SUMMARY

Gesture recognition is becoming an increasingly popular classification problem, with the rise in demand for reliable gesture recognition in various fields. Different technologies can be used for collecting data for said classification problem, most common among them is surface electromyography and force myography. In our pre-thesis, we explored the potential of using deep neural networks for force myography hand gesture recognition, and found it to perform comparably to traditional classification algorithms, such as linear discriminant analysis and support vector machines.

A common issue with neural networks and other machine learning approaches is that they are reliant on sufficient data for training, which is often difficult to obtain. As such, in this paper we explore the potential of using transfer learning to obtain a better prediction accuracy, by leveraging multi-source data collection.

In addition, we observe from the literature, on the field of force myography gesture recognition, that a lot of studies use custom hardware and do not publish their data, making it difficult to accurately compare or reproduce results. We address the reproducibility problem by making the dataset publicly available, by using a commercially available device for data collection thus ensuring the reproducibility of our results. Making the dataset publicly available should further allow easier entry into this field of research, by reducing the necessity of gathering new data.

Our benchmark dataset includes data collected from 20 people. For each person we have recorded 5 seconds of data for each of a set of 18 unique gestures repeated 5 times. By recording data at a very high frequency compared to what we have observed from other studies, we hope to accommodate as many applications as possible, as the data can be down sampled to a lower frequency as desired.

We use hyperparameter optimisation to find the best one, two and three layer configurations for our baseline model, and train all of our models using the three respective configurations. Furthermore, we also go into detail of how our data is split during hyperparameter optimisation, training and evaluation of our models. Using the aforementioned models, we explore the performance of the different architectures, showing that while transfer learning has the potential to improve performance, one can easily end up with negative transfer, making further research on how to best apply transfer learning for this task necessary. Additionally we explore how the models performed with respect to individual gestures for some subjects, showing that careful selection of which gestures to use is necessary as some gestures are difficult to distinguish from one another.

Finally, it is our hope that the benchmark dataset we have constructed and made publicly available, may contribute to the advancement of the fields of gesture recognition and transfer learning.

Force Myography Hand Gesture Recognition Using Transfer Learning

Thomas Buhl Andersen
Department of Computer Science, Aalborg University
Aalborg, Denmark
tban09@student.aau.dk

Rógvi Eliasen
Department of Computer Science, Aalborg University
Aalborg, Denmark
relias13@student.aau.dk

Mikkel Jarlund
Department of Computer Science, Aalborg University
Aalborg, Denmark
mjarlu15@student.aau.dk

Bin Yang*
Department of Computer Science, Aalborg University
Aalborg, Denmark
byang@cs.aau.dk

ABSTRACT

Force myography has recently gained increasing attention for hand gesture recognition tasks. In this paper, we attempt to improve gesture recognition accuracy by utilising data from multiple persons using transfer learning. We experiment with both a progressive neural network architecture and a variation, a combined progressive neural network which seeks to learn more generalised features from the source domains. We show that while transfer learning can improve performance over a fully-connected neural network, care needs to be taken lest one end up with negative transfer. In the process, we note a lack of publicly available benchmark data in this field, with most existing studies collecting their own data often with custom hardware and for varying sets of gestures. This limits the effectiveness of such transfer learning approaches as well as the ability to compare various algorithms. We therefore also contribute to the advancement of this field by making accessible a benchmark dataset, collected using a commercially available sensor setup from 20 persons covering 18 unique gestures. We hope that others may utilise our data for similar transfer learning approaches, while also allowing further comparison of results and easier entry into the field of force myography gesture recognition.

1 INTRODUCTION

Gesture recognition is increasingly becoming more popular, as it can be used in various fields, such as rehabilitation in healthcare, smart-homes where gestures can be used as commands, and prosthetic limbs [11, 15, 22]. Different technologies can be used for collecting gesture recognition data, most common among them is Surface Electromyography (sEMG) which measures the electrical signals when activating the muscles, and Force Myography (FMG) which measures the mechanical activity of the muscles, i.e. how a muscle changes shape when it is used [23]. While feature extraction is important for sEMG to extract useful information due to its inherent noisy nature, FMG does not require feature extraction to extract useful information [5, 7].

The most popular classification algorithms using FMG data are Linear Discriminant Analysis (LDA) and Support Vector Machine (SVM), however not much research has been devoted to the use of deep neural networks [23]. In our previous work [2], we explored the potential of deep neural networks for FMG hand gesture recognition by comparing it with LDA and SVM. We showed that there was potential in utilising deep neural networks for FMG hand gesture recognition.

Recent studies in this field often collect their own data, e.g., different sets of gestures, using their own custom hardware [3, 9, 13, 14]. This creates a reproducibility problem where the custom hardware settings cannot be exactly configured by other researchers and the results obtained on different setups cannot be effectively compared to each other. We argue that using a commercially available product eases the hardware configuration issues and publishing a FMG benchmark dataset collected using the product makes the comparison of different results easier. In addition, having a publicly available dataset allows for easier entry into the field of FMG hand gesture recognition, as it will not require everyone entering the field to first collect their own data.

Transfer learning is a relevant topic for hand gesture recognition, as it allows the re-use of previously collected data. The idea of transfer learning is that previous learned knowledge can be transferred to new domains, to speed up the learning process and or increase performance. However in order to transfer knowledge from a source domain, it is important to know *what to transfer*, *how to transfer* and *when to transfer*. As such, there are three main categories of transfer learning, namely *Inductive*, *Transductive* and *Unsupervised*. The different categories can provide information into *what to transfer*, as *Inductive* transfer learning requires the tasks to be different for both source domain and target domain. *Transductive* transfer learning requires the task to be the same for both source domain and target domain, and *Unsupervised* transfer learning is similar to *Inductive* transfer learning but the focus is on unsupervised learning in the target domain [17].

In this paper we will use deep learning models with *Transductive* transfer learning. Specifically, we will use domain adaptation transfer learning using Progressive Neural Networks (PNN) and compare the results to a baseline model architecture that does not utilise transfer learning. In addition, the collected data is made publicly available[1] for the sake of reproducibility and such that those entering the field of FMG hand gesture recognition can use this data in their research. Henceforth we will refer to the people that data has been collected for as subjects.

*Supervisor

[1] https://github.com/exoskelebox/force-myography-hand-gesture-recognition-benchmark-data

2 RELATED WORK

In our previous work [2] we looked into approaches used for hand gesture recognition in the literature. We implemented several deep learning algorithms to evaluate the potential of introducing deep learning to FMG hand gesture recognition. Furthermore we also assessed the potential of using transfer learning for FMG hand gesture recognition, as has been done with sEMG [8]. [8] has shown promising results by utilising transfer learning, as transfer learning can minimise the amount of samples required to effectively identify gestures thereby reducing the training time, and increase the overall accuracy for a given subject [16, 21]. Transfer learning for sEMG hand gesture recognition has been done using an variation of PNN [8]. PNN's architecture starts with single model, or a column, for a given task. When it switches to a new task the parameters of the previous model is frozen and a new column is instantiated. The newly instantiated column also receives input from the frozen model via lateral connections, such that it can learn from previous learned features. The lateral connections are layer-by-layer connected with the newly created column [18]. An issue with PNN, however, is that with a large sequence of tasks, the complexity quickly becomes problematic as the number of columns and lateral connections grow. [8] thus used a variation of PNN, where rather than training a column for each subject in their pre-training dataset, they trained a single column on all the subjects of the pre-training dataset and then included this single pre-trained column for the final model according to the PNN method.

It is beneficial for transfer learning to have a large set of source domain data to draw from, and it is easier to transfer knowledge if the data distribution is similar to the target domain. It would therefore be helpful if similar datasets were publicly available. We review four independent research papers that utilise FMG data to uncover their data collection process. In the first paper we examined [3], data was collected using a custom armband with 8 sensors wrapped around the upper forearm. Data was collected for 10 subjects each performing 6 gestures with a sampling frequency of 10 Hz. In the second paper, data was collected using a custom wristband with 15 sensors wrapped around the wrist. Data was collected for 10 subjects each performing 6 gestures 12 times with a sampling frequency of 30 Hz [9]. In the third paper, data was collected using a custom armband with 16 sensors wrapped around the upper forearm. Data was collected for 12 subjects each performing 48 gestures 5 times with a sampling frequency of 10 Hz [13]. In the fourth paper, data was collected using a custom armband with 16 sensors on the dorsal side and 16 sensors on the volar side of the forearm. Data was collected for 6 subjects each performing 17 gestures 4 times with a sampling frequency of 100 Hz [14]. In addition, none of these studies have made their data publicly available.

In summary, we observe that there is a lack of a publicly available FMG datasets, which precludes using them for transfer learning. Furthermore, their collection process varies widely in method and execution, making it hard to perform any kind of meaningful method comparisons to assess the performances of different methods and to identify the state-of-the-art methods.

3 PROBLEM DEFINITION

Transfer learning uses domains D and tasks T, where in D we have a source domain D_s, and a target domain D_t for which we want to transfer knowledge to. A domain consists of a feature space X and a marginal probability distribution $P(X)$, where $X = \{x_1, ..., x_n\}$ and a task $T = \{Y, f(\cdot)\}$ where $f(\cdot)$ is a predictive function and Y the label space [17]. There are various approaches to transfer learning depending on how knowledge can be transferred effectively from source domain to target domain [17].

We utilise domain adaptive transfer learning, where we assume that D_s and D_t are different but T_s and T_t are the same [17], which is appropriate for our problem, as we have different domains (i.e. different subjects) but the gesture recognition task for all domains is to recognise the gestures shown in Figure 1.

4 DATA COLLECTION

We collect data from a total of 20 subjects, and describe the collected data for each subject as well as the protocol we follow during our data collection process. For each subject we collect contextual information, fit the sensors, perform calibration and collect sensor readings.

4.1 Equipment

For this study we use a setup of 2 BIOX Armbands[2], one with 7 sensors for the wrist and a larger one with 8 sensors for the forearm. We connect both sensors to our laptop, and during data collection we gather sensor readings from both sensors with a frequency of 975 Hz-1000 Hz, limited by our laptops processing speed. We sample at the highest frequency possible, to avoid limiting the potential applications, as the data can be down sample to a lower frequency as appropriate.

4.2 Consent

All data is gathered from volunteers. Before any data is collected, subjects are presented with a disclaimer informing them of what data we will collect and how we intend to use it, including informing them of our intent to make the data publicly available:

> We are a research group at Aalborg University attempting to break new ground in the field of gesture recognition, and we need your data to do so. We will collect data such as your age, gender, fitness as well as record your arm / wrist while you do different gestures. That is to say, we will not collect any personally identifiable information (PII) such as your name, address etc. By participating in our data collection, you agree to have your data shared in a public dataset. We share the data such that our results may be reproduced and improved upon in the future.

4.3 Subject Information

The subjects are presented with a form and asked to provide some contextual information that we expect to have some impact on the gesture identification process. As part of this, we measure the circumference of wrist and arm at the locations we will apply the

[2] https://www.bioxgroup.dk/product/biox-armband/

(a) Neutral closed | (b) Neutral rest | (c) Neutral open | (d) Neutral wide | (e) Neutral flexion | (f) Neutral extension

(g) Supine closed | (h) Supine rest | (i) Supine open | (j) Supine wide | (k) Supine flexion | (l) Supine extension

(m) Prone closed | (n) Prone rest | (o) Prone open | (p) Prone wide | (q) Prone flexion | (r) Prone extension

Figure 1: Gestures.

sensor armbands. For the armband with 7 sensors, this is ~ 5 cm below the wrist, while the 8 sensor armband is below the elbow at the maximal bulge of the forearm. The contextual information is described below and the distribution of the data is noted in Table 1.

- **Age** Research has shown that ageing reduces joint mobility [12] which we expect to impact subjects performance of the requested gestures. Additionally age has effects on the mechanical properties of soft tissue (e.g. elasticity, density, thickness) [19] which we also expect to affect gesture identification performance.
- **Gender** Research shows that there are physiological differences [4] including variation in wrist joint mobility [6, 20] between genders. As such, we expect that the differences will affect gesture identification.
- **Handedness** As the joint mobility might be better in the dominant hand.
- **Weekly Exercises** As what we measure is the mechanical activation of the muscles we expect that physical fitness will impact the readings. We have therefore asked each subject to give the number of days per week when they exercise for at least half an hour.
- **Injury** If a subject has an injury or condition that affects the mobility of the wrist or forearm this will likely impact the gesture identification. As such, we have asked the subjects whether they have an injury that affects their wrist or hand mobility.
- **Wrist and Forearm circumference** We expect that the wrist and forearm circumference will affect gesture identification as it will impact how well the sensor armbands fit the subject.

Gender	17 Male, 3 Female
Age	23.85(±1.53)
Handedness	18 Right, 2 Left
Weekly Exercise	2.35(±1.79)
Injury	1 Yes, 19 No
Wrist circumference	17.86(±1.07)
Forearm circumference	26.9(±1.66)

Table 1: Subject Distribution. For numeric properties, population mean and standard deviation is given.

4.4 Sensor Fitting

The armband is laid on a flat surface with the cables to the left and we ask the subjects to lay their right forearm on the middle of the armband with the wrist prone, such that the armband match the measured location. Once the subjects have positioned the arm correctly we close the armbands.

The subjects are instructed to sit such that there is ∼ 1 m free space in front of them and ∼ 0.5 m to their sides and back. We then ask them to position their arm such that the upper arm is parallel to the body, the elbow does not touch their side, and their forearm is perpendicular to the upper arm. We then inform them to hold their elbow as still as possible and keep their forearm horizontal and straight in front of them.

When the subjects are positioned correctly, we give them a remote controller and instruct them to follow the prompts on the screen and to click the remote when they have assumed the displayed gesture. They must then hold the position shown until the data collection is done and a new gesture is shown. During the calibration and collection processes, we will keep watch to ensure the correct execution of the gestures. Should we see any errors we will intervene and ask them to redo the gesture and give directions for correct execution.

4.5 Calibration

Calibration is performed because the sensors will likely only utilise part of the possible output range. We thus perform calibration in order to better utilise the full output range of the sensors. In order to capture the upper limit of the subjects muscle activation, the subjects are told that they should try to exert their muscles as much as possible when the sensors were being calibrated. Since the two sensor armbands are activated differently for each gesture, they are calibrated separately using the two gestures that we found to best activate the sensors of the respective armband. These gestures were Figure 1a and Figure 1r for the arm and wrist, respectively.

A side-effect of calibration is that the resting value[3] of the sensors is also increased, depending on the number of calibration steps (i.e., how much the sensitivity is increased). Since the armbands are calibrated separately they will likely require a different number of calibration steps, leading to different resting values. For this reason we record the number of calibration iterations for both armbands for each subject as well as the final sensor values on calibration in the belief that this could be used to account for these factors.

4.6 Data Collection

The subjects are instructed that they do not need to exert maximum force during the data collection, and that they should proceed at their own pace, resting between gestures as necessary. We collect data for a set of 6 hand gestures (closed, rest, straight, wide, flexion and extension) with 3 different wrist positions (neutral, prone and supine), resulting in a total of 18 different gestures which can be seen in Figure 1. We collect data for each gesture 5 times, each time collecting 5 seconds of sensor readings at a frequency of 975 Hz-1000 Hz.

In addition to the value of each sensor we record a label signifying which gesture was performed, as well as which subject performed it. Each reading has a timestamp, though only the time difference in-between sensor readings of the same subject is valid, not the time of day due to the implementation of the timer. As we collected data over several repetitions of the gestures we have also included a numeric indicator of which repetition the reading is from.

The collected data is of the form $\langle arm\_sensors, wrist\_sensors, timestamp, repetition, subjectID, gestureID \rangle$ where $arm\_sensors$ is a 8 dimensional vector and $wrist\_sensors$ is a 7 dimensional vector of sensor readings.

[3]The resting value is the value the sensors outputs when no pressure is exerted on them.

4.7 Risks and Assumptions

There are a couple of areas of potential risk with our data collection protocol.

With respect to the contextual information, we rely on the subjects to provide the information which may have inaccuracies, especially on fields such as frequency of exercise where subjects may embellish their details.

Furthermore, there is some degree of inaccuracy in regards to the wrist and arm measurements. Though we try to measure as consistently as possible, when dealing with something as inherently soft as an arm it is difficult to manually measure at exactly the same tightness each time.

The same applies to the fitting, which while we strive for consistency, likely exhibits some degree of variance, which could lead to difference in the needed calibration and subsequent data collection. As described in subsection 4.5, we include the calibration information in order to alleviate this issue.

In regards to the gesture data collection, we choose to let the subjects determine their own pace, signalling when they have assumed the next gesture. While we supervise the data collection, and ask the subjects to redo any gestures where we observe errors, it is possible that some subjects may push the button a bit too early.

Additionally since we leave it up to the subjects to decide when and how long they need to rest between gestures, there may be some variance in their fatigue levels throughout the data collection.

5 FEATURE ENGINEERING

We use domain knowledge to engineer the features in our dataset.

5.1 Subject Information

As we mention in section 4, we gather contextual information about every subject as we believe it can aid in hand gesture recognition. The idea behind wanting to have these extra features, for example, is to evaluate if a model is able to better classify the gestures, if the model knows the age of the subject. Unfortunately, due to COVID-19 [1], we are unable to collect data from a significantly diverse demographic and have thus decided not to utilise the contextual information.

5.2 Feature Scaling

Based on findings in our previous work [2], we implement scaling. We scale the dataset to take into account how the calibration functions, so as to normalise the data while considering the *resting* values of the sensors.

Let D be the raw dataset collected for a given subject, consisting of 15 dimensional vectors containing a sensor reading from each of

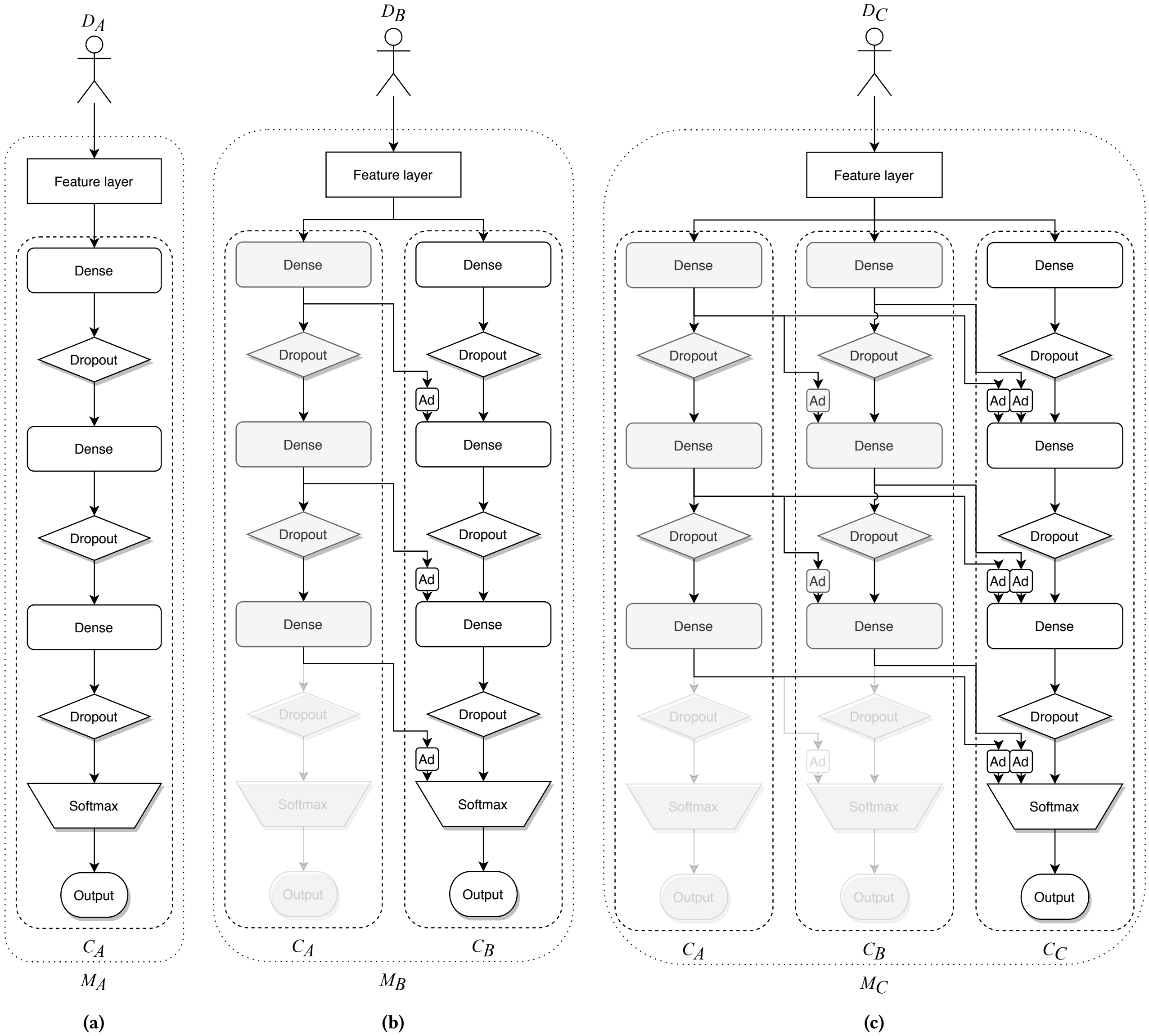


Figure 2: PNN Overview. Given 3 subjects with domains D_A, D_B and D_C, assume $D_A,\ D_B \in D_S$ belong the source domain and $D_C = D_T$ is the target domain. First in a model M_A containing a single 'column' C_A is trained for D_A. Then that models weights are frozen and a new model M_B is created in b with 2 columns, C_A and a new C_B, with lateral connections from the C_A to C_B through adaptors Ad. This model is trained on D_B and the process is then repeated with the weights being frozen and a new model M_C being created in c with an new column C_C and lateral connections from both columns C_A and C_B. This model is then trained on the target domain D_C. The model thus increases in size with the number of source domains.

the sensors. Let d^i be the set of data readings from sensor i, and d_j be the jth sample of D. The jth reading of sensor i is thus d^i_j. For each $d^i_j \in D$ we subtract the minimum reading for the corresponding sensor, $min(d^i)$, and divide by the overall maximum reading across every sensor $max(D) = max(\ \{max(d^i)|i \in \{1, ..., 15\}\}\ )$ which is also adjusted by $-min(d^i)$.

$$Scale(\ d^i_j) \ = \ \frac{d^i_j - min(d^i)}{max(D) - min(d^i)} \tag{1}$$

We consider the local minimum of each sensor, as the resting values may be different due to calibration (see section 4). We are also interested in preserving the relative values between the sensors, we therefore consider the global maximum of the sensors, as the amount of force exerted vary when performing different gestures.

6 MODELS

In our previous work [2] we have shown that simple deep learning can perform as well as the most commonly used models SVM and LDA. Therefore, we experiment with different deep learning models to evaluate the potential of applying transfer learning to this task.

6.1 Fully-Connected Neural Network

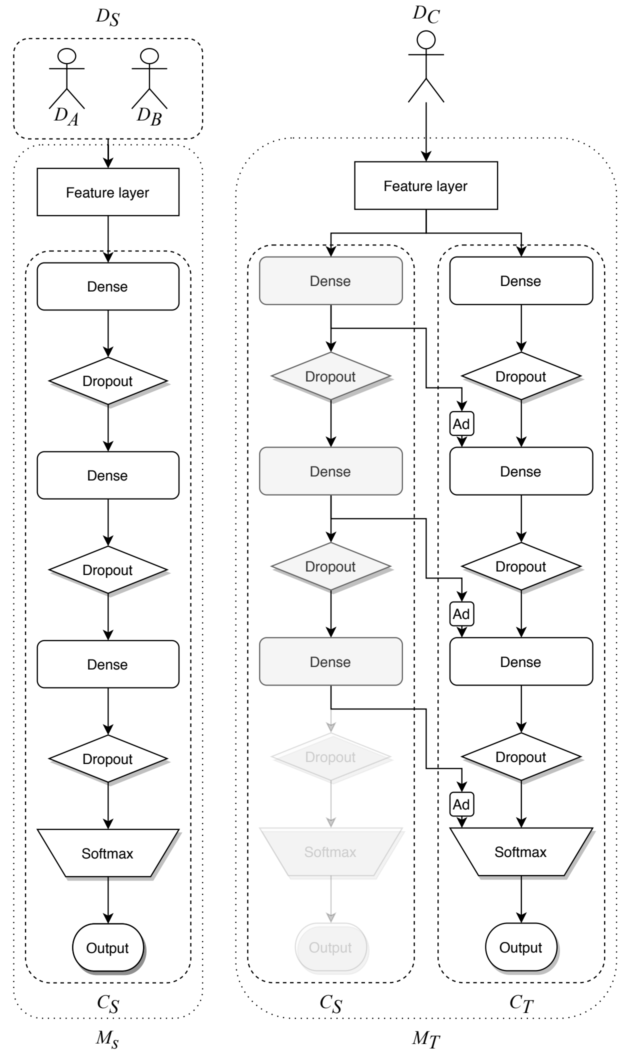
Our non-transfer learning baseline consists of a series of Fully-Connected Neural Network (FCNN) layers, using the ReLU activation function, followed by a Softmax layer. In order to avoid overfitting we apply dropout between each layer. This architecture also serves as the basis for the columns of our transfer learning approaches. As we consider a non knowledge transfer setup, we train a baseline model for each subject.

6.2 Progressive Neural Network

PNN was proposed by [18] as a way of applying transfer learning to a sequence of tasks while avoiding catastrophic forgetting. Figure 2 shows the training process of a PNN model with 3 domains, 2 source domains and 1 target domain. It works by preserving models trained on the source domains as columns, 2a+2b, which are then frozen and connected to the target column for the target task through lateral connections in Figure 2c. These lateral connections go through adapters that serve to provide dimensionality reduction on the source inputs, and aid in learning how best to draw on the pre-trained columns. Dimensionality reduction is performed to reduce the total number of parameters as the number of columns increases [18]. The reason for freezing all the source columns, is that during back propagation we only want to adjust the weights for the target column, as the source columns already have been trained and are only used to transfer knowledge. Because no changes are being made when the weights are frozen for the source columns, we prevent catastrophic forgetting while the current column can draw on all previously learned features across all layers.

6.3 Combined Progressive Neural Network

As the PNN architecture exhibits quadratic growth in the number of parameters when increasing the number of source domains [8], there is a limit to the number of source domains we can reasonably draw on. Hence, if we want to be able to learn from a large set of source domains we need an alternative to having a column for each. One possible approach, which was proposed by [8], is to combine the source domain datasets and only train a single column on this combined source domain as seen in Figure 3a. Like the PNN, the source column is then connected to the target column with lateral connections through adapters, as seen in Figure 3b. Combining the source domains is possible because, unlike what the original PNN was proposed for [18], we do not have different tasks in addition to the different domains. We can thus combine the source domains and train a single column to learn the general features across the source domains which are helpful for our task. With Combined Progressive Neural Networks (CPNN) it is thus possible to draw on a large number of source domains without increasing the number of parameters of the model.



(a) **(b)**

Figure 3: CPNN Overview. Given 3 subjects with domains D_A, D_B and D_C, assume $D_A, D_B \in D_S$ belong the source domain and $D_C = D_T$ is the target domain. Different from the PNN, all source domains in D_S are combined to train a single column C_S in a. Then, like with PNN, that columns weights are frozen and in b a new column C_T is created with lateral connections from C_S through adapters Ad. This model is then trained on the target domain D_C. As such, unlike with the PNN, the size of the model will not change regardless of the number of source domains.

7 HYPERPARAMETER OPTIMISATION

For Hyperparameter Optimisation (HPO) we utilise Bayesian optimisation in which previous trial scores are exploited to determine what combination of Hyperparameters (HPs) should be explored next to get a better score [10]. In terms of scoring, we perform *leave one group out* cross-validation on the repetitions for all subjects

#	Dense 1	Dense 2	Dense 3	Dropout	Score
1	**256**	-	-	**0.4**	**79.51%**
2	256	-	-	0.5	79.45%
3	128	-	-	0.5	79.32%
4	256	-	-	0.2	79.23%
5	256	-	-	0.1	79.06%
6	**256**	**32**	-	**0.1**	**79.23%**
7	128	128	-	0.2	79.10%
8	64	256	-	0.4	78.93%
9	64	64	-	0.2	78.90%
10	128	128	-	0.3	78.82%
11	**256**	**64**	**32**	**0.1**	**79.27%**
12	64	128	64	0.4	79.16%
13	256	64	32	0.2	79.16%
14	256	64	32	0.4	79.16%
15	128	32	128	0.2	78.87%

Table 2: FCNN HPO Results.

and use the evaluation mean as the score - the repetition used for *holdout testing* we mention in section 8 is not included. To reduce the complexity of the task, we focus solely on the FCNN, which should apply to the PNN and CPNN, seeing as the architecture of the FCNN is identical to that of the columns in the PNN and CPNN with the exception of the adapter layers.

We perform three independent HPO searches for one, two and three hidden layers respectively. Based on preliminary testing, we have determined the range for each HP in the search space. The search space for each hidden layer includes the number of neurons limited to the set $\{32, 64, 128, 256\}$ as well as the dropout rate limited to the set $\{.0, .1, .2, .3, .4, .5\}$. Our reasoning for not including more hidden layers is that the size of the search space exhibits exponential growth as the number of hidden layers n increases, with $4^n \times 6$ possible combinations of HPs, which quickly becomes unfeasible.

We run each HPO search for 10 trials and note the best scoring trials in Table 2. From the results, we observe that the single layered FCNN with 256 neurons and a dropout rate of 0.4 achieves the best score, with the best performing two and three layered FCNN performing $\approx$ .25% worse but within .05% of one another, which suggests that the FCNN favours a shallow architecture.

From this point forth, we will use the HPs from the best performing trial for one, two and three hidden layers respectively for our models and compare the results with one another.

8 TRAINING AND EVALUATION

As mentioned in section 4, we collect data for a series of repetitions, each repetition covering 5 seconds of data for every gesture. Of the 5 total repetitions we collect for each subject, we will use the last repetition for testing and the remaining repetitions for training. We believe this best fits a case where a person may equip the device and collect some initial data to train a model that should then perform predictions on subsequent actions. For each of the models, we split the training data into a training and validation set 75/25% and utilise early stopping monitoring the validation loss delta with a patience of 5 and delta threshold of 0.001 to determine when to stop training.

8.1 FCNN

For the baseline FCNN model, we train a model for each subject using the data for the last repetition as test data and the remainder as training data.

8.2 CPNN

During the training of CPNN we start by combining the data for all subjects except the one we want to train the current model for. This combined dataset includes all 5 repetitions for these subjects. With this combined dataset we pre-train a model as described in section 6. We then take the data for the target subject and separate the last repetition for testing as mentioned above, before training on the remaining repetitions.

8.3 PNN

For the PNN, we train a column for each source subject, using all repetitions gathered for that subject. When all the source columns have been trained, we separate the last repetition of the target subjects data for testing and train the target model with the remainder.

9 RESULTS

Based on the best model configurations in section 7 we fit three FCNN, CPNN and PNN models. The performance of the models can be seen in Table 3 and Figure 4 shows the per subject accuracy for the FCNN, CPNN and PNN models with the highest mean accuracy. Based on the best and worst subject models of each architecture we construct a confusion matrix such that we can observe the classification differences for each model.

9.1 FCNN

Looking at the test results in Table 3, we observe that, contrary to the HPO results seen in Table 2, the deeper models outperform the shallower ones.

Looking at the confusion matrix for our best and worst performing FCNN subject models in Figure 5, we observe some noteable misclassifications. Looking at the confusion matrix for our worst performing FCNN model in Figure 5a, we observe that the gesture *supine closed* is predicted correctly, however the model cannot distinguish between *supine closed* and *supine rest*, both of which it classifies as *supine closed*. Likewise in the confusion matrix for our best performing FCNN model in Figure 5b, we observe that the model cannot distinguish between *supine straight* and *supine wide* as well as *prone rest* and *prone wide*.

9.2 CPNN

As mentioned in subsection 6.3 the columns of our CPNN models are based on the FCNN architecture, and we have trained 3 CPNN models based on our HPO. CPNN accuracy for the first two model configurations, shows a higher mean accuracy than FCNN and PNN, however the mean accuracy for CPNN when using three layers is worse than that of FCNN.

#	Dense 1	Dense 2	Dense 3	Dropout	FCNN		CPNN		PNN	
					μ	σ	μ	σ	μ	σ
1	256	-	-	0.4	77.59%	13.20%	**78.12**%	12.83%	76.65%	13.55%
2	256	32	-	0.1	77.81%	13.06%	**78.28**%	12.88%	75.42%	14.24%
3	256	64	32	0.3	**78.14**%	12.63%	77.11%	14.01%	76.27%	12.11%

Table 3: Evaluation Results. Mean accuracy μ and standard deviation σ. Optimal number of neurons per layer and dropout rate were identified by hyperparameter optimisation, described in section 7.

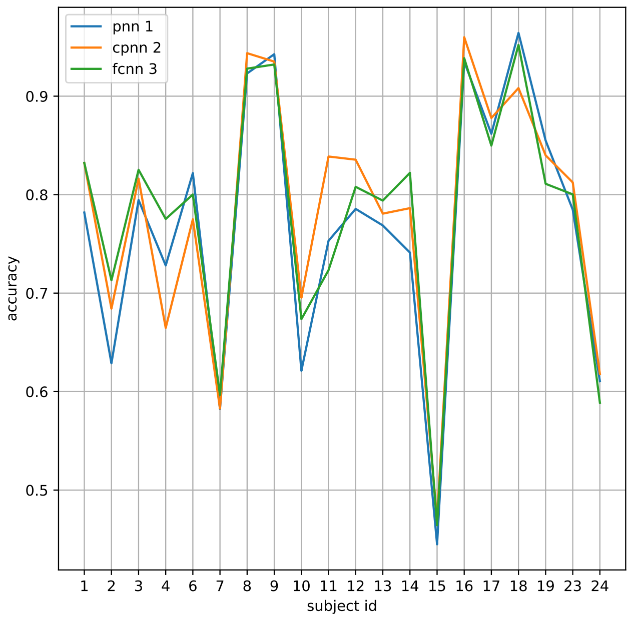


Figure 4: Subject Model Accuracies.

Looking at the confusion matrix for our best and worst performing CPNN subject models in Figure 6 and compare with Figure 5, we observe some interesting differences. The worst performing subject model correctly classifies *prone closed*, but has a harder time classifying gestures such as *prone flexion* and *neutral rest*. The best performing subject model has a hard time classifying *neutral wide* as it is often misclassified as *neutral straight*.

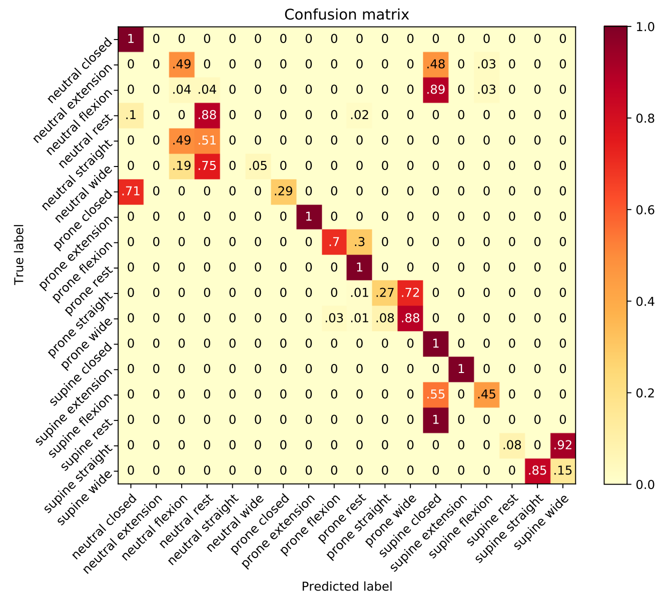
9.3 PNN

We also based our PNN models on the best performing baseline FCNN HP settings. Our PNN models performed worse than the FCNN and CPNN models regardless of the number of layers, as can be seen in Table 3.

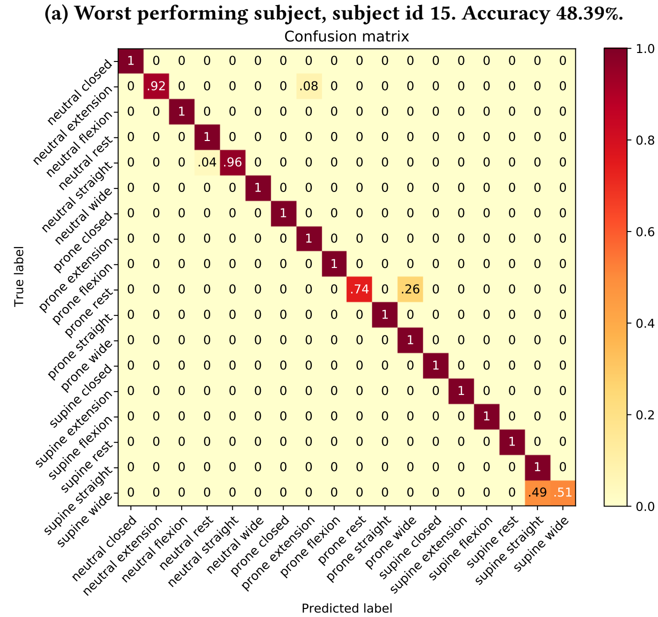
As can be seen in Figure 7, PNN classifies gestures for the best performing subject model better than CPNN and FCNN, however, looking at the worst performing subject model, we can see it has a harder time classifying the gestures than CPNN and FCNN.

10 DISCUSSION

In this section, we discuss our experiments and our benchmark dataset.

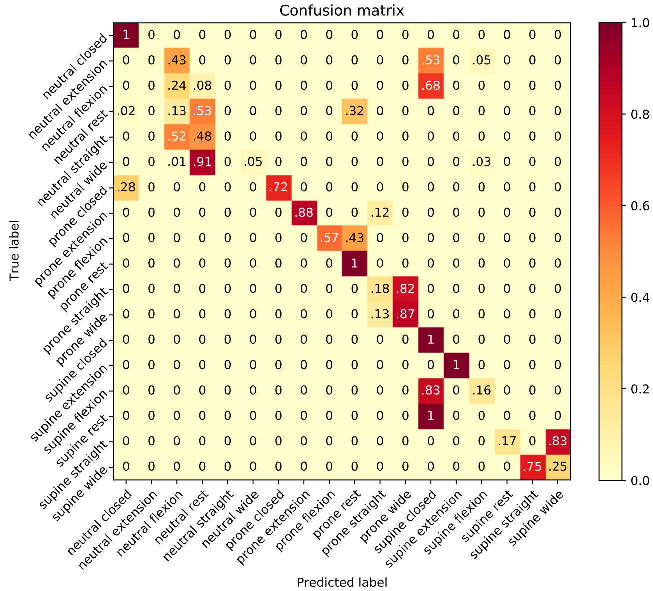


(a) Worst performing subject, subject id 15. Accuracy 48.39%.

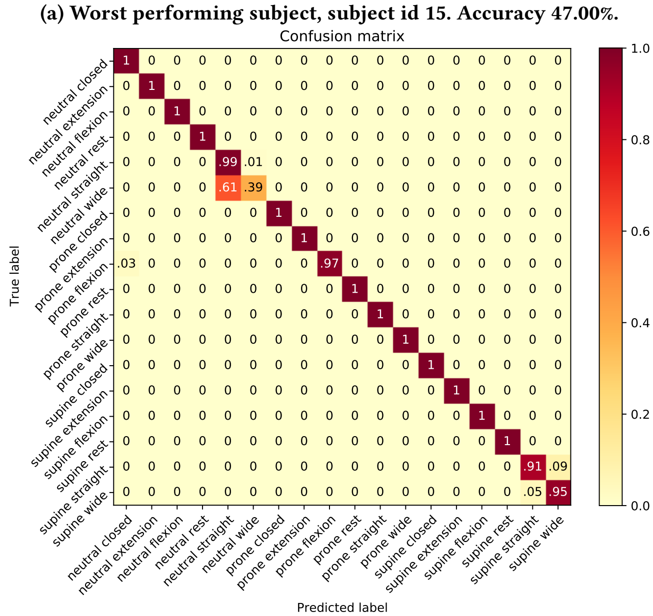


(b) Best performing subject, subject id 18. Accuracy 95.21%

Figure 5: Confusion matrix for best performing baseline FCNN based on our results in Table 3.



(a) Worst performing subject, subject id 15. Accuracy 47.00%.



(b) Best performing subject, subject id 16. Accuracy 95.61%

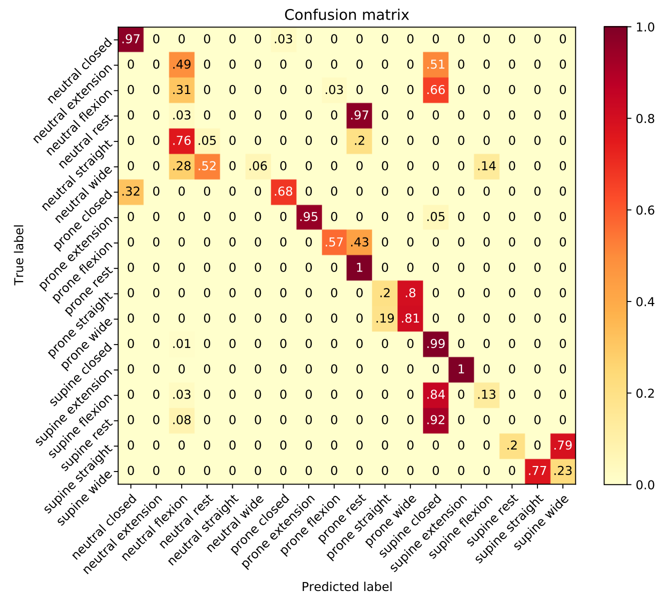
Figure 6: Confusion matrix for best performing CPNN based on our results in Table 3.

10.1 Transfer Learning

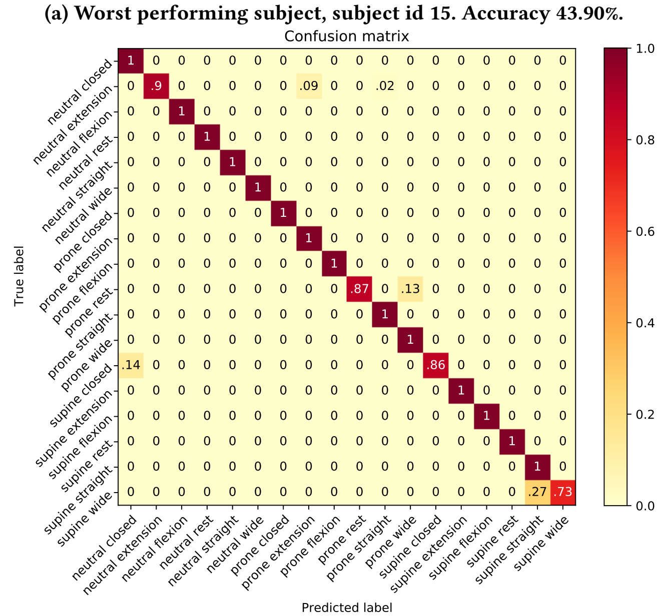
Our best transfer learning model, CPNN, performed slightly better than our baseline FCNN, the PNN model performed worse. This suggests that there is potential for knowledge transfer but that one should be careful when applying transfer learning lest one end up with negative transfer.

10.2 PNN vs CPNN

While CPNN generally performed slightly better than FCNN, our PNN models performed noticeably worse. This suggests that the CPNN architectures single pre-trained column better captures more general features across subjects, which give a better transfer compared to the collection of individually pre-trained columns for PNN which learn subject specific features for the individual source subjects.



(a) Worst performing subject, subject id 15. Accuracy 43.90%.



(b) Best performing subject, subject id 18. Accuracy 96.43%

Figure 7: Confusion matrix for best performing PNN based on our results in Table 3.

10.3 Hyperparameter Optimisation

We have performed HPO on the number of neurons in each layer and the dropout rate to find the best HPs for our task. We did not include activation functions or learning rate in the search, but as we used the Adam optimiser, the learning rate dynamically adjusts during training.

We only performed HPO on the FCNN model, as that also served as the basis for our transfer learning models. It might, however, be that the optimal HPs differ for our transfer learning architectures. As such it might be interesting to perform HPO for these models as well, including optimising the adapters and possibly having different HPs for our source and target models.

10.4 Data Collection

We mention in section 5 that we did not manage to collect data from a sufficiently diverse demographic, and therefore decided not to use the contextual information. However, we still think that it is relevant and should be collected and considered under normal circumstances, as we believe it could help transfer knowledge more effectively. In general, we would have liked to collect more data, as it would have enabled us to explore further issues such as comparing performance between different demographics, or isolate some of the subjects to be used exclusively for the pre-trained models to reduce the training time.

11 CONCLUSION

We have collected a FMG benchmark dataset for hand gesture recognition using a commercially available sensor setup. We have collected benchmark data for 20 subjects, including contextual information about the subject, for a total of 18 unique gestures.

The data is collected at a very high frequency at approximately 1000 Hz, to accommodate as many applications as possible, as the data can be down sampled to a lower frequency as appropriate.

We have used this dataset to show that transfer learning has the potential to increase recognition accuracy by incorporating knowledge learned from other subjects. However, negative transfer may happen. Both the dataset and the source code of the use-case have been made publicly available on GitHub[4]. We believe that this dataset will facilitate research both on FMG based hand gesture recognition and on transfer learning.

GLOSSARY

CPNN Combined Progressive Neural Networks. 8–11

FCNN Fully-Connected Neural Network. 8–12

FMG Force Myography. 3, 4, 12

HP Hyperparameter. 8–10, 12

HPO Hyperparameter Optimisation. 8, 9, 12

LDA Linear Discriminant Analysis. 3, 8

PNN Progressive Neural Networks. 3, 4, 7–11

sEMG Surface Electromyography. 3, 4

SVM Support Vector Machine. 3, 8

[4] https://github.com/exoskelebox/force-myography-hand-gesture-recognition-benchmark-data

REFERENCES

[1] [n.d.]. https://politi.dk/en/coronavirus-in-denmark

[2] Thomas Buhl Andersen and Rógvi Eliasen. 2020. Hand gesture recognition using dual FSR sensorarmbands. https://projekter.aau.dk/projekter/files/320177856/Hand_gesture_recognition_using_dual_FSR_sensor_armbands.pdf

[3] M. Anvaripour and M. Saif. 2018. Hand gesture recognition using force myography of the forearm activities and optimized features. In *2018 IEEE International Conference on Industrial Technology (ICIT)*. 187–192.

[4] Martha L. Blair. 2007. Sex-based differences in physiology: what should we teach in the medical curriculum? *Advances in Physiology Education* 31, 1 (2007), 23–25. https://doi.org/10.1152/advan.00118.2006 arXiv:https://doi.org/10.1152/advan.00118.2006 PMID: 17327578.

[5] Kaixuan Chen, Dalin Zhang, Lina Yao, Bin Guo, Zhiwen Yu, and Yunhao Liu. 2020. Deep Learning for Sensor-based Human Activity Recognition: Overview, Challenges and Opportunities. arXiv:2001.07416 [cs.HC]

[6] Meng-Jung Chung and Mao-Jiun J. Wang. 2009. The effect of age and gender on joint range of motion of worker population in Taiwan. *International Journal of Industrial Ergonomics* 39, 4 (2009), 596–600. https://doi.org/10.1016/j.ergon.2007.11.004 Special issue: Felicitating Colin G. Drury.

[7] E.A Clancy, E.L Morin, and R Merletti. 2002. Sampling, noise-reduction and amplitude estimation issues in surface electromyography. *Journal of Electromyography and Kinesiology* 12, 1 (2002), 1 – 16. https://doi.org/10.1016/S1050-6411(01)00033-5

[8] U. Côté-Allard, C. L. Fall, A. Drouin, A. Campeau-Lecours, C. Gosselin, K. Glette, F. Laviolette, and B. Gosselin. 2019. Deep Learning for Electromyographic Hand Gesture Signal Classification Using Transfer Learning. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 27, 4 (4 2019), 760–771. https://doi.org/10.1109/TNSRE.2019.2896269

[9] Artem Dementyev and Joseph Paradiso. 2014. WristFlex: low-power gesture input with wrist-worn pressure sensors. (10 2014).

[10] Ian Dewancker, Michael McCourt, and Scott Clark. 2016. Bayesian Optimization for Machine Learning : A Practical Guidebook. arXiv:1612.04858 [cs.LG]

[11] Jacob A. George, Tyler S. Davis, Mark R. Brinton, and Gregory A. Clark. 2020. Intuitive neuromyoelectric control of a dexterous bionic arm using a modified Kalman filter. *Journal of Neuroscience Methods* 330 (2020), 108462. https://doi.org/10.1016/j.jneumeth.2019.108462

[12] Amjad Hussain, Keith Case, Russell Marshall, and Steve Summerskill. 2016. Joint mobility and inclusive design challenges. *International Journal of Industrial Ergonomics* 53 (2016), 67 – 79. https://doi.org/10.1016/j.ergon.2015.10.001

[13] Xianta Jiang, Lukas-Karim Merhi, Zhen Gang Xiao, and Carlo Menon. 2017. Exploration of Force Myography and surface Electromyography in hand gesture classification. *Medical Engineering & Physics* 41 (2017), 63 – 73. https://doi.org/10.1016/j.medengphy.2017.01.015

[14] Nan Li, Dapeng Yang, Li Jiang, Hong Liu, and Hegao Cai. 2012. Combined Use of FSR Sensor Array and SVM Classifier for Finger Motion Recognition Based on Pressure Distribution Map. *Journal of Bionic Engineering* 9, 1 (2012), 39 – 47. https://doi.org/10.1016/S1672-6529(11)60095-4

[15] W. Li, C. Hsieh, L. Lin, and W. Chu. 2017. Hand gesture recognition for post-stroke rehabilitation using leap motion. In *2017 International Conference on Applied System Innovation (ICASI)*. 386–388.

[16] F. Orabona, C. Castellini, B. Caputo, A. E. Fiorilla, and G. Sandini. 2009. Model adaptation with least-squares SVM for adaptive hand prosthetics. In *2009 IEEE International Conference on Robotics and Automation*. 2897–2903.

[17] S. J. Pan and Q. Yang. 2010. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 10 (2010), 1345–1359.

[18] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2016. Progressive Neural Networks. arXiv:1606.04671 [cs.LG]

[19] Waneen W. Spirduso. 2005. *Physical Dimensions of Aging.* (2nd ed. ed.). Human Kinetics, Champaign, Ill.

[20] Thida Than, Aye Aye San, and Tin Tin Myint. 2012. Biokinetic Study of the Wrist joint. *International Journal of Collaborative Research on Internal Medicine & Public Health* 4, 5 (2012).

[21] T. Tommasi, F. Orabona, C. Castellini, and B. Caputo. 2013. Improving Control of Dexterous Hand Prostheses Using Adaptive Learning. *IEEE Transactions on Robotics* 29, 1 (2013), 207–219.

[22] K. N. Trong, H. Bui, and C. Pham. 2019. Recognizing hand gestures for controlling home appliances with mobile sensors. In *2019 11th International Conference on Knowledge and Systems Engineering (KSE)*. 1–7.

[23] Zhen Gang Xiao and Carlo Menon. 2019. A Review of Force Myography Research and Development. *Sensors* 19, 20 (2019). https://doi.org/10.3390/s19204557