UNIVERSITY OF CAPE TOWN

DEPARTMENT OF COMPUTER

Computer Science Honours
Final Paper
2016

Title: Using the Myo to recognise the South African Sign Language alphabet

Author: Erin Versfeld

Project Abbreviation: HANDGR

Supervisor(s): Assoc. Prof. James Gain and Assoc. Prof. Deshendran Moodley

Category	**Min**	**Max**	**Chosen**
Requirement Analysis and Design	0	20	0
Theoretical Analysis	0	25	0
Experiment Design and Execution	0	20	20
System Development and Implementation	0	15	5
Results, Findings and Conclusion	10	20	20
Aim Formulation and Background Work	10	15	15
Quality of Paper Writing and Presentation	10		
Quality of Deliverables	10		
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	0	10	0
Total marks		**80**	

Using the Myo to recognise the South African Sign Language alphabet

Honours in Computer Science Thesis

Erin Versfeld
University of Cape Town, Computer Science Department
erinversfeld@gmail.com

ABSTRACT

Gesture recognition seeks to improve how humans interact with computers through the correct identification of meaningful movement and its implications. Gestures from the South African Sign Language (SASL) alphabet are a particularly interesting set to apply this domain, as it is modestly sized (26 gestures), contains both static and dynamic gestures, and many of the gestures are similar. This study investigated the use of machine learning techniques (hidden Markov models [HMMs], K-nearest neighbour [KNN], artificial neural networks [ANNs], Extra Trees [ETs], RandomForests [RFs] and voting classifiers) with commercially available electromyographic (EMG) sensor-based technology in the form of the Myo. A corpus of these gestures performed by 49 different participants, with an average of ten samples per gesture per person, was gathered. Data from eleven of these is invalid, hence only data from 37 was used to train models. The best performing classifier was an ET trained which judged split quality using Gini impurity, required 262 samples to split a node, considered 14 features when splitting, had 366 trees with a maximum of 3774 leaves each and weighted classes according to their subsampling. It produced an accuracy score of 8.05%, precision score of 8.12% and could be trained in ~1.5 minutes.

1. INTRODUCTION

Gesture recognition (GR) is one of many technologies representing the drive in recent years to make human-computer interfaces more naturalistic. The domain seeks to establish a robust means of recognising a range of movements in diverse settings [2, 5, 29, 42]. Applying GR to South African Sign Language (SASL) holds the potential to bring fresh insights to the field.

SASL is the sign language (SL) spoken by the Deaf community in South Africa. The capitalisation of the letter 'D' distinguishes the linguistic community from deaf, which refers to those people grouped according to a medical condition. SLs make use of the entire body to convey meaning, with facial expressions forming a particularly important component. This latter caveat means facial recognition technology needs to progress further before GR can be applied to the language as a whole. However, the alphabet is a viable subset of the language to examine at this time.

The SASL alphabet is gestured with a single hand (Figure 1). Although not a major component of SASL, the alphabet's twenty-six distinct gestures with large amounts of similarity make it a large enough set that accurate recognition implies potential for further generalisability to single-handed SASL gestures. It is a traditional medium of communication between the Deaf and the hearing in situations where one or both do not know or do not have a sign for a word. Therefore, a machine-learning algorithm that is capable of recognising this set of gestures is a worthwhile and valuable starting point.

Application of GR to this sphere also offers an opportunity to work towards bridging the divide between the Deaf and the hearing. Historically, the difference between communication mediums used by these groups has resulted in the Deaf experiencing a reduced set of social opportunities [38]. In South Africa, this has been exacerbated by the legacy of Apartheid and current education policies on SASL,

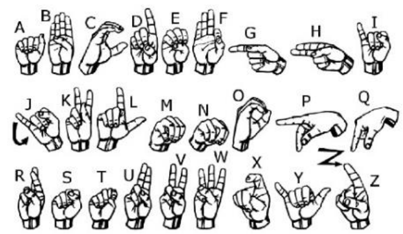

Figure 1. The letters of the SASL alphabet [45]

both as a first and second language [27]. Machine learning (ML) and GR can be applied to this domain so that aids for translating and teaching SASL can be developed in the future. It should be noted that this application can only be realised much further down the line, when the fine level of facial expression recognition required is feasible. Several studies have begun working on closing this gap with other SLs [4, 10, 12, 22, 24, 28, 32, 41, 44, 46]. This study seeks to bring the insights from this research to the South African context, using commercially available technologies.

One such insight, explored here, is that GR devices making use of electromyographic (EMG) sensors are particularly good at recognising SASL alphabet gestures [4, 46]. The Myo armband is a commercially available example of this implementation. This study made use of the Myo in conjunction with HMMs, ANNs, KNNs and ensemble classifiers to assess the viability of EMG-based GR devices with the selected algorithms for recognising these gestures.

1.1 Combinations of devices

This study was conducted in parallel with two similar studies which looked at the use of different GR devices, specifically the Leap Motion Controller (LMC) and Microsoft Kinect (Kinect), to achieve the same goal. The study has therefore been designed in such a way that future research can expand on whether combinations of these devices would be effective for tackling this challenge.

2. RESEARCH QUESTION, AIMS AND CONTRIBUTIONS

The question which this study set out to answer, is: *Which of the six ML algorithms studied performs best when using data from the Myo to recognise gestures from the SASL alphabet?* The four algorithms in question are hidden Markov models, K-nearest neighbour algorithms, artificial neural networks, extra trees, random forests and voting ensembles. The descriptions of the algorithms and motivation for their inclusion are detailed in Section 4.3. The best algorithm will be the one which is able to classify gestures the most accurately while not resulting in overfit. It should produce minimal false positive (Type I errors), and preferably not take more than a few minutes to fit the training data. This will be evaluated by using the model accuracy scoring and standard statistical measures for ML models. Further elaboration on these measures is given in Section 4.4. In answering this question, we hope to inform future studies into the development of a tool for recognising more SASL gestures. The aim is not to produce this tool, nor to develop any tool. The purpose is purely an investigation into the logical mechanics of such a tool. A major

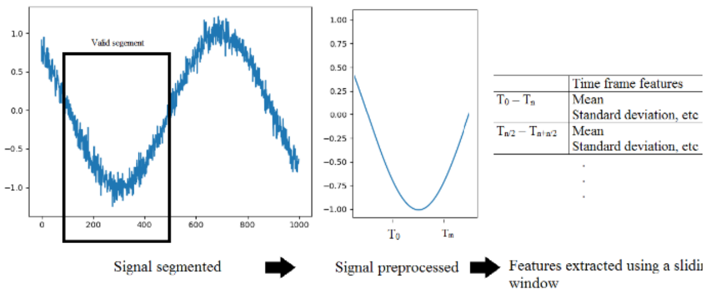

Figure 2. The pre-classification pipeline

contribution is therefore insight into the viability of the Myo in combination with the explored techniques for developing this possible tool.

An additional contribution made by this study is the data set it compiled in order to train the ML classifiers. This data set contains performances of 37 individuals who progress from being unskilled at signing the SASL alphabet to being somewhat skilled at doing so. While not as skilled as those fluent in SASL, the researchers observed that participants initially fumbled and struggled to perform the gestures, but were able to do so smoothly by the end of the recording session. Some individuals did not complete their recordings due to time constraints and did not develop this fluidity. The recordings are made using three devices simultaneously - the Leap Motion Controller (LMC), Microsoft Kinect (Kinect) and Myo, which enables future research to explore the combination of these devices in recognising SASL alphabet. Such a database does not appear to exist for SASL.

3. BACKGROUND

Numerous studies have been conducted on the use of GR technology to recognise SL gestures. However, few of these have looked at SASL, and EMG-based technologies are a relatively recent focus. Nevertheless, the literature motivates our study's design.

While attempts have been made to make this literature widely accessible, several key concepts may require further explanation. A list of excellent resources on the theory and algorithms behind ML is provided in the References [3, 15, 17, 19, 30, 31, 37].

3.1 Technical design

Various EMG technologies and classifier workflows have been used, with a variety of results. For the Myo alone, results vary from 100% person-dependent accuracy of GR for Sinhala [36], to being judged to have low accuracy for SLs [1]. Each of these factors offers insight as to how we should progress with the study.

Overall, the major shortcoming in previous studies was the lack of explanation for the choice of, or obvious experimentation with, various stages of the classifier workflow (Figure 2) for a given classifier. The workflow typically consisted of segmenting the data so that the ideal portion of the recording is used; preprocessing the signal to remove excessive spikiness; extracting key features from the signal to avoid training with redundant data; and finally training the selected model on the data. This is also described as fitting the data to the model. A summary of these combinations on a study by study basis can be seen in Table 1. The major strengths found in this body of research are the motivation for further investigation in the domain, the use of the Myo, and the success of various classifiers.

3.1.1. Hardware. Previous experimentation with hardware configurations indicated that several EMG sensors, in conjunction with an accelerometer and a gyroscope, would be the ideal choice for this study [12, 18, 20, 24, 26, 34]. As the only readily available commercial GR device meeting these specifications, the Myo was selected as the hardware of choice for this study.

The Myo is a wearable GR armband that makes use of eight EMG sensors, a 3D gyroscope (gyro), a 3D accelerometer (ACC) and a magnetometer. The EMG sensors are used to measure muscle tension in the forearm, while the other three sensors form the inertial measurement unit (IMU) of a gesture. The device uses a Bluetooth connection to connect to a digital device[1] and thereby provides a gesture-based interface. Taylor [41] provides a more rigorous description of the Myo and its functionality.

Unfortunately, one of the documented shortcomings of using the Myo is that one cannot control which muscles the EMG sensors receive input from. As demonstrated in Chen et al [12] and Kosmidou et al [22], the control of this variable can mean that a study that used fewer sensors can achieve results comparable with studies that used more, such as Georgi et al [18], Naik et al [34] and Zhang et al [46]. This and other studies with the Myo were only able to enforce its general placement, specifically the forearm [1, 36, 39, 41]. Over the course of this study, an additional short-coming was established: the Myo was unable to consistently stream data to be recorded. It would stop streaming data at unpredictable intervals and required the computer to be restarted before a new, 'stable' connection could be established. It is unclear what caused this issue, but at the time of this writing no dependable solution has been established.

3.1.2. Sensor experimentation. Li et al [24] established that combining EMG and IMU data into a single vector produces the highest accuracy score. However, they only examined this variable when training Gaussian hidden Markov models and K-means clustering classifiers. Further research needs to be done to determine whether these findings are generalisable to other classifiers.

3.1.3. The classifier pipeline. The data processing and classifier training pipeline described in most of the reviewed studies has been: segment the data, preprocess it, extract features and then train the classifiers. This study’s data recording process meant that segmentation was not a concern, and hence has been left out of this review. It is important to note that the data from the EMG and IMU sensors undergo separate preprocessing and feature extraction mechanisms. Whilst some mechanisms are the same, such as normalisation as a preprocessing method, and mean absolute value (MAV) and standard deviation as features, the literature and this study place a lot more emphasis on refining the EMG as opposed to the IMU data. EMG data is finer grained and more distinct to each individual performing the gesture as well as each performance, hence the greater emphasis.

While under reported in comparison to feature extraction, preprocessing techniques are less computationally expensive and can go a long way to improving the performance of a trained model. This is demonstrated excellently by Senevirthne et al. [36], where only two features from each sensor were extracted, and the final model scored 100%. Full-wave rectification [1, 36], normalisation [18] and moving average are all popular techniques. Full wave rectification, inverts any curves of the signals which fall below zero [1, 36] while normalisation fits the data to a bell curve [18]. Finally, the moving average calculates the average for a portion of the signal, and uses this to smooth out that signal portion. This is referred to as a window, and is slid over the data set with some overlap [1, 36]. In the broader signal processing literature, a median filter is preferred to a moving average filter, as the average is distorted by the presence of outliers, while the median is more robust to these effects. Median filters replace the amplitude of the signal at a given point with the median amplitude of the points around it [25]. Scaling is another method which does not appear to have been applied to this problem. This involves standardising the range of features in the data [6, 14, 33].

Feature extraction has been better reported than data preprocessing and segmentation. It is rarely left out of the pipeline, indicating its importance. This assumption was verified in this study by comparing the performance of the classifiers when using only preprocessing against only feature extraction. The most common feature utilised, regardless of the classifier utilised, appears to be the MAV [4, 20, 24, 26, 46], calculated for each sensor signal.

[1] https://www.myo.com/techspecs

Table 1. Designs of stages used in pre-classifier pipelines in previous work

	Preprocessing	Feature extraction	Classifier	Accuracy
[1]	Full wave rectification; Mean	N/A	SVM	Low for fine gestures Potential for dynamic
[4]	N/A	MAV; Moving variance	ANN	95%
[12]	N/A	MAV; Ratio of mean absolute values; Fourth order AR coefficients	BLC	5-10% improvement when compared to that obtained using EMG sensors only
[18]	Z-normalisation	Mean; Standard deviation	HMM	97.8% session-independent accuracy 74.3% user-independent accuracy
[20]	N/A	MAV; Zero crossings; Slope sign changes; Waveform length	LDA	92.6% (air gestures) 88.8% (surface gestures)
[22]	N/A	Integral of absolute value; Difference of absolute mean value; k-th order zero crossings; Skewness; Kurtosis; AR coefficients; Mean frequency; Cepstral parameters; Cepstral coefficients	HMM LDA	97.7%
[24]	N/A	Normalised data segments; Third order AR coefficients; MAV	HMM KNN LDA	95.78%
[26]	Moving average	MAV; Fourth order AR coefficients; DSA; DGA; DIA	BLC HMM	95% user-dependent 89.6% user-independent
[34]	N/A	N/A	SVM	84.83% sensitivity 88.1% specificity
[36]	Remove DC offset; Full wave rectification; Butterworth filter; Moving average	MAV; Standard deviation	ANN	100% user-dependent 94.4% user-dependent
[39]	N/A	Daubechies wavelets	ANN	88.2%
[41]	N/A	Normalised aggregation of data into single vector	LDA KNN	94 – 98 %
[46]	N/A	MAV; Fourth – sixth order AR coefficients; Normalisation; Mean Value; Standard deviation	HMM KNN LDA	93.1% word 72.5% sentence 97.6% user-dependent 90.2% user-independent

This and other averaging operations smooth the signal and add some robustness against noise [1, 4, 18]. The MAV and the standard deviation for each signal are extracted for both IMU and EMG sensors, but the latter typically has several others too. Amongst these are auto-regressive (AR) model coefficients and zero crossings. AR model coefficients offer a more detailed description of the signal than the mean and standard deviation alone. The third through to the sixth coefficients appear to be good choices for EMG signals [12, 22, 25, 26, 46]. Zero crossings are the points at which the signal changes from positive to negative [20, 22].

Once data has undergone preprocessing and had its key features extracted it is fed into a classifier. Frustratingly, although the methods and their improvement of data quality are well motivated, the literature makes no attempt to motivate the choice of these methods in conjunction with the choice of classifiers. For example, Lu et al. [26] claim that they "designed features and algorithms to maintain performance while reducing the computation", without offering an explanation of why the performance is maintained. Therefore, little understanding can be gained regarding the effect of these methods on the performance of the classifier. In the reviewed literature, the classifiers were trained using supervised learning. This means that all data fed into the classifiers had associated with them a label, in this case the letter. These labels are also described as classes. The classifier then trains itself to be able to predict the label which should be associated with new Myo data. It can also then score its accuracy by calculating how often it correctly identified a gesture.

Previous studies have applied support vector machines (SVMs) [1, 34], artificial neural networks (ANNs) [4, 36, 39], Bayesian linear classifiers (BLCs) [12, 26], hidden Markov models (HMMs) [18, 22, 24, 26, 46], linear discriminant analysis (LDA) [20, 22, 24, 41, 46] and K-nearest neighbour (KNN) [24, 41, 46] classifiers to this problem. SVMs belong to the class of discriminitive classifiers, which is to say they learn by modelling the dependence of the label on the associated data [8]. They were the only models in the literature to have performed poorly. However, both of these studies neglected to make use of feature extraction techniques [1, 34], which could have impacted their performance. ANNs are essentially directed graphs whose nodes correspond to neurons and edges to the links between them. A node receives as input the weighted sum of the outputs of the nodes connected to it, and uses this to predict the outcome [37]. Together with KNNs and HMMs, these classifiers performed the most consistently in the literature [4, 18, 22, 24, 26, 36, 39, 41, 46]. BLCs are special case of naive Bayesian classifiers (a probabilistic classifier) where the probability distributions used to predict class membership are all from exponential families [37]. HMMs are statistical classifiers which take as input a set of states, and predict an outcome based on the transition probabilities between these states. LDA is also a statistical method; it attempts to find a linear combination of features in the data to distinguish between classes [37]. HMMs are by far the most preferred model for this domain in the literature. In addition to this, methods for using combinations of models to predict outcomes, referred to as ensemble methods, have become increasingly popular due to their demonstrable improvements over the use of a single classifier [8, 23, 40, 43]. None appear to have been applied to this problem, hence our study investigated their use.

3.2 Experimental designs

Our key criticisms on the reporting of the experimental design in previous studies are their reporting of participant demographics and sampling.

3.2.1. Gestures. Several studies have included both static and dynamic gestures, and those focussing on SLs tend to focus on the recognition of individual letters of the alphabet. In addition to this, they tend to exclude the dynamic gestures, limiting the generalizability of their findings to other single-handed SL gestures. It is also difficult to ascertain whether the exclusions cover gestures that are similar to one another, which would affect classifier performance. The effect of these limitations can be seen in Table 2.

Table 2. Experimental designs used in previous research

	No. participants x No. gestures x No. gesture samples (notes)	Participant demographics	Total samples (notes)	Gesture set characteristics				Accuracy
				Sign language or other	Dynamic or static (notes)	Similar gestures included (notes)	Ratio of gestures examined to those available (reason)	
[1]	Not given x 20 x 20 (training) Not given x 20 x 110 (training)	Not given	+-28 500 (training) Not given (testing)	SL	Static	Yes	20 : 26	Low for fine gestures; Potential for dynamic gestures
[4]	Not given x 10 x 100 (training) Not given x 10 x 100 (testing)	Not given	Not given	Other	Dynamic	No	10 : 64	95%
[12]	5 x 19 x 20 (training) 5 x 19 x 4 (testing)	3 male; 2 female; 20-24 years; performed with left hand	Not given	SL and other	Static (SL) Both (other)	Yes	6 : 26 (SL) N/A (other)	5-10% improvement compared to using EMG sensors only
[18]	5 x 12 x 15 (done five times)	4 male, 1 female; 23-34 years	4 500	Other	Dynamic	No	N/A (other)	97.8% session-independent accuracy; 74.3% user-independent accuracy
[20]	10 x 8 x 3 (training, air) 10 x 8 x 10 (testing, air) 10 x 4 x 3 (training, surface) 10 x 4 x 10 (testing, surface)	Not given	Not given	Other	Both (air) Static (surface)	No (air)	N/A (other)	92.6% (air gestures); 88.8% (surface gestures)
[22]	Not given x 9 x 20	Not given	Not given	SL	Not given	Not given	N/A (words)	97.7%
[24]	1 x 121 x 5 (done four times)	26 year old; right handed	2 420	SL	Not given	Not given	N/A (subwords)	95.78%
[26]	20 x 4 x 32 (small scale) 20 x 15 x 10 (large scale)	13 male, 7 female; 22-27 years	5 560	Other	Both	No (small) Yes (large)	N/A (other)	95% user-dependent; 89.6% user-independent
[34]	7 x 7 x 12 (done twice)	6 male, 1 female; mean age 25.2; mean weight 70.2 kg; mean height 170.2 cm	1 176	Other	Dynamic	Yes	N/A (other)	84.83% sensitivity; 88.1% specificity
[36]	6 x 12 x 150	3 male, 3 female; right handed; 50-70 kg	750	SL	Both	Yes	N/A (words)	100% user-dependent; 94.4% user-dependent
[39]	3 x 17 x 20	Not given	1 020	Other	Both	Yes	17/17	88.2%
[41]	Not given x 10 x60	No given	Not given	SL	Not given	Not given	N/A (words)	94 – 98%
[46]	2 x 72 x 60 (words) 2 x 40 x 10 (sentences)	1 male, 1 female; right handed	8640 (words) 800 (sentences)	SL	Both	Not given	N/A (words and sentences)	93.1% word; 72.5% sentence; 97.6% user-dependent; 90.2% user-independent

3.2.2. Participant demographics. Important participant demographics, such as age, gender and handedness, are also as commonly undocumented as documented. When handedness is reported, right-handedness appears to be the dominant choice [22, 24, 34, 36] and sample population sizes tend to be quite small [1, 12, 18, 24, 34, 36, 39, 46]. These factors limit the generalisability of the studies' findings, and hampers the ability of other researchers to verify their findings. Given the importance of both of these properties, this study sought to avoid this.

4. TECHNICAL AND EXPERIMENTAL DESIGN

All computations were run on a MSI GE702PE and a Lenovo Y50 laptop. Both make use of Intel i7 4710HQ processors, with 2.50GHz clock-speeds. The MSI has 16GB RAM and ran Windows 8.1, and the Lenovo 8GB and ran Solus.

4.1 Data gathering

The study made use of a Myo armband to record data subsequently used to train classifiers. This was done in parallel with the data gathering for the Kinect and the LMC, two devices which were investigated in companion studies. This simultaneous data gathering was achieved by adapting LightBuzz[2], a tool for tracking fingers using the Kinect, to call and kill the respective data gatherers almost simultaneously. The Myo in particular made use of the Myo Data Capture program[3]. Each gesture was recorded for three seconds. The Myo's erratic disconnecting meant that gestures had to be recorded in sets of ten, with the recording program[4] being restarted in between sets. Data gathering took one hour per participant, including the time taken to explain the informed consent form and documenting demographic information. Participant data was annotated with a participant number to preserve anonymity and allow participants to remove their data without impacting the integrity of the database. The Myo was worn by the participant, positioned so that it was secure but comfortable. They were seated in front of the Kinect with their arm resting at approximately 45 degrees to the LMC and hand over the LMC's range of detection. Figure 3 demonstrates the data gathering set up.

Each participant performed the full alphabet ten times. To avoid boredom causing the participants performing the gestures poorly, the order of the letters was randomised. The order was also randomised per participant, as the Kinect in particular would occasionally take some time to 'warm up' and record high quality data. By randomising the order per participant, we ensured that this did not discriminate against particular gestures.

[2] https://github.com/LightBuzz/Kinect-Finger-Tracking
[3] https://market.myo.com/app/55009793e4b02e27fd3abe79/myo-data-capture

[4] The source code for this program and other code relevant to this study can be found at https://github.com/erinversfeld/HonoursProject

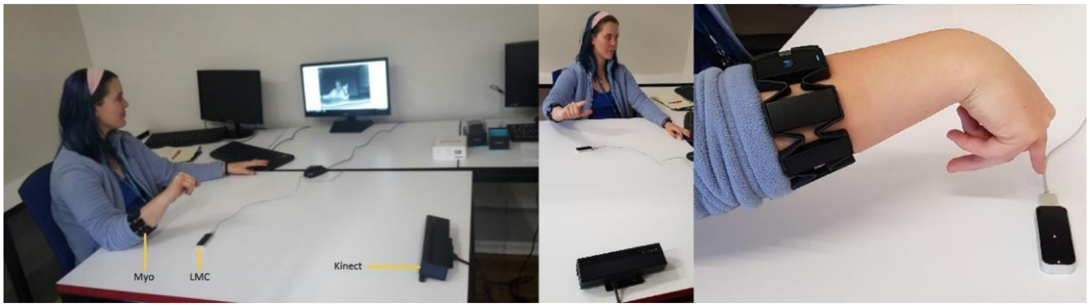

Figure 3. Illustrations of the data gathering set up

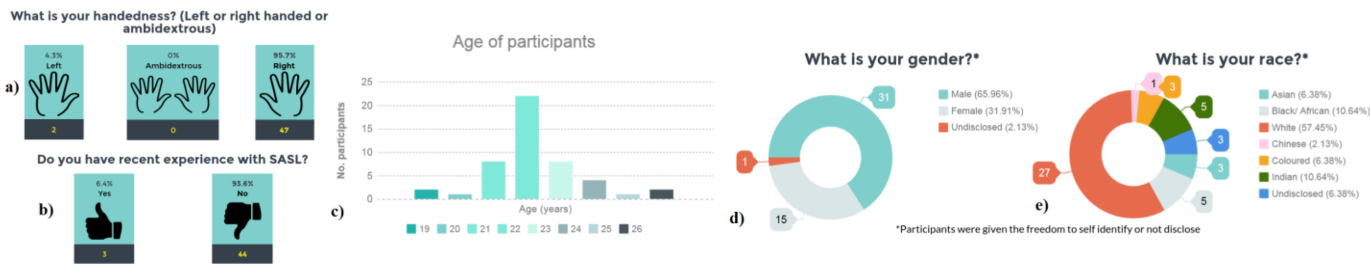

Figure 4. Participant demographics: a) handedness, b) SASL experience, c) age, d) gender and e) race

Table 3. The number of recordings of each letter present in the database

Letter	Number of samples in database	Letter	Number of samples in database	Letter	Number of samples in database
A	1795	J	1860	S	1840
B	1800	K	1825	T	1790
C	1825	L	1830	U	1780
D	1790	M	1850	V	1785
E	1810	N	1855	W	1830
F	1770	O	1870	X	1830
G	1880	P	1855	Y	1780
H	1855	Q	1815	Z	1855
I	1825	R	1805		

An error occurred when gathering data from the first fourteen participants (participant number 0 – 13). The researchers demonstrated several of the gestures incorrectly. They were later corrected by June Bothma, a teacher of SASL at the University of Cape Town (UCT). While this data has been included as part of the final data set because of the value it holds in training classifiers to detect incorrect attempts at gestures, it was left out of the training of the classifiers in this study. This decision was motivated by the need to limit the project scope.

4.1.2. Sample population. Forty-nine unskilled signers were conveniently sampled from the student body of a local university (see Figure 4). Eleven of these were excluded from the set used to train the classifier as discussed previously. Therefore, only thirty-seven were included in the training and testing data for the classifiers. By recruiting a large number of participants relative to previous studies ($\bar{x} \approx 7$, $\sigma = 6$ for previous sample population sizes), demographic variables such as handedness and technical variables such as position of the Myo on the forearm, could be controlled. The change from the study's initial proposal to recruit skilled signers was necessary due to unforeseen circumstances, specifically that the skilled population originally targeted for recruitment were unavailable due to prior commitments. This did not, however, impact the significance of the results of the study or the importance of the data base it produced. Ethical clearance to recruit participants was obtained from the UCT's Faculty of Science and Department of Student Affairs. Copies of these certificates are available upon request.

4.1.3. Gesture corpus. Individual letters were recorded, as opposed to finger-spelt words, as fingerspelling by SASL speakers is very fast, and the capability of the devices' in such cases is not well documented. The inclusion of all the letters of the alphabet meant that the classifiers could be trained to accommodate very similar gestures - such as A, E and S - as well as static and dynamic gestures. The distribution of these letters is given in Table 3.

4.2 The pre-classification pipeline

In accordance with the literature, this study preprocessed and extracted features from EMG and IMU data differently. Segmentation was built into the data gathering. The selected algorithms were implemented in Python 3 using the numpy[5], pandas[6], scipy[7], biosppy[8], statsmodels[9] and scikit-learn libraries[10].

Both the EMG and the IMU data underwent normalisation and scaling when being preprocessed. Full wave rectification, normalisation, a median filter and scaling were also applied to the EMG data. For features, the MAV and standard deviation were calculated for both the IMU and EMG data. The EMG data also had the moving average, third through sixth AR coefficients and zero crossings calculated. The features from both sets of data were then concatenated into a single vector to describe a performance of a gesture.

4.3 Classifier development and training

Six classifiers were investigated in this study. The study's limited time frame motivated the number of explored classifiers. The specific classifiers investigated were HMMs with multinomial discrete emissions, KNN classifiers, multilayer perceptron ANNs, and RandomForest (RF), Extra Trees (ET) and Voting ensembles. Python 3 was used to implement these classifiers. The scikit-learn and seqlearn[11] libraries in particular were used.

[5] http://www.numpy.org/
[6] https://pandas.pydata.org/
[7] https://www.scipy.org/
[8] http://biosppy.readthedocs.io/en/stable/
[9] http://www.statsmodels.org/stable/index.html
[10] http://scikit-learn.org/stable/index.html
[11] http://larsmans.github.io/seqlearn/

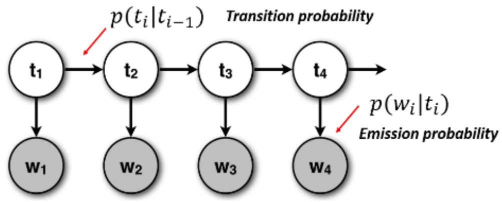

Figure 5. A diagram of the construction of an HMM[12]

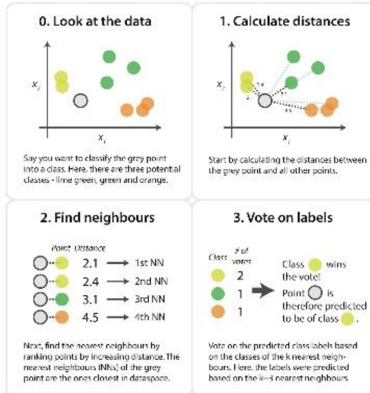

Figure 6. A diagram of the KNN algorithm[13]

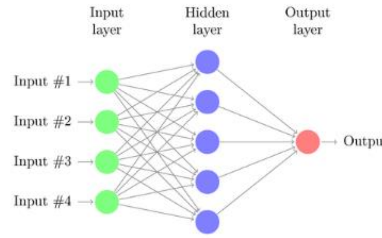

Figure 7. A diagram of an ANN[14]

When exploring the best parameters to use for implementing these algorithms, two techniques, namely, GridSearchCV[15] and RandomizedSearchCV[16] were utilised. These algorithms automate the exploration of the parameter space to find the best construction of a classifier for the data provided. Training a classifier involves reserving a small portion of the data for evaluating the model's accuracy (see Section 4.4) and then providing the classifier with the rest. It then fits this data as best as it can. When performing this search the test data was the data from a randomly selected participant. This participant was kept constant across the searches for all of the classifiers. This process is repeated several times to settle on those parameter value which are selected most often and hence determined to be best suited for this data set.

4.3.1. Hidden Markov models. HMMs are built on the theory behind Markov chains, and describe a process which moves between a set of states. The Markov chain property means that the probability of each subsequent state is only dependent the probability of each subsequent state is only dependent upon the previous states. Therefore, HMMs are initialized using a set of initial probabilities and a set of transition probabilities [5, 12, 28]. This process is illustrated in Figure 5.

HMMs exhibit a degree of invariance to local time warping which is associated with things such as natural variations in the speed of speech [11]. A similar warping was expected in this study, as participants perform dynamic gestures at differing speeds. While only two out of the 26 gestures were dynamic, this characteristic could prove useful when generalising the results to other single-handed gestures.

The particular HMM implemented in this study is a multinomial HMM (MNHMM). Other studies have made use of a variety of different HMMs, and by investigating a new one this study will contribute to the knowledge surrounding this problem. MNHMMs are a special case of HMM where at each moment, a finite set of environmental or experimental conditions enter the system and effect the parameters on the state space [13]. This dependence is represented as a multinomial function, hence the name.

The parameters available for tuning in seqlearn's HMM API are limited. For example, one cannot experiment with the number of hidden layers. One is able to experiment with the alpha value used to smooth the model and the algorithm used to find the best sequence of hidden states to predict a class. Both available algorithms were experimented with, as was a range of 5-10 in steps of 0.001 for alpha. This range was decided upon through trial and error. One has no control over the initial and transition probabilities in seqlearn's HMM, but instead the API estimates these values itself. This is a shortcoming as investigation of and experimentation with these parameters could prove beneficial to the performance on HMMs in this domain.

4.3.2. K-nearest neighbor. When training a K-nearest neighbor classifier, the data is mapped to a feature space (for example, the two-dimensional feature space in Figure 6), and given the label associated with it. The model does no classifying or calculations yet, it merely holds the data. When test data is fitted, it maps the data onto the feature space and uses the k-nearest nodes to this new node to determine its classification (see Figure 6).

While similar to the nearest neighbor algorithm, this draws on more neighbours, and in so doing, produces a more robust classifier and a smoother decision boundary. When K is very large, the classifications tend towards being the same. Therefore, cross-validation is utilised to obtain the most optimal value for K [37]. K-nearest neighbour has performed well in studies that utilised a combination of static and dynamic gestures, hence its inclusion.

The KNN provided by scikit-learn has a number of parameters which one can optimize. The ones investigated here, are the algorithm used to calculate the value a new data point should have (*weights*); the algorithm used to calculate the nearest neighbours to a given point (*algorithm*); the number of neighbours to take into account when calculating the value of a new point (*n_neighbors*); and the number of nodes used to construct the model (*leaf_size*). All *weight* and *algorithm* values provided by the API were experimented with. The range of *n_neighbors* experimented with was [1 – number of samples in the training set], inclusive, while for *leaf_size* it was [5 – 1000] inclusive. The ranges were estimated based on trial and error, and limited by the time it took to run them.

4.3.3. Artificial neural networks. An ANN consists of layers of nodes connected by 'neurons' of varying weights [32]. Data enters at the input layer of nodes, is processed through hidden layers, and the output layer of nodes gives the result (see Figure 7).

[12]http://www.cs.virginia.edu/~hw5x/Course/CS6501-Text-Mining/_site/mps/mp3.html
[13]https://cambridgespark.com/content/tutorials/implementing-your-own-knearest-neighbour-algorithm-using-python/index.html
[14] http://www.texample.net/tikz/examples/neural-network/
[15]http://scikitlearn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
[16]https://scikitlearn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html

ANNs' are able to generalise and associate data through reinforcement learning. This means, an ANN can find reasonable solutions for similar problems of the same class, even if not trained for them. Hence, they, have a high fault tolerance to noisy data [44]. This is useful in this context, as many variables affect the accurate classification of gestures across participants.

The particular class of ANN implemented in this study is a multilayer perceptron (MLP) ANN. Scikit-learn's API allows for effective experimentation with parameters for MLPs. Variables such as the size of the hidden layers in the model, the algorithm used to determine a node's output (*activation* algorithm), step size and the penalty maximum number of iterations allowed to search the solution space, the step size (*learning rate*) and penalty weighting (*alpha*) were all experimented with. All values for activation, and *learning* rate which are supported by the API were explored. The range of hidden layer sizes explored was 26 (the number of classes) - the number of data sample, and [1.00000000e-05, 1.00000000e-03, 1.00000000e-01, 1.00000000e+01, 1.00000000e+03] for *alpha*. The decision regarding which values to investigate was initially based on previous work, but then expanded based on the poor performance observed during the trial phases. The model was run up to 2000 times in an attempt to find the best weight for the data.

4.3.4. RandomForest ensemble. RFs make use of multiple trees based on a random selection of features and individuals using bootstrap resampling. They provide an average of these multiple classification trees [31, 37].

The averaging between multiple trees makes it hard to over-fit RFs. They are also easy to tune, and quick to train, making them a viable option to explore. Values for the number of trees to use (*n_estimators*) and the algorithm to measure the quality if the data split (*criterion*) were experimented with in addition to the weighting of the classes (*class_weight*), minimum number of samples required per leaf (*min_samples_leaf*) and the maximum number of leaf nodes per tree (*max_leaf_nodes*). Both *criterion* values supplied by the API are investigated, along with 'balanced' and 'balanced_subsample' as options for *class_weight*. *n_estimators* in the range 26 – 1000 inclusive, *min_samples_leaf* in the range 1 – half the number of samples, and *max_leaf_nades* in the range 2 – the number of samples were examined. As the literature was unable to guide the selection of these value these ranges were determined largely through trial and error.

4.3.5. Extra Trees ensemble. ET is a special case of RFs where the entire sample rather than a resampled subset is used at each step, and decision boundaries are picked at random. These classifiers often perform comparably to RFs, and have been known to outperform them when trained on real-world data. The same parameters and associated values explored for RandomForests were explored for Extra Trees.

4.3.6. Voting ensemble. Simply put, voting ensembles take the output of several models (which are not necessarily the same type) and based on this output, vote which class should be assigned to a given input. A simple majority vote can be used, or a weighted one. Voting classifiers are well suited to label-centric problems, such as this [9].

Multiple constructions of voting classifiers were experimented with, from combinations of two of the other explored classifiers, to combinations of all of them. All possible combinations of ANN, KNN, ET and RF were experimented with. In each case these models were created using the best parameters found for them in the preceding investigations. HMMs were excluded from this experimentation due to their excessively poor performance (see Table 4). Soft voting was used to decide which of these classifiers' outputs should be used.

4.4 Classifier experimentation and evaluation

When training the classifiers, the study examined user-dependent scenarios in which the data had undergone both preprocessing and feature extraction. User dependence here means that all the data from a single participant was reserved as test data. Data from all other users forms the training data. Initial investigations found that the use of only EMG or only IMU data produced results too poor to warrant further study, verifying the findings of previous works. This also generalizes those findings to the classifiers explored here. Having found the best set of parameters to use to train a given class of model, 37 instances of that model were created, trained and evaluated, with each participant's data having the opportunity to form the test data. These models were then used to describe the average performance of this class of model for the data.

Several measurements were used to assess the classifiers' accuracy. The rate at which a classifier produces type I (false positive) and II (false negative) errors was using confusion matrices. True positive (TP) is the case when the predicted and actual classes are the same; false positive (FP) when the predicted class is incorrect; true negative (TN) when the gesture is correctly predicted as not being part of a class it is not part of; and false negative (FN) when it is incorrectly predicted as not being a part of a class. Confusion matrices plot the number of times an instance of a class is correctly identified Other measures of a classifier's accuracy include its recall (Equation 1) and precision (Equation 2).

$$\frac{TP}{TP+FN} \qquad \text{Equation 1}$$

$$\frac{TP}{FP+TP} \qquad \text{Equation 2}$$

Having estimated a classifier's accuracy using these measures, we need to verify our model is not performing well due to over-fitting. This is achieved using the k-fold or holdout measure of accuracy. A K-fold measure refers to when a data set is divided into K number of "folds", K-n of which are used for training the model, and n reserved for testing its accuracy [16]. In all of the libraries used here to implement classifiers, the models are provided with a score() method, which produces this measure.

5. RESULTS AND DISCUSSION

ETs are the best performing classifiers in terms of model accuracy, rate of false positives and time taken to train. ANNs offer marginally better model accuracy over ETs and voting classifiers, although the performance range is the same for KNNs, ANNs, ETs, RFs and voting classifiers (see Figure 8). The low performances conflict with the literature, but can be accounted for by the nature of the data recording. The lowest performing classifier was the HMM, which produces truly random results at <3.8% accuracy. Time restrictions meant no exploration of the impact of preprocessing technique-extracted features combinations on classifier performance could be done.

5.1 Parameter experimentation

The extensive search of the parameter space indicated that the following sets of parameters were optimal for the current data:

5.1.1. K-Nearest Neighbour. KNN's performed best when weighting votes by distance, calculating the distance between nodes using a k-d tree, considering 182 neighbours in order to determine membership, and limiting the size of each leaf to 614 samples. Weighting votes by distance assigns weight proportional

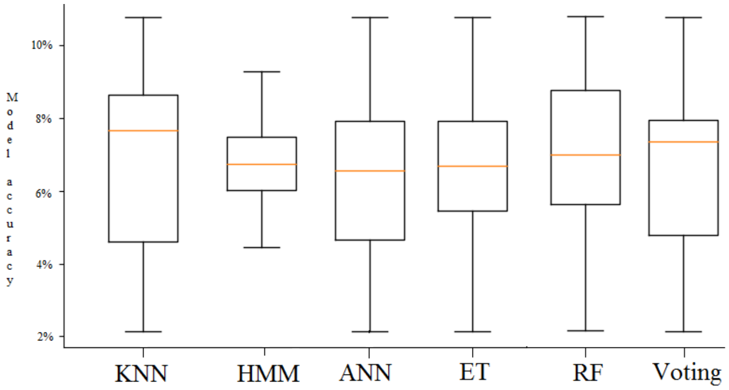

Figure 8. Boxplots of the spread of accuracy scores for the various classifiers over 37 runs

Table 4. Average accuracy of each classifier

	Mean recall (Standard deviation)	Mean precision (Standard deviation)	Mean accuracy (Standard deviation)	Time taken to train
KNN	7.78% (3.39%)	6.78% (2.59%)	6.78% (2.49%)	>1 minute
HMM	1.73% (2.05%)	3.7% (0.996%)	3.72% (0.94%)	~1 minute
ANN	4% (2.27%)	8.08% (2.66%)	8.11% (2.61%)	~2.5 minutes
ET	5.05% (2.45%)	8.12% (2.81%)	8.05% (2.75%)	~1.5 minutes
RF	6.24% (2.28%)	6.62% (2.2%)	6.62% (2.15%)	>1 minute
Voting	7.16% (2.56%)	8% (2.45%)	8.07% (2.46%)	~3 minutes

to the inverse of the distance from the query point while the k-d tree constructs a tree structure which recursively partitions the parameter space along the data axes[17]. This weighting ensures that nearer neighbours contribute more to the final vote than further neighbours, eliminating, any skewness in the data from irregular sampling. Skewness in certain sensor measurements is counter-acted against by the preprocessing methods applied to the data. The large number of neighbours provides a smoother transition boundary between classes, making it easier to distinguish between classes and reducing the model's variance. Whilst this would ordinarily be an issue as it introduces bias and over-fitting [17], the low accuracy of this model assures us that this is not the case. The massive reduction of the feature space (from over 3400 per gesture in the original data to 100 per gesture following feature extraction), also boosted the KNN's performance. The rapid construction time of the model is accounted for by the use of the k-d tree and large number of samples per leaf.

5.1.2. Hidden Markov Model. HMM's performed best when decoded using the Viterbi algorithm and a value of 5.0 as the smoothing factor for the model. Given the accuracy of the model (see Table 4), and the fact that it appears to classify all gestures as *c* by default (see Figure 9), it's likely that the smoothing factor is the dominant contributor to this model's performance. It is reducing all categorical variables to *c*. It is unclear why in particular it is reducing them to *c* as opposed to any other letter, but this ultimately produces the highest accuracy as it will always classify *c* correctly. Having this 100% accuracy rate for a single letter then skews the accuracy of the model as a whole.

5.1.3. Artificial Neural Network. An MLP with 5664 nodes per hidden layer, L2 penalty parameter of 0.001, logistic sigmoid activation function and which adapted its learning strategy according to the rate of training loss was able to produce an optimal solution within 1000 iterations. Requiring so many iterations mean that it took significantly longer to get the final model, hence ANNs took the longest out of all of the models to train (see Table 4). While normally having so many nodes would result in an over-fit model, the penalty is sufficient to avoid overfitting while not over penalizing either. The logistic sigmoid activation function introduces nonlinearity into the model. EMG data is inherently non-linear, hence bringing non-linearity into the model improves its ability to model this data.

5.1.4. Extra Trees. The best performing classifier was an ET trained with parameters which judged split quality using Gini impurity, required 262 samples to split a node, considered 14 features when splitting, had 366 trees with a maximum of 3774 leaves each and weighted classes according to their subsampling. The Gini evaluation is ultimately what makes the ET better at minimizing misclassification[18] even if its model accuracy is not as high as the MLP.

5.1.5. RandomForests. Scikit-learn's default RF performed best. Much like the ET, its misclassification rate was minimized thanks to the fact that the Gini evaluation of impurity is the default for this model. It produced 10 trees with a maximum of 10 features (the square root of the number of features in the data) being considered when splitting into a new tree. Trees could have an unlimited number of leaf nodes but required at least two samples in a node before splitting could occur.

5.1.6. Voting. A voting classifier constructed out of a KNN, ANN, ET and RF with the best parameters for each of these was the best possible construction of a voting classifier for this data. Soft voting meant that it was able to make use of the best performances of each of these classifiers when classifying new input. However, with so many additional considerations going into a single classification it took so long to train and evaluate.

As predicted by Abreu et al [1], the classifiers all indicate that the Myo is very good at correctly classifying dynamic gestures, and poor with static gestures. This can be seen in the confusion matrices for the classifiers in Figure 9, where the darkest blocks are those in which either *j* or *z* are correctly identified. The exception to this rule is, of course, the HMM, because it randomly classifies gestures as *c* more often than not. The ANN is best able to classify *j* and *z* because its parameter tuning has made it more sensitive to the differences between all of the gestures.Voting classifiers as they are constructed here perform similarly to ANNs and ETs because they make use of the strengths of these models. However, they do not obtain equal performancesbecause occasionally another model's output will have the highest vote in spite of being incorrect. Although it lowers the average accuracy of the model, the voting classifier performs more consistently as a result of having these other classifiers form part of its makeup. This is evidenced by its lower standard deviation than that of ANNs or ETs (see Table 4).

[17] http://ogrisel.github.io/scikit-learn.org/sklearn-tutorial/modules/neighbors.html

[18] https://www.quora.com/Machine-Learning/Are-gini-index-entropy-or-classification-error-measures-causing-any-difference-on-Decision-Tree-classification

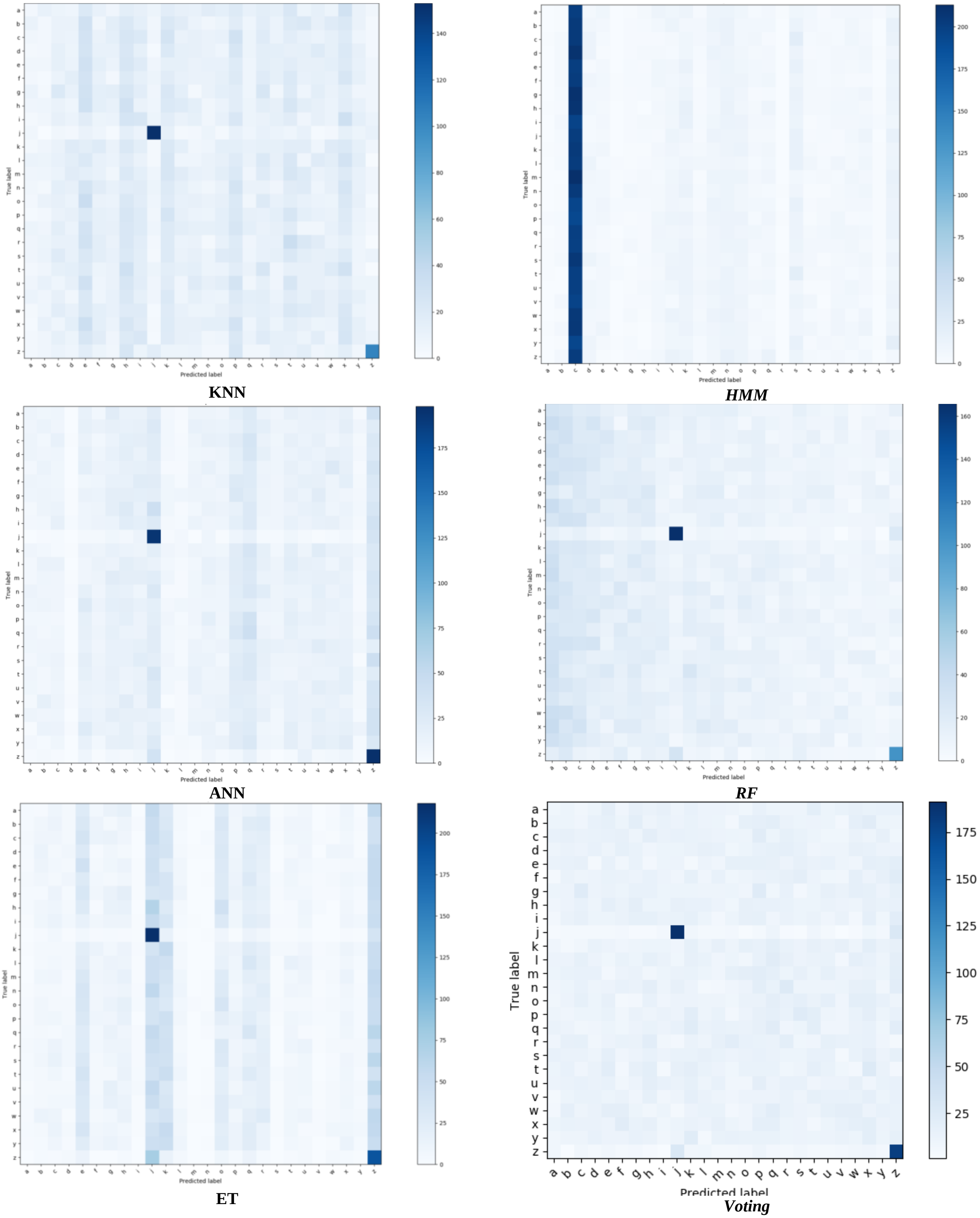

Figure 9. Confusion matrices for each of the classifiers based on performances with 37 different sets of test data

5.2 Comparison to previous work

In comparison to the previous work, the results obtained by this study are, in short, dismal. This experience is not unique to this work, however, as the companion studies by Borysova [7] and Kooverjee [21] found similar albeit not such extreme discrepancies between their findings and those of previous work. The hypothesis is that this is as a result of the data gathering set up which was used.

The set up strove to control the minimum number of variables, so as to preserve as natural an environmental profile as possible. The setting had natural light and the participants were encouraged to perform the gestures as was most comfortable for themselves as opposed to what would produce the best quality data for a given device. In addition to this, negotiating the simultaneous recording of the three devices appears to have severely compromised the Myo's data. The practice of waiting for the participant to have their hand in the position for the gesture before beginning recording appears to have ended up producing lower quality data than previous studies had available to them. This is because the muscle signal that the Myo uses to recognise a gesture is strongest when one is in the process of performing said gesture [35]. The signal the Myo records when one is already in position is merely a maintenance signal. That is to say, rather than the signal which tells the muscles how to arrange themselves, it is a signal which says, 'just keep on doing what you're doing'. There is far less to distinguish this signal for different letters than there is for the former. However, in waiting for the participant to be ready for the recording to begin the Myo ended up recording the latter. Hence the classifiers have a substantially harder time distinguishing between the gestures.

To compound this issue further, more distinct participants made up this data set than in any of the previous work. While this may not have been a major issue for the LMC and the Kinect, which are both essentially visually based devices, it has a severe impact on the ability of the Myo. Each performance of a gesture produces a unique EMG signal, even when performed by the same person [35]. The physical characteristics of participants such as perspiration rate, heart rate and muscle density all contribute to this variation [35]. Therefore, there is substantially more noise in the data for the Myo when using more participants than there is for alternate devices. This noise makes it harder to construct accurate, adaptable models with the data.

5.3 Comparison to companion studies

Borysova's companion study on the LMC [7] looked at SVMs, KNNs, ANNs and voting classifiers as potential solutions to recognizing SASL alphabet gestures, as did Kooverjee's with the Kinect [21]. Much like this study, KNNs were among the top performing classifiers, but not the best. In both instances SVMs appear to be the top performers.

Both Borysova and Kooverjee [7, 21] also found that the data used to train the classifiers may have been flawed and thereby produced weaker results than those in their literature. This is a similar finding to the one presented here, but one which they were able to verify by visualizing the gestures. While there is no readily available software to render visualization of gestures from Myo data and hence no such confirmation can be done, the fact that this was the case for one of the three devices lends credence to the possibility that this was the case for the Myo too.

6. CONCLUSIONS AND FUTURE WORK

In answer to this study's research question, best performing classifier in terms of model accuracy, lowest rate of false positives and time taken to train was ET. While not offering the highest model accuracy, it is still within 0.06% of the highest model accuracy, therefore its higher precision and faster training time make it a more favourable option.

Building on the findings of this study, there are two major streams of future work. The first seeks to continue tackling the problem of GR of SASL, while the second deals with advancing the field as a whole more directly.

6.1 Gesture recognition of South African Sign Language

The data gathering procedure must be redesigned and reevaluated to establish whether or not the poor performance of the classifiers in this study is due to the inherently poor quality of data produced by the Myo or due to the poor data gathering design as hypothesised. Doing so will inform future studies as to whether or not the Myo is a worthwhile tool for such investigations.

Investigation into alternative EMG and IMU based technologies is necessary. Even if future work shows the data gathering process here to be at fault rather than the data the Myo records, the Myo will still not be an ideal tool for research or home use, on account of its temperamental behaviour. Should the Myo prove to record poor quality data it would be worthwhile having alternative hardware to compare it to to establish whether the weakness is unique to the Myo's design or whether it is a phenomenon universal to EMG-based GR devices.

The corpus must also be developed further. Currently, right handed white males in their twenties are represented most strongly. While racial identity will probably not impact the performance of the Myo in the same way it might a visually based GR device, the difference in muscle densities and other physical characteristics between men and women and young and mature adults may have an effect. Additionally, this database should continue to be developed using all three devices, to allow for future research into their combinations.

Future work should analyse combinations of devices as true recognition of SASL will require some means of interpreting facial expression. This need not necessarily be visual as engineers may invent something far more sophisticated in the future, but visual technology is currently the best means we have of tackling this part of the problem. Therefore, the Kinect in combination with the Myo may prove to have many benefits for tackling this problem.

6.2 Advancements in the understanding of ML in general

It remains necessary is to investigate the effects of various combinations of feature extraction and preprocessing techniques on the performance of classifiers, and to develop a deeper understanding of why this is the case.

7. ACKNOWLEDGEMENTS

The author would like to thank her supervisors, Assoc. Profs. James Gain and Deshendran Moodley for their guidance throughout the year, and their assistance in developing and refining this research as well as her team-mates, Anna Borysova and Shaheel Kooverjee. Special thanks must also be given to Prof. Edwin Blake, for his insights into research of SASL and his experience in doing research with the community. June Bothma, the SASL teacher to the University of Cape Town audiology students, is also thanked for her provision of resources to learn the SASL alphabet and her willingness to provide access to her students. She is responsible for the massive improvement in the quality of the recordings as a result of her time and these resources.

8. REFERENCES

[1] João Gabriel Abreu, João Marcelo Teixeira, Lucas Silva Figueiredo, and Veronica Teichrieb. 2016. Evaluating Sign Language Recognition Using the Myo Armband. In Virtual

and Augmented Reality (SVR), 2016 XVIII Symposium on. IEEE, 64–70.
[2] Suat Akyol, Ulrich Canzler, Klaus Bengler, and Wolfgang Hahn. 2000. Gesture Control for Use in Automobiles. In MVA. 349–352.
[3] Robbie Allen. [n. d.]. My curated list of AI and machine learning resources from around the web. https://unsupervisedmethods.com/my-curated-list-of-aiand-machine-learning-resources-from-around-the-web-9a97823b8524. ([n. d.]). Accessed: 2017-09-17.
[4] Varadach Amatanon, Suwatchai Chanhang, Phornphop Naiyanetr, and Sanitta Thongpang. 2014. Sign language-thai alphabet conversion based on electromyogram (emg). In Biomedical Engineering International Conference (BMEiCON), 2014 7th. IEEE, 1–4.
[5] Nicola Bizzotto, Alessandro Costanzo, Leonardo Bizzotto, Dario Regis, Andrea Sandri, and Bruno Magnan. 2014. Leap motion gesture control with Osirix in the operating room to control imaging first experiences during live surgery. Surgical innovation (2014), 1553350614528384.
[6] Avrim L Blum and Pat Langley. 1997. Selection of relevant features and examples in machine learning. Artificial intelligence 97, 1 (1997), 245–271.
[7] Anna Borysova. 2017. Leap Motion Controller for South African Sign Language alphabet recognition. Honours Thesis. University of Cape Town, Cape Town.
[8] Pascal Brandt. 2014. An Investigation of Multi-label Classification Techniques for Predicting HIV Drug Resistance in Resource-limited Settings. Ph.D. Dissertation. University of KwaZulu-Natal, Durban.
[9] Gavin Brown. 2010. Ensemble learning. Springer Press.
[10] Tiago Cardoso, João Delgado, and José Barata. 2015. Hand gesture recognition towards enhancing accessibility. Procedia Computer Science 67 (2015), 419–429.
[11] Lingchen Chen, FengWang, Hui Deng, and Kaifan Ji. 2013. A survey on hand gesture recognition. In Computer Sciences and Applications (CSA), 2013 International Conference on. IEEE, 313–316.
[12] Xiang Chen, Xu Zhang, Zhang-Yan Zhao, Ji-Hai Yang, Vuokko Lantz, and Kong-Qiao Wang. 2007. Hand gesture recognition research based on surface EMG sensors and 2D-accelerometers. In Wearable Computers, 2007 11th IEEE International Symposium on. IEEE, 11–14.
[13] Marina Cidota and Monica Dumitrescu. 2012. A multinomial-hidden Markov model for communication systems influenced by external factors. In Applied Computational Intelligence and Informatics (SACI), 2012 7th IEEE International Symposium on. IEEE, 2012.
[14] Thomas G Dietterich et al. 2000. Ensemble methods in machine learning. Multiple classifier systems 1857 (2000), 1–15.
[15] Pedro Domingos. 2012. A few useful things to know about machine learning. Commun. ACM 55, 10 (2012), 78–87.
[16] Scott Fortmann-Roe. 2012. Accurately measuring model prediction error. http://scott.fortmann-roe.com/docs/MeasuringError.html. Accessed: 2017-09-17.
[17] Scott Fortmann-Roe. 2012. Understanding the bias-variance tradeoff. http://scott.fortmann-roe.com/docs/BiasVariance.html. Accessed: 2017-09-17.
[18] Marcus Georgi, Christoph Amma, and Tanja Schultz. 2015. Recognizing Hand and Finger Gestures with IMU based Motion and EMG based Muscle Activity Sensing. In BIOSIGNALS. 99–108.
[19] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. Deep Learning. MIT Press. http://www.deeplearningbook.org.
[20] Shuo Jiang, Bo Lv, Xinjun Sheng, Chao Zhang, Haitao Wang, and Peter B Shull. 2016. Development of a real-time hand gesture recognition wristband based on sEMG and IMU sensing. In Robotics and Biomimetics (ROBIO), 2016 IEEE International Conference on. IEEE, 1256–1261.
[21] Shaheel Kooverjee. Investigating machine learning classifiers for sign language recognition using the Microsoft Kinect. Honours Thesis. University of Cape Town, Cape Town.
[22] Vasiliki E Kosmidou, Leontios J Hadjileontiadis, and Stavros M Panas. 2006. Evaluation of surface EMG features for the recognition of American Sign Language gestures. In Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE. IEEE, 6197–6200.
[23] Sotiris B Kotsiantis, Ioannis D Zaharakis, and Panayiotis E Pintelas. 2006. Machine learning: a review of classification and combining techniques. Artificial Intelligence Review 26, 3 (2006), 159–190.
[24] Yun Li, Xiang Chen, Jianxun Tian, Xu Zhang, Kongqiao Wang, and Jihai Yang. 2010. Automatic recognition of sign language subwords based on portable accelerometer and EMG sensors. In International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction. ACM, 17.
[25] [30] Shin Ta Liu. 2006. Nonlinear signal processing: a statistical approach. (2006).
[26] Zhiyuan Lu, Xiang Chen, Qiang Li, Xu Zhang, and Ping Zhou. 2014. A hand gesture recognition framework andwearable gesture-based interaction prototype for mobile devices. IEEE transactions on human-machine systems 44, 2 (2014), 293–299.
[27] Lucas Magongwa. 2010. Deaf education in South Africa. American Annals of the Deaf 155, 4 (2010), 493–496.
[28] Giulio Marin, Fabio Dominio, and Pietro Zanuttigh. 2014. Hand gesture recognition with leap motion and kinect devices. In Image Processing (ICIP), 2014 IEEE International Conference on. IEEE, 1565–1569.
[29] Mark T Marshall, Joseph Malloch, MarceloMWanderley, et al. 2007. Gesture Control of Sound Spatialization for Live Musical Performance. In Gesture Workshop. Springer, 227–238.
[30] Abdelhamid Mellouk and Abdennacer Chebira. 2009. Machine Learning. InTech.
[31] Tom Michael Mitchell. 2006. The discipline of machine learning. Vol. 3. Carnegie Mellon University, School of Computer Science, Machine Learning Department.
[32] Sushmita Mitra and Tinku Acharya. 2007. Gesture recognition: A survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 37, 3 (2007), 311–324.
[33] Eric Mjolsness, David H Sharp, and Bradley K Alpert. 1989. Scaling, machine learning, and genetic neural nets. Advances in applied mathematics 10, 2 (1989), 137–163.
[34] Ganesh R Naik, Dinesh Kant Kumar, et al. 2010. Twin SVM for gesture classification using the surface electromyogram. IEEE Transactions on Information Technology in Biomedicine 14, 2 (2010), 301–308.
[35] M.B.I. Raez, M.S. Hussain and F. Mohd-Yasin. 2006. Techniques of EMG signal anaylsis: Detection, processing,

classification and applications. Biological Procedures Online 8 (2006): 11 – 35.

[36] RGDC Senevirathne, ALP Madushanka, and LMH Wijesekara. 2016. Framework for Sinhala sign language recognition and ranslation using a wearable armband. Ph.D. Dissertation.

[37] Shai Shalev-Shwartz and Shai Ben-David. 2014. Understanding Machine Learning: From theory to algorithms. Cambridge University Press.

[38] Laurene Simms, Melissa Rusher, Jean F Andrews, and Judy Coryell. 2008. Apartheid in deaf education: Examining workforce diversity. American Annals of the Deaf 153, 4 (2008), 384–395.

[39] Ananta Srisuphab and Piyanuch Silapachote. 2016. Artificial neural networks for gesture classification with inertial motion sensing armbands. In Region 10 Conference (TENCON), 2016 IEEE. IEEE, 1–5.

[40] Aik Choon Tan and David Gilbert. 2003. Ensemble machine learning on gene expression data for cancer classification. (2003).

[41] Jackson Taylor. 2016. Real-time translation of American Sign Language using wearable technology. (2016).

[42] Shawn Trail, Michael Dean, Gabrielle Odowichuk, Tiago Fernandes Tavares, Peter F Driessen,WAndrew Schloss, and George Tzanetakis. 2012. Non-invasive sensing and gesture control for pitched percussion hyper-instruments using the Kinect. In NIME.

[43] Giorgio Valentini and Francesco Masulli. 2002. Ensembles of learning machines. In Italian Workshop on Neural Nets. Springer, 3–20.

[44] Lynette Van Zijl and Dean Barker. 2003. South african sign language machine translation system. In Proceedings of the 2nd international conference on Computer graphics, virtual Reality, visualisation and interaction in Africa. ACM, 49–52.

[45] Xu Zhang, Xiang Chen, Yun Li, Vuokko Lantz, Kongqiao Wang, and Jihai Yang. 2011. 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 (2011), 1064–1076.