no computational overhead at prediction time, using it to adapt to never-before-seen data is a net benefit both for long-term use but also right after recalibration (as shown in Figure 7.5).

Future works will focus on implementing SCADANN to update in real-time while in use by participants. The interaction between human and machine adaptation and its impact on self-adaptive algorithms like SCADANN will be investigated by leveraging a reinforcement learning framework.

Funding

This research was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC)[funding reference numbers 401220434, 376091307, 114090], the Institut de recherche Robert-Sauvé en santé et en sécurité du travail (IRSST), and the Research Council of Norway, project numbers 262762 and 274996.

7.12 Bibliography

- [1] Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. Domain-adversarial neural networks. *arXiv preprint arXiv:1412.4446*, 2014.
- [2] Ali Al-Timemy, Guido Bugmann, and Javier Escudero. Adaptive windowing framework for surface electromyogram-based pattern recognition system for transradial amputees. *Sensors*, 18(8):2402, 2018.
- [3] Ali H Al-Timemy, Rami N Khushaba, Guido Bugmann, and Javier Escudero. Improving the performance against force variation of emg controlled multifunctional upper-limb prostheses for transradial amputees. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 24(6):650–661, 2015.
- [4] Ulysse Côté Allard, François Nougrou, Cheikh Latyr Fall, Philippe Giguère, Clément Gosselin, François Laviolette, and Benoit Gosselin. A convolutional neural network for robotic arm guidance using semg based frequency-features. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2464–2470. IEEE, 2016.
- [5] Ulysse Côté Allard, François Nougrou, Cheikh Latyr Fall, Philippe Giguère, Clément Gosselin, François Laviolette, and Benoit Gosselin. A convolutional neural network for robotic arm guidance using semg based frequency-features. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2464–2470. IEEE, 2016.

- [6] Sebastian Amsüss, Peter M Goebel, Ning Jiang, Bernhard Graimann, Liliana Paredes, and Dario Farina. Self-correcting pattern recognition system of surface emg signals for upper limb prosthesis control. *IEEE Transactions on Biomedical Engineering*, 61(4):1167–1176, 2013.
- [7] Manfredo Atzori, Arjan Gijsberts, Claudio Castellini, Barbara Caputo, Anne-Gabrielle Mit-taz Hager, Simone Elsig, Giorgio Giatsidis, Franco Bassetto, and Henning Müller. Elec-tromyography data for non-invasive naturally-controlled robotic hand prostheses. *Scientific data*, 1:140053, 2014.
- [8] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(Feb):281–305, 2012.
- [9] Evan Campbell, Angkoon Phinyomark, and Erik Scheme. Current trends and confounding factors in myoelectric control: Limb position and contraction intensity. *Sensors*, 20(6):1613, 2020.
- [10] Evan Campbell, Angkoon Phinyomark, and Erik Scheme. Differences in perspective on inertial measurement unit sensor integration in myoelectric control. *arXiv preprint arXiv:2003.03424*, 2020.
- [11] Alexandre Campeau-Lecours, Ulysse Côté-Allard, Dinh-Son Vu, François Routhier, Benoit Gosselin, and Clément Gosselin. Intuitive adaptive orientation control for enhanced human–robot interaction. *IEEE Transactions on Robotics*, 35(2):509–520, 2018.
- [12] Jacob Cohen. *Statistical power analysis for the behavioral sciences*. Routledge, 2013.
- [13] Ulysse Côté-Allard, Evan Campbell, Angkoon Phinyomark, François Laviolette, Benoit Gosselin, and Erik Scheme. Interpreting deep learning features for myoelectric control: A comparison with handcrafted features. *Frontiers in Bioengineering and Biotechnology*, 8:158, 2020.
- [14] Ulysse Cote-Allard, Cheikh Latyr Fall, Alexandre Campeau-Lecours, Clément Gosselin, François Laviolette, and Benoit Gosselin. Transfer learning for semg hand gestures recognition using convolutional neural networks. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1663–1668. IEEE, 2017.
- [15] Ulysse Côté-Allard, Cheikh Latyr Fall, Alexandre Drouin, Alexandre Campeau-Lecours, Clément Gosselin, Kyrre Glette, François Laviolette, and Benoit Gosselin. Deep learning for electromyographic hand gesture signal classification using transfer learning. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(4):760–771, 2019.
- [16] Ulysse Côté-Allard, Gabriel Gagnon-Turcotte, François Laviolette, and Benoit Gosselin. A low-cost, wireless, 3-d-printed custom armband for semg hand gesture recognition. *Sensors*, 19(12):2811, 2019.

- [17] Ulysse Côté-Allard, Gabriel Gagnon-Turcotte, Angkoon Phinyomark, Erik Scheme, François Laviolette, and Benoit Gosselin. Virtual reality to study the gap between offline and real-time emg-based gesture recognition. *arXiv preprint*, 2019.
- [18] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006.
- [19] Yu Du, Wenguang Jin, Wentao Wei, Yu Hu, and Weidong Geng. Surface emg-based inter-session gesture recognition enhanced by deep domain adaptation. *Sensors*, 17(3):458, 2017.
- [20] Yu Du, Wenguang Jin, Wentao Wei, Yu Hu, and Weidong Geng. Surface emg-based inter-session gesture recognition enhanced by deep domain adaptation. *Sensors*, 17(3):458, 2017.
- [21] Kevin Englehart and Bernard Hudgins. A robust, real-time control scheme for multifunction myoelectric control. *IEEE transactions on biomedical engineering*, 50(7):848–854, 2003.
- [22] Yarín Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.
- [23] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.
- [24] Weidong Geng, Yu Du, Wenguang Jin, Wentao Wei, Yu Hu, and Jiajun Li. Gesture recognition by instantaneous surface emg images. *Scientific reports*, 6:36571, 2016.
- [25] Marcus Georgi, Christoph Amma, and Tanja Schultz. Recognizing hand and finger gestures with imu based motion and emg based muscle activity sensing. In *Biosignals*, pages 99–108, 2015.
- [26] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005.
- [27] Yikun Gu, Dapeng Yang, Qi Huang, Wei Yang, and Hong Liu. Robust emg pattern recognition in the presence of confounding factors: features, classifiers and adaptive learning. *Expert Systems with Applications*, 96:208–217, 2018.
- [28] Maria Hakonen, Harri Piitulainen, and Arto Visala. Current state of digital signal processing in myoelectric interfaces and related applications. *Biomedical Signal Processing and Control*, 18:334–359, 2015.

- [29] David Holz, Kyle Hay, and Michael Buckwald. Electronic sensor, April 14 2015. US Patent App. 29/428,763.
- [30] Qi Huang, Dapeng Yang, Li Jiang, Huajie Zhang, Hong Liu, and Kiyoshi Kotani. A novel unsupervised adaptive learning method for long-term electromyography (emg) pattern recognition. *Sensors*, 17(6):1370, 2017.
- [31] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
- [32] Rami N Khushaba, Ali Al-Timemy, Sarath Kodagoda, and Kianoush Nazarpour. Combined influence of forearm orientation and muscular contraction on emg pattern recognition. *Expert Systems with Applications*, 61:154–161, 2016.
- [33] Rami N Khushaba, Ali H Al-Timemy, Ahmed Al-Ani, and Adel Al-Jumaily. A framework of temporal-spatial descriptors-based feature extraction for improved myoelectric pattern recognition. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(10):1821–1831, 2017.
- [34] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [35] Solomon Kullback. *Information theory and statistics*. Courier Corporation, 1997.
- [36] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
- [37] Jianwei Liu, Xinjun Sheng, Dingguo Zhang, Jiayuan He, and Xiangyang Zhu. Reduced daily recalibration of myoelectric prosthesis classifiers based on domain adaptation. *IEEE journal of biomedical and health informatics*, 20(1):166–176, 2014.
- [38] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.
- [39] Mohammadreza Asghari Oskoei and Huosheng Hu. Myoelectric control systems—a survey. *Biomedical signal processing and control*, 2(4):275–294, 2007.
- [40] Ozan Ciga. Github repository for pytorch implementation of a dirt-t approach to unsupervised domain adaptation, 2019.
- [41] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

- [42] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [43] Angkoon Phinyomark, Rami N Khushaba, Esther Ibáñez-Marcelo, Alice Patania, Erik Scheme, and Giovanni Petri. Navigating features: a topologically informed chart of electromyographic features space. *Journal of The Royal Society Interface*, 14(137):20170734, 2017.
- [44] Angkoon Phinyomark, Rami N Khushaba, and Erik Scheme. Feature extraction and selection for myoelectric control based on wearable emg sensors. *Sensors*, 18(5):1615, 2018.
- [45] Angkoon Phinyomark, Franck Quaine, Sylvie Charbonnier, Christine Serviere, Franck Tarpin-Bernard, and Yann Laurillau. Emg feature evaluation for improving myoelectric pattern recognition robustness. *Expert Systems with applications*, 40(12):4832–4840, 2013.
- [46] Angkoon Phinyomark and Erik Scheme. A feature extraction issue for myoelectric control based on wearable emg sensors. In *2018 IEEE Sensors Applications Symposium (SAS)*, pages 1–6. IEEE, 2018.
- [47] Stefano Pizzolato, Luca Tagliapietra, Matteo Cognolato, Monica Reggiani, Henning Müller, and Manfredo Atzori. Comparison of six electromyography acquisition setups on hand movement classification tasks. *PloS one*, 12(10), 2017.
- [48] Michael A Powell, Rahul R Kaliki, and Nitish V Thakor. User training for pattern recognition-based myoelectric prostheses: Improving phantom limb movement consistency and distinguishability. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(3):522–532, 2013.
- [49] Cosima Prahm, Benjamin Paassen, Alexander Schulz, Barbara Hammer, and Oskar Aszmann. Transfer learning for rapid re-calibration of a myoelectric prosthesis after electrode shift. In *Converging clinical and engineering research on neurorehabilitation II*, pages 153–157. Springer, 2017.
- [50] Erik Scheme and Kevin Englehart. Electromyogram pattern recognition for control of powered upper-limb prostheses: state of the art and challenges for clinical use. *Journal of Rehabilitation Research & Development*, 48(6), 2011.
- [51] Erik J Scheme and Kevin B Englehart. Validation of a selective ensemble-based classification scheme for myoelectric control using a three-dimensional fitts’ law test. *IEEE transactions on neural systems and rehabilitation engineering*, 21(4):616–623, 2012.
- [52] Erik J Scheme, Bernard S Hudgins, and Kevin B Englehart. Confidence-based rejection for improved pattern recognition myoelectric control. *IEEE Transactions on Biomedical Engineering*, 60(6):1563–1570, 2013.

- [53] Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. *arXiv preprint arXiv:1802.08735*, 2018.
- [54] Lauren H Smith, Levi J Hargrove, Blair A Lock, and Todd A Kuiken. Determining the optimal window length for pattern recognition-based myoelectric control: balancing the competing effects of classification error and controller delay. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 19(2):186–192, 2010.
- [55] David St-Onge, Ulysse Côté-Allard, Kyrre Glette, Benoit Gosselin, and Giovanni Beltrame. Engaging with robotic swarms: Commands from expressive motion. *ACM Transactions on Human-Robot Interaction (THRI)*, 8(2):11, 2019.
- [56] Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A Efros. Unsupervised domain adaptation through self-supervision. *arXiv preprint arXiv:1909.11825*, 2019.
- [57] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.
- [58] Aaron Tabor, Scott Bateman, and Erik Scheme. Game-based myoelectric training. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*, pages 299–306. ACM, 2016.
- [59] Frank Wilcoxon. Individual comparisons by ranking methods. In *Breakthroughs in statistics*, pages 196–202. Springer, 1992.
- [60] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.
- [61] Xiaolong Zhai, Beth Jelfs, Rosa HM Chan, and Chung Tin. Self-recalibrating surface emg pattern recognition for neuroprosthesis control based on convolutional neural network. *Frontiers in neuroscience*, 11:379, 2017.
- [62] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.
- [63] Muhammad Zia ur Rehman, Asim Waris, Syed Gilani, Mads Jochumsen, Imran Niazi, Mohsin Jamil, Dario Farina, and Ernest Kamavuako. Multiday emg-based classification of hand motions with deep learning techniques. *Sensors*, 18(8):2497, 2018.

7.A ConvNet’s comparison with Handcrafted feature sets

To better interpret the contributions of this manuscript, it is important to contextualize the ConvNet’s classification performances with respect to the state of the art in sEMG-based gesture recognition.

The comparison considers the simple ConvNet employed throughout this work with six high performing feature sets presented in the following subsections. The python implementation of the different feature sets are available on this work’s [repository](https://github.com/UlysseCoteAllard/LongTermEMG): (<https://github.com/UlysseCoteAllard/LongTermEMG>) and a detailed description of most of the features are given in [15]. Note that the hyperparameters associated with these feature sets employed the ones recommended in their respective original paper.

7.A.1 Hudgin’s features [21]

Hudgin’s features are a set of four features all in the time-domain comprised of: Mean Absolute Value, Zero Crossing, Slope Sign Changes and Waveform Length. As all the features are in the time-domain, this feature set is often referred to (and will be in this work) as TD. TD is arguably the most commonly employed feature set [28] and serves as a baseline when comparing different handcrafted feature sets.

7.A.2 NinaPro feature set [7, 47]

The NinaPro feature set has been successfully employed on the diverse NinaPro datasets and consist of the concatenation of the TD features alongside Histogram and marginal Discrete Wavelet Transform.

7.A.3 SampEn pipeline [45]

The SampEn pipeline consists of Sample Entropy, Cepstral Coefficients, Root Mean Square and Waveform Length. This feature set was found to be the best combination of features amongst the 50 considered in the original work (brute-force search).

7.A.4 LSF9 [44, 9]

LSF9 is a newly proposed feature set which was originally developed specifically for low sampling rate recording devices (200Hz). Nevertheless, this feature set also offers exceptional performance on higher sampling rate datasets. LSF9 consists of: L-scale, Maximum Fractal Length, Mean Value of the Square Root, Willison Amplitude, Zero Crossing, Root Mean Square, Integrated Absolute Value, Difference Absolute Standard Deviation Value and Variance.

7.A.5 TDPSD [3, 43]

TDPSD proposes to consider the EMG signal alongside their nonlinear cepstral representation. Then, one vector per representation is created by computing the: Root squared zero, second and fourth moments as well as Sparseness, Irregularity Factor and the Waveform Length Ratio. The final vector used for classification is obtained from the cosine similarity of the two previous

Table 7.6 – Comparison between the ConvNet employed in this work and Handcrafted feature sets

	ConvNet	TD	NinaPro	SampEn Pipeline	LSF9	TDPSD	TSD
7 Gestures	93.13%	89.18%	89.48%	91.03%	94.45%	92.67%	95.01%
STD	6.44%	8.34%	7.87%	7.48%	5.89%	6.26%	5.47%
Friedman Rank	3.50	5.86	5.61	4.42	2.51	3.98	2.13
H0	N\A	0 (<0.00001)	0 (<0.00001)	0 (0.04023)	0 (0.03578)	1	0 (0.00196)
11 Gestures	85.42%	81.11%	81.32%	83.57%	87.94%	84.86%	91.03%
STD	9.69%	9.97%	9.80%	9.71%	9.26%	9.61%	8.73%
Friedman Rank	4.07	5.70	5.52	4.41	2.69	4.01	1.61
H0	N\A	0 (0.00017)	0 (0.00095)	1	0 (0.00147)	1	0 (<0.00001)

vectors. The interested reader is encouraged to consult [3] for a detailed description of this feature set.

7.A.6 TSD [33]

TSD represents the evolution of TDPSD. The idea of leveraging the cosine similarity between two vectors of the same features computed from different representation of the signal remain. However, the features have been updated and now consist of: the Root squared zero, second and fourth moments as well as the Sparseness, Irregularity Factor, Coefficient of Variation and the Teager-Kaiser energy operator. Most importantly, this feature set not only considers the similarities between the signal of a particular channel and its nonlinear transformation but also considers these similarities across channels. The interested reader is encouraged to consult [33] for a detailed description of this feature set.

7.A.7 Dataset and Classifier

A standard Linear Discriminant Analysis [28] is selected for classification as it is widely employed in the field and is a computationally and time efficient classification technique both at training and prediction time, while still achieving high classification accuracy [28, 15, 45].

The Long-term 3DC Dataset is employed for comparison. For each Training Recording of each participant (20 participants \times 3 sessions). The first two cycles are employed for training, while the last cycle is reserved for testing (total of 60 train/test per method). The comparison is done for both the seven and eleven gestures considered in this work. The ConvNet’s architecture and hyperparameters are exactly as described in Section 7.6.3. The LDA implementation is from scikit-learn [42] with its defaults parameters.

7.A.8 Comparison of results

Table 7.6 presents the comparison between the ConvNet and the six feature sets.

When testing on the Evaluation Recording, the ConvNet obtained an average accuracy of $49.84\% \pm 10.93\%$, while TD obtained $48.90\% \pm 10.80\%$, TDPSD obtained $50.55\% \pm 10.89\%$ and TSD obtained $56.50\% \pm 11.27\%$. The comparison shows that despite the simplicity of the ConvNet used in this work, it performs almost identically to TDPSD on average and similarly to the five other feature sets considered.

7.B Pseudo-labeling Heuristic

Algorithm 2 Pseudo-labeling Heuristic

```
1: procedure GENERATEPSEUDOLABELS(unstable_len, threshold_stable, max_len_unstable,  
   max_look_back, threshold_derivative)  
2:   pseudo_labels  $\leftarrow$  empty array  
3:   arr_preds  $\leftarrow$  network's predictions  
4:   arr_net_out  $\leftarrow$  network's softmax output  
5:   begin_arr  $\leftarrow$  The unstable_len first elements of arr_net_out  
6:   stable  $\leftarrow$  TRUE arr_unstable_output gets empty array  
7:   current_class  $\leftarrow$  The label associated with the output neuron with the highest median  
   value in begin_arr  
8:   for i from 0..arr_preds length do  
9:     if current_class different than arr_preds[i] AND stable TRUE then  
10:      stable  $\leftarrow$  FALSE  
11:      first_index_unstable  $\leftarrow$  i  
12:      arr_unstable_output  $\leftarrow$  empty array  
13:      if stable is FALSE then  
14:        APPEND arr_net_out to arr_unstable_output  
15:        if length of arr_unstable_output is greater than unstable_len then  
16:          REMOVE the oldest element of arr_unstable_output  
17:        if length of arr_unstable_output is greater or equal to unstable_len then  
18:          arr_median  $\leftarrow$  The median value in arr_unstable_output for each gesture  
19:          arr_percentage_medians  $\leftarrow$  arr_median / the sum of arr_median  
20:          gesture_found  $\leftarrow$  The label associated with the gesture with the highest  
   median percentage from arr_percentage_medians  
21:          if arr_percentage_medians[gesture_found] greater than threshold_stable  
   then  
22:            stable  $\leftarrow$  TRUE  
23:            if current_class is gesture_found AND The time within instability is  
   less than max_len_unstable then  
24:              Add the predictions which occurred during the unstable time to  
   pseudo_labels with the gesture_found  
25:            else if current_class is different than gesture_found AND The time  
   within instability is less than max_len_unstable then  
26:              index_start_change  $\leftarrow$  GetIndexStartChange(arr_net_out,  
   first_index_unstable, max_look_back)  
27:              Add the predictions which occurred during the unstable time to  
   pseudo_labels with the gesture_found label  
28:              Re-label the predictions from pseudo_labels starting at in-  
   dex_start_change with the gesture_found label  
29:              current_class  $\leftarrow$  gesture_found  
30:              arr_unstable_output  $\leftarrow$  empty array  
31:            else  
32:              Add current prediction to pseudo_labels with the current_class label  
   return pseudo_labels
```

Algorithm 3 Find index start of transition heuristic

```
1: procedure      GETINDEXSTARTCHANGE(arr_net_out,      first_index_unstable,  
   max_look_back, threshold_derivative)  
2:   data_uncertain  $\leftarrow$  Populate the array with the elements from arr_net_out starting  
   from the first_index_unstable-max_look_back index to the first_index_unstable index  
3:   discrete_entropy_derivative  $\leftarrow$  Calculate the entropy for each element of data_uncertain  
   and then create an array with their derivatives.  
4:   index_transition_start  $\leftarrow$  0  
5:   for i from 0..data_uncertain length do  
6:     if discrete_entropy_derivative[i] greater than threshold_derivative then  
7:       index_transition_start  $\leftarrow$  i  
8:     Get out of the loop  
return first_index_unstable + index_transition_start
```

Conclusion

The goal of this thesis was eloquently summarized by one of the participants of an early experiment: "It's like I'm a Jedi and I can control the [robotic] arm with the force!". Within this thesis, research relating directly to myoelectric control progressed from a single-user-single-day setting using deep learning, to a new transfer learning algorithm to leverage multi-user aggregated signals and finally to propose a new domain adversarial self-calibration algorithm for long-term classification. During research on these topic, the need for a new sEMG armband arose, as the only real consumer grade armband available ceased production. This thesis also proposed the first topological analysis of deep features within the context of sEMG-based gesture recognition as to better understand what type of information was encoded within the network. One last contribution, specifically stemming from this thesis, is the virtual reality environment which enabled us to study the four main dynamic factors affecting real-time classification in depth. In particular, and in conjunction with the leap motion, the VR experiment allowed us to track the participant limb's position precisely and in 3D, showing the extent of the classifier's deterioration from external forearm rotation for the first time.

Thesis by article, of which this document is an example, seems to be somewhat controversial within certain groups in academia. This view is held to such extent that some researchers do not believe that this type of thesis can rise to the level of quality as can be displayed by a traditional one. While I do not adhere to this idea (as demonstrated by the existence of this thesis), it is important to acknowledge the differences between traditional thesis and thesis by articles; namely the link between the chapters and the inherent restrictions imposed when writing scientific papers. The articles included within this thesis were written to be coherent within themselves which by default generates information-overlap between the different articles. At the same time, conference venues and journals are aimed at a specific readership, which mean that the type and depth of the information conveyed within the paper has to cater to that particular group. Requirements in relation to the maximum number of pages for an article also has a large impact on how the article is articulated. Therefore, the degree of detail given for different aspects of a scientific paper is indubitably tainted by these important considerations. Finally, the structure of this type of thesis makes it hard to have a meta-chapter discussion of the role and impact of the different contributions within the field of interest. The conclusion section, however, offers an opportunity to address this last concern.

8.3 Discussion

At the time of writing these lines, the field of sEMG-based gesture recognition is operating in two different realities. On the surface, offline classification shows exceptional performances with near perfect accuracies over a wide range of recognized gestures simultaneously. More importantly, these performances can be obtained using a wide variety of features (both learned or handcrafted) and a wide variety of classifiers. Thus, it is my opinion that when the four main dynamic factors (i.e. gesture intensity, limb position, electrode shift and transient changes in the signal) are omitted, the problem of classifying sEMG signals is more or less solved. As such, the research priority should be oriented towards long term, real-time gesture classification in activity-based evaluation. This however is an unattainable standard in research due to the vast increase in time and resources required for that type of experiment compared to offline recording. The main reason for this resource difference stems from the fact that each new method would require a completely new dataset to be recorded. In other words, any comparison between different methods requires one new dataset per tested method (using the tested method as the controller to interface with the participant). This also means that these datasets are "burned" once the performance evaluation has been calculated and thus they cannot be re-used by the research community to test new methods. This problem was the main motivation behind the "Virtual Reality to Study the Gap Between Offline and Real-Time EMG-based Gesture Recognition". That is, to propose a type of dataset in-between offline and online, where the four main dynamic factors are included and the users are interacting with a controller not driven by EMG-based signals. This *dynamic dataset* can then be re-used by the research community as a benchmark to compare multiple methods (similarly to an offline dataset), but with data that will be closer to an online dataset than what can be offered with an offline dataset.

8.4 Scientific Contributions Overview

During the completion of my graduate studies, I made several contributions through the publication of peer-reviewed scientific papers, some of which were not included in this thesis as they were only tangential to the main objective of my Ph.D. research. Therefore, I would like to take the opportunity offered by this conclusion to give a brief overview of the other contributions which were not included within the main text corpus of this document. It is of course important to note that none of these contributions I made alone, and that without the co-authors who worked on the papers mentioned within this section, these contributions would not have been possible.

My first foray into research was to explore how genetic algorithms could be used as a way to optimize Question-Answering Systems by automatically selecting which module to use and in which order. This research cumulated in my first conference paper published in *Flairs* [1].

The following year, while I was working on controlling a robotic arm using sEMG data [2], the non-intuitiveness associated with the orientation control of the robot's end-effector became painfully obvious. In fact, most of the errors that the participants made while controlling the robotic arm, came not from the misclassification of the sEMG-based controller, but the highly unintuitive behaviors of the robotic arm's end-effector. As it turned out, at the time and to the best of my current knowledge, there was no intuitive alternative. This problematic led to the publication of a new intuitive adaptive orientation control for robotic arms aimed specifically for people living with upper limb disabilities. This algorithm was published at the *International Conference on Rehabilitation Robotics* [8]. The work was later substantially extended through a vast new number of validation experiments into a journal paper published in *IEEE Transactions on robotics* [3]. Note that, I was a first co-author in the conference paper and second author in the journal paper.

In parallel to the work with this adaptive orientation algorithm and the EMG gesture recognition research for my Ph.D., I also worked on recognizing the mood (e.g. Happy, Sad, Disgust) expressed by professional dancers during an improvised performance from sEMG and orientation (through inertial measurement units) data. A first use case study with a professional dancer was published in *IEEE International Symposium on Robot and Human Interactive Communication* [6]. This use-case study was then extended to include 27 professional dancers interacting in real-time (one at a time) with a robotic swarm to create an improvised, artistic performance between a human and machines, through the use of this mood classification algorithm. This work was published in *ACM Transactions on Human-Robot Interaction* [7]. Note that, while I was the first author in the conference paper, I was the second author in the journal paper.

Finally, I also want to highlight the fact that the work that lead to the journal paper *Deep Learning for Electromyographic Hand Gesture Signal Classification Using Transfer Learning* [5] presented in this thesis, first started as a conference paper. While the journal paper improved upon the transfer learning algorithm and provided vastly more experimental results, it is notable that the original conference paper [4] published in *IEEE International Conference on Systems, Man, and Cybernetics* won the best paper award of the conference.

8.5 Future works

The next step in my work in relation to this thesis is to implement SCADANN in real-time to see the effect of real-time adaptation on a classifier's usability. Furthermore, as was briefly touched on in the "Unsupervised Domain Adversarial Self-Calibration for Electromyographic-based Gesture Recognition" paper, the restriction in term of data accumulation within a window is in part due to the absence of gesture transitions available within the training dataset. Self-learning classifiers used in real-time would thus be exposed to multiple examples of gesture transitions

which can then be added to a re-calibration set, allowing the classifier to recognize a change in the current gesture more rapidly. In effect, this would allow for the use of wider window size without compromising the responsivity of the system. Future work will thus focus on implementing this self-learning classifier as to improve classification performance while also trying to lower the overall latency of the system.

Bibliography

- [1] Ulysse Côté Allard, Richard Khoury, Luc Lamontagne, Jonathan Bergeron, François Laviolette, and Alexandre Bergeron-Guyard. Optimizing question-answering systems using genetic algorithms. In *The Twenty-Eighth International Flairs Conference*, 2015.
- [2] Ulysse Côté Allard, François Nougrou, Cheikh Latyr Fall, Philippe Giguère, Clément Gosselin, François Laviolette, and Benoit Gosselin. A convolutional neural network for robotic arm guidance using semg based frequency-features. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2464–2470. IEEE, 2016.
- [3] Alexandre Campeau-Lecours, Ulysse Côté-Allard, Dinh-Son Vu, François Routhier, Benoit Gosselin, and Clément Gosselin. Intuitive adaptive orientation control for enhanced human–robot interaction. *IEEE Transactions on Robotics*, 35(2):509–520, 2018.
- [4] Ulysse Cote-Allard, Cheikh Latyr Fall, Alexandre Campeau-Lecours, Clément Gosselin, François Laviolette, and Benoit Gosselin. Transfer learning for semg hand gestures recognition using convolutional neural networks. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1663–1668. IEEE, 2017.
- [5] Ulysse Côté-Allard, Cheikh Latyr Fall, Alexandre Drouin, Alexandre Campeau-Lecours, Clément Gosselin, Kyrre Glette, François Laviolette, and Benoit Gosselin. Deep learning for electromyographic hand gesture signal classification using transfer learning. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(4):760–771, 2019.
- [6] Ulysse Côté-Allard, David St-Onge, Philippe Giguère, François Laviolette, and Benoit Gosselin. Towards the use of consumer-grade electromyographic armbands for interactive, artistic robotics performances. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 1030–1036. IEEE, 2017.
- [7] David St-Onge, Ulysse Côté-Allard, Kyrre Glette, Benoit Gosselin, and Giovanni Beltrame. Engaging with robotic swarms: Commands from expressive motion. *ACM Transactions on Human-Robot Interaction (THRI)*, 8(2):1–26, 2019.
- [8] Dinh-Son Vu, Ulysse Côté Allard, Clément Gosselin, François Routhier, Benoit Gosselin, and Alexandre Campeau-Lecours. Intuitive adaptive orientation control of assistive robots for

people living with upper limb disabilities. In *2017 International Conference on Rehabilitation Robotics (ICORR)*, pages 795–800. IEEE, 2017.