Towards Building an Intelligent Tutor for Gestural Languages using Concept Level Explainable AI

by

Prajwal Paudyal

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved January 2020 by the
Graduate Supervisory Committee:

Sandeep K.S. Gupta, Chair
Ayan Banerjee
Ihan Hsiao
Tamiko Azuma
Yezhou Yang

ARIZONA STATE UNIVERSITY

May 2020

ABSTRACT

Languages, specially gestural and sign languages, are best learned in immersive environments with rich feedback. Computer-Aided Language Learning (CALL) solutions for spoken languages have successfully incorporated some feedback mechanisms, but no such solution exists for signed languages. Computer Aided Sign Language Learning (CASLL) is a recent and promising field of research which is made feasible by advances in Computer Vision and Sign Language Recognition(SLR). Leveraging existing SLR systems for feedback based learning is not feasible because their decision processes are not human interpretable and do not facilitate conceptual feedback to learners. Thus, fundamental research is needed towards designing systems that are modular and explainable. The explanations from these systems can then be used to produce feedback to aid in the learning process.

In this work, I present novel approaches for the recognition of location, movement and handshape that are components of American Sign Language (ASL) using both wrist-worn sensors as well as webcams. Finally, I present Learn2Sign(L2S), a chatbot based AI tutor that can provide fine-grained conceptual feedback to learners of ASL using the modular recognition approaches. L2S is designed to provide feedback directly relating to the fundamental concepts of ASL using an explainable AI. I present the system performance results in terms of Precision, Recall and F-1 scores as well as validation results towards the learning outcomes of users. Both retention and execution tests for 26 participants for 14 different ASL words learned using learn2sign is presented. Finally, I also present the results of a post-usage usability survey for all the participants. In this work, I found that learners who received live feedback on their executions improved their execution as well as retention performances. The average increase in execution performance was 28% points and that for retention was 4% points.

DEDICATION

This thesis is dedicated to my parents Dhama Nath Paudyal and Prabha Sharma, to my sister and brother-in-law Pallavi Poudel and Kushal Sharma and my loving wife Phan Tsan.

ACKNOWLEDGMENTS

I thank my advisor, Dr. Sandeep K.S. Gupta, for his guidance and support throughout my doctoral study. Most of the ideas and their implementation discussed in this dissertation were the result of stimulating discussions with him. My research, technical writing, and presentation skills have also refined greatly due to his continuous encouragement and feedback. I also thank Dr. Ayan Banerjee from whom I learned a lot while working very closely for all the work in this dissertation. I thank the rest of my dissertation committee members: Dr. Ihan Hsiao, Dr. Tamiko Azuma, Dr. Yezhou Yang for their expertise and guidance which greatly helped shape the quality of this work. Further, I thank the National Science Foundation (NSF), the National Institute of Health, the Graduate and Professional Students Association (GPSA) of Arizona State University (ASU), and the entire ASU ecosystem for supporting this work. I thank my colleagues Dr. Junghyo Lee, Prasanth Sukhapalli, Azamat Kamzin, Imane Lamrani with whom I had the pleasure of working closely on various projects. I also thank all the other members of the Impact Lab- Dr. Priyanka Bagade, Dr. Madhurima Pore, Vinaya Chakati, Javad Sohanker, Dr. Koosha Sadeghi, Bernard Ngabonziza, Sameena Hossain, Apurupa Amperayani, Anurag Agrawal for their friendship, guidance and support. I thank my mom, dad, wife, sister, and brother-in-law for encouraging me to start and constantly supporting so I could finish. I thank my sister Pallavi Poudel and brother-in-law Dr. Kushal Sharma for their constant support. Finally, I thank my wife Phan Tsan for her kind presence and encouragement that helped me cross the finish line.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

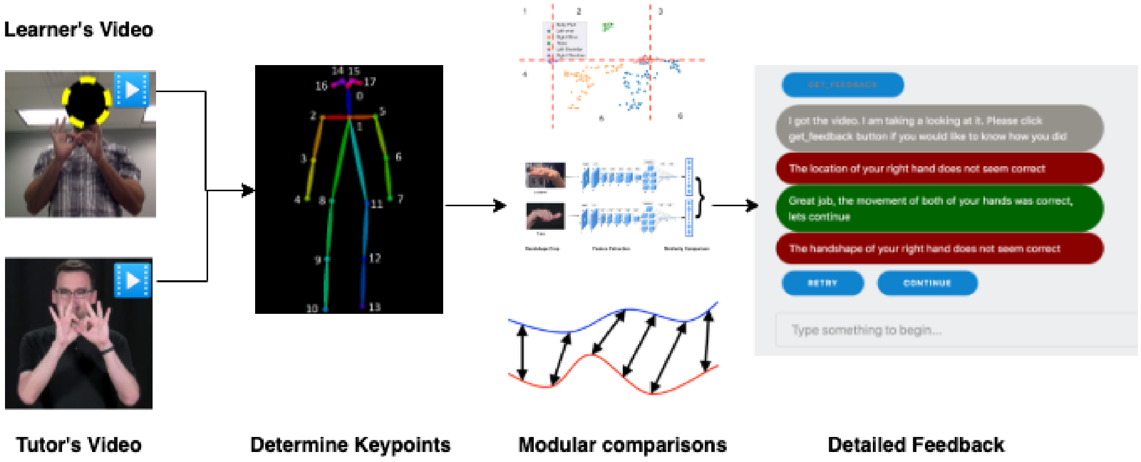
Chapter 1

INTRODUCTION

1.1 Background

Non-verbal communication is a big part of day-to-day interactions. Body movements can be a powerful medium for non-verbal communication, which is done most effectively through gestures. However, the human-computer interaction (HCI) interfaces today are dominated by text-based inputs and are increasingly moving towards voice-based control. Although speech is a very natural way to communicate with other people and computers, it can be inappropriate in certain circumstances that require silence, or impossible in the case of deaf people. Researches in sign language recognition (SLR) aim to bridge this gap by allowing computers to recognize gestures to facilitate HCI applications as well as to allow communication between the deaf and hearing people by recognizing and translating sign language gestures such as those of American Sign Language (ASL) Paudyal *et al.* (2016).

The aforementioned communication gap can also be bridged if the hearing population were to acquire sign languages. Learning a sign language like learning any other language is a difficult process. World Health Organization(WHO) estimates that around 466 million people worldwide have disabling hearing loss. This number is estimated to rise to 900 million by 2050 Organization (2018). Signed language are natural languages for deaf or hard of hearing people since only a fraction of the population with disabling hearing loss can benefit from cochlear implants. Other means of communication like lip-reading or writing are not natural for daily life conversations. Many family and friends of deaf or hard of hearing people also benefit from



I got the video. I am taking a looking at it. Please click get_feedback button if you would like to know how you did

The location of your right hand does not seem correct

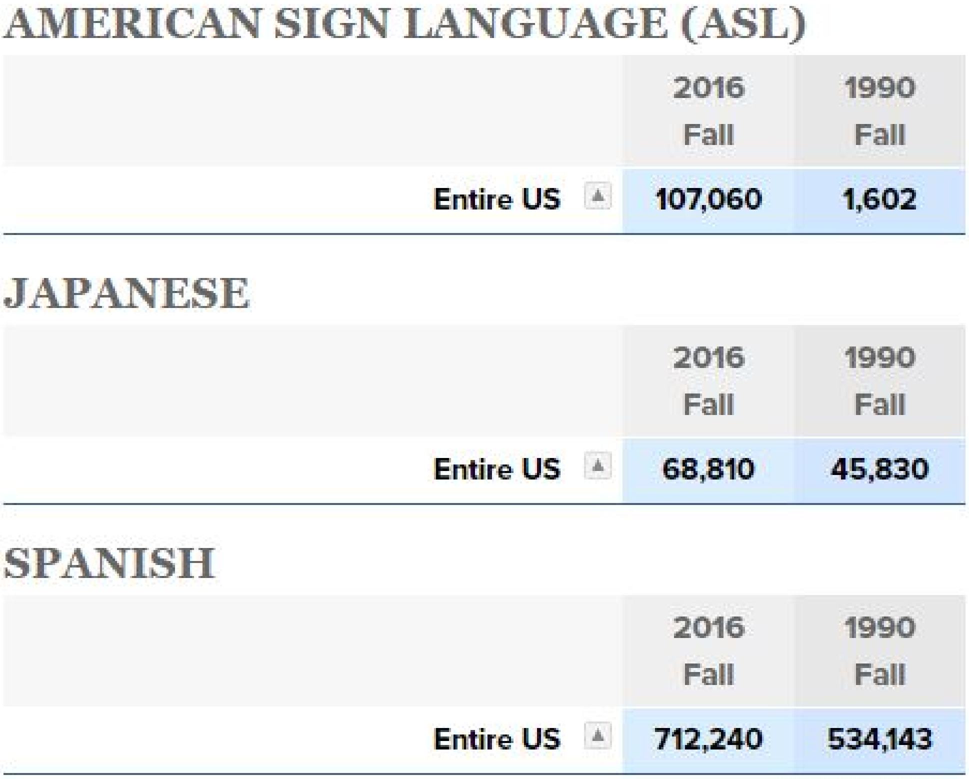
Great job, the movement of both of your hands was correct, lets continue

The handshape of your right hand does not seem correct

Figure 1.1: Learn2Sign System Design for Feedback.

being able to sign. Learning Sign Language is also a very popular choice to fulfill additional language requirements for colleges or high-schools. The Modern Language Association Association (2016) reports that the enrollment in American Sign Language (ASL) courses in the U.S. has increased nearly 6,000 percent since 1990 while that for other languages has been relatively constant as seen in Figure 1.2. This shows that the demand for sign language learning is increasing.

People may want to learn a sign language for various reasons including being born deaf in a hearing household or having friends/family who use a sign language. The process of learning can be aided by the use of technology that provides intelligent and effective feedback as exists for almost all major spoken languages, however, no such technology exists for learning signed languages. The ability to provide correct/incorrect decisions and ultimately feedback to learners depends first on the presence of an accurate SLR technique and then secondly on that technique being explainable enough to be able to give finer-grained feedback. Additionally, the techniques have to support a growing set of vocabulary and should function in ubiquitous environments. In this work, I propose novel ways to recognize various components of



AMERICAN SIGN LANGUAGE (ASL)

	2016 Fall	1990 Fall
Entire US ▲	107,060	1,602

JAPANESE

	2016 Fall	1990 Fall
Entire US ▲	68,810	45,830

SPANISH

	2016 Fall	1990 Fall
Entire US ▲	712,240	534,143

Figure 1.2: Enrollment Statistics for Various Languages from 1990 to 2016

signed languages. Then, I propose Learn2Sign(L2S) which is a feedback-driven way to learn American Sign Language using any Smartphone or any PC with webcam. I propose a novel three-tier feedback system and provide analysis on the design choices and their effectiveness as seen in Figure 1.1.

1.1.1 Gestural Languages

Some human communities communicate in entirely using gestures or have developed gestural systems of communication for occasions where speech is impossible or for some reason not permitted. Examples include religious communities sworn to silence, people working in noisy environments such as airplane traffic ground control, military personnel in covert operations or drills and most of all, deaf people. The multitude of sign languages invented and used by deaf people all over the world, is evidence that most of what is needed for human communication can be performed using gestures Corballis and Corballis (2002). Many of the signs that are used in signed languages throughout the world are iconic in nature i.e. they mimic the objects of the phenomenon they represent Valli and Lucas (2000). Even spoken languages retain manual gestures as an additional channel to emphasize or clarify meaning to varying extents throughout various cultures and languages Lieberman (2003). Thus, the ability to correctly perceive and understand gestural languages is of paramount importance for HCI interfaces of the future as well as for human-machine collaboration systems such as a sign-language tutor system.

1.1.2 American Sign Language

American Sign Language (ASL) is a natural language used primarily by members of the North American Deaf community Valli and Lucas (2000). It is one of the most widely used and well studied of the sign languages in the world. ASL is utilized for all instructions at Gallaudet University. While there is no universal sign language, for this work, ASL is chosen as a test-bed. This is primarily because of its widespread adoption and availability of literature, data and other resources for extensive study. ASL is also growing rapidly in popularity in many U.S. States as can be seen in Figure 1.2. ASL has a very distinct language structure and the Deaf culture associated with it, has qualified it to be a valid foreign language for college credits Compton (2016). For the scope of this work, the challenges for the recognition and tutoring of ASL words and phrases are considered.

1.1.3 Sign Language Recognition

Sign Language Recognition (SLR) can be understood as the usage of technology to interpret, translate or to identify sign language words, phrases or sentences. SLR systems vary with the complexities of supported vocabulary, the language(s) supported, with whether the system is meant to identify individual gestures or signs in isolation or if continuously signed words, phrases, and sentences are supported. Accordingly, the output of SLR systems may be in the form of isolated identified gestures which are equivalent to words or some phrases in the sign language supported, or a translation to another language: either a spoken language such as English or another sign language. SLR systems can be divided into several modules as seen in Figure 4.1. According to the techniques used for recognition, SLR systems can either be i) classification based where parameters of a model are learned by utilizing existing labeled data or ii) similarity-based techniques where no such model is learned, and the recognition is done by a comparison with known examples.

1.1.4 Sign Language Tutoring

The ideal environment for language learning is immersion with rich feedback Skehan (1998) and this is especially true for sign languages Emmorey (2001). Extended studies have shown that providing item-based feedback in CALL systems is very important Van der Kleij *et al.* (2015). Towards this goal, language learning software for spoken languages such as Rosetta Stone or Duolingo support some form of assessments and automatic feedback Stone (2016). Although there are numerous instructive books Rosen (2010), video tutorials or smartphone applications for learning popular sign languages, there hasn't yet been much work towards providing automatic feedback as seen in Table 1.2. We conducted a survey Lab (2018) of 52 first-time ASL

Table 1.1: Survey Results from 52 Users of the Application.

Category	Response	
Importance of Feedback	Yes: 96.2%	No: 3.8%
Movement Feedback	Correctness: 9.6% Colored Bones: 1.9% Sentence: 15.4% Correctness+Sentence: 9.6% All: 65.3	
Handshape Feedback	Circle around handshape: 63.5% Actual handshape: 36.5%	
Self-Assessment	Not helpful: 1.9% Somewhat: 5.8% Very helpful: 93.3%	
Expandability	Not helpful: 5.8% Somewhat: 7.7% Very helpful: 86.5%	

users (29M, 23F) in 2018 and 96.2 % said that reasonable feedback is important but lacking in solutions for sign language learning (Table 1.1).

For this work, I have implemented Learn2Sign (L2S), an interactive Artificially Intelligent(AI) tutor that uses feedback to teach sign language words and phrases to learners. The effectiveness of this approach is tested towards learning outcomes for American Sign Language (ASL) by using some commonly used words and phrases in ASL classroom instructions. Many research works point out to the positive relationship between feedback given through interaction and the learning performance of second-language learners Lightbown and Spada (1990); Mackey (2006). The ability to practice and receive feedback is also a positive aspect of immersive environments for second language learning such as study abroad programs and even classroom environment to some extent Magnan and Back (2007). Many software applications for spoken languages incorporate some form of feedback to help improve the pronunciation of learners Stone (2016). Applications like DuoLingo also provide interactive chat-bot like environments with feedback to increase immersion Vesselinov and Grego (2012). However, such applications are not available for learners of sign languages. Although the very recent SignAll System for Education has a system designed to aid classroom instruction for ASL, it requires a custom setup with multiple specialized cameras that require manual calibration as well as the use of colored gloves which make its usage limited and expensive for self-paced learning SignAll (2020). This is in part due to the inherent technical difficulties for providing feedback to sign language learners as discussed in Section 5.3.

Table 1.2: Some Applications for Learning ASL on Smartphones.

Application	Can Increase Vocab	Feedback
ASL Coach	No	None
The ASL App	No	None
ASL Fingerspelling	No	None
Marlee Signs	Yes	None
SL for Beginners	No	None
WeSign	No	None

The important role of practice and feedback for learners of sign language is also well-established Emmorey *et al.* (2009). However, currently to the best of our knowledge, there are no educational tools that exist that provide automatic analysis and feedback for sign language learners for self-paced learning. Huenerfauth et al. performed a Wizard of Oz. study and determined that displaying videos to students of their signing, augmented with corrective feedback messages results in better performances Huenerfauth *et al.* (2015). Other research points out to the fact that second language learners of sign languages usually made mistakes on the location, movement, handshape and/or facial expressions of the signs Rosen (2004). These also correspond to the phonetic components that make up signs in most sign languages including ASL Stokoe (2005). Of these, the location, movement, and handshape are generally acquired first Mayberry (2007), thus we choose these three as the concepts to give feedback on.

In theory, state-of-the-art video recognition systems Feichtenhofer *et al.* (2016), or sign language recognition systems Cihan Camgoz *et al.* (2018) could be utilized for recognition and towards providing feedback to sign language learners. However, the algorithms that these techniques use are designed to automatically extract low-level features and attributes and classify a given input among a variety of previously known classes. Although the features and attributes these systems extract are useful for the classification problem itself, they are rarely semantically meaningful for providing an explanation or feedback as to why a given input belonged to a certain class i.e. why a given execution was correct or not. Also, these classification techniques do not scale well to unseen vocabularies which would be very important for tutoring purposes where the vocabulary is constantly updated with the progress made by the learner.

1.1.5 Concept Learning

To provide the most benefit towards the learning outcome, Learn2Sign is designed to give feedback for the concepts of location, movement, and handshape of the signs. For instance if a signer performs the sign in the correct location, and with the right movement for both hands, but the shape of her right hand was not correct, Learn2Sign will send a feedback highlighting this so that the learner can focus on the incorrect aspects when she tries again as seen in Figure 5.5. This is achieved by using a recognition module that is a combination of several modular sub-modules each of which is designed to recognize a specific aspect of the signing. In our case, these concepts are the concepts of ASL in terms of location, movement, and handshape.

Feedback based learning is a richly studied topic, both in the field of Education Fukkink *et al.* (2011) and also in the field of computer-aided learning Hudson (2004). The effect of feedback can be directly measured by relating it to the learning outcomes. To measure the effectiveness of feedback on learning outcomes we designed a test and enrolled 26 University students with little to no prior ASL exposure. The participants used the interface to watch videos for 14 different ASL signs, and then performed a multiple-choice retention test as well as an execution test where their videos were collected and analyzed for correctness. These experiments show that providing automatic and interactive feedback had a notable benefit on the execution performance of the signers. Their overall retention of the signs also went slightly up with feedback. This follows intuition since the feedback that Learn2Sign provides is geared towards correct execution of the signs.

Additionally, there is also evidence that AI systems that provide feedback and interactivity tend to increase trust and subsequently adherence and usage Doran *et al.* (2017). Although a longitudinal study would need to be done to test adherence directly, the results of the post-usage survey we conducted on all the participants (summarized in Table 5.2) shows that the majority of participants (88.4%) preferred to use Learn2Sign over just watching videos for self-paced learning.

1.1.6 Explainable AI

According to the Oxford English Dictionary, the word explanation is A statement or account that makes something clear; a reason or justification for an action or belief. With the success of machine learning, many critical decisions are being derived using AI systems, there is an increasing need for trust, accountability, and clarity behind the decision making. At the same time, machine learning systems have become more and more complex and difficult to interpret. While the simpler machine learning algorithms like logistic regression can be interpreted using the weights as feature importance, more complex decision systems such as random forests or deep learning have no such interpretability. However, such systems can be designed in a way to be more comprehensible by humans if they provide some reasonings for their decisions Doran *et al.* (2017). Another technique of making decisions from AI systems explainable is to make or interpret the decision using human-understandable features of the input data. This can be achieved by breaking the overall decision problem into sub-problems that are themselves more human-understandable. In this way, a reasoning for the decision can be derived using the sub-modules that were utilized in the decision process. There is understood to be a trade-off between explainability and performance, perhaps because the simpler or modular explainable models are typically not as powerful as some of the more complex but less explainable models.

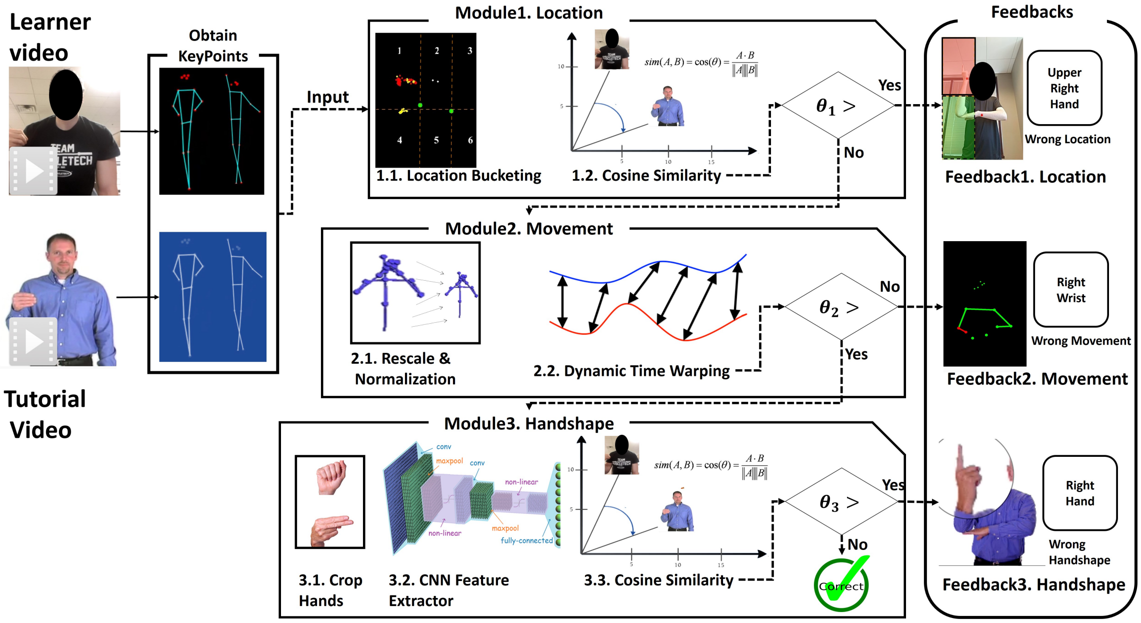


Figure 1.3: System Model for Feedback.

1.2 Significance

Recognizing sign language signs correctly is a required step for building tutoring systems. Sign language recognition systems by themselves are already very useful for accessibility. In this section, I discuss the significance of sign language recognition systems as well as those of sign language tutoring systems.

1.2.1 Sign Language Recognition

Sign Language Recognition systems can have a big impact on providing accessibility for the deaf and hard-of-hearing. Even without the realization of a full-blown machine-translation system, even a system that recognizes signs within a small specialized domain like medical emergencies, classrooms, airports etc. can be very beneficial Banerjee and Gupta (2015). Different approaches to SLR such as using cameras, specialized gloves, 3-D cameras, multiple-cameras or body sensors will have trade-offs in performance, usability, and functionality. However, pursuing all these avenues of research is critical to making progress in this field.

1.2.2 Sign Language Tutoring

Studies show that elaborated feedback such as providing meaningful explanations and examples produce a larger effect on learning outcomes than just feedback regarding the correctness Van der Kleij *et al.* (2015). The simplest feedback that can be given to a learner is whether their execution of a particular sign was correct. State-of-the-art SLR and activity recognition systems can be easily trained to accomplish this. However, to truly help a learner identify mistakes and learn from them, the feedback and explanations generated must be more fine-grained.

The various ways in which a signer can make mistakes during the execution of a sign can be directly linked to how minimum pairs are formed in the phonetics of that language. The work of Stokoe postulates that the manual portion of an ASL sign is composed of 1) location, 2) movement and 3) hand-shape and orientation Stokoe (2005). A black box recognition system cannot provide this level of feedback, thus there is the need for an explainable AI system because feedback from the system is analogous to explanations for its final decision. Non-manual markers such as facial expressions and body gaits also change the meaning of signs to some extent but they are less important for beginner-level language acquisition, so these will be considered for future work.

Studies have also shown that the effect of feedback is highest if provided immediately Van der Kleij *et al.* (2015), thus feedback systems should be real-time. The requirement for immediate feedback also restricts the usage of complicated learning algorithms that require heavy computing Cihan Camgoz *et al.* (2018); Lee *et al.* (2017) and extensive training. The usability and usefulness of applications is enhanced if learning is self-paced, learners are allowed to use their own devices, and the learning vocabulary can be easily extended. However, current solutions for SLR require retraining to support unseen words and large datasets initially. To solve these challenges, we designed Learn2Sign(L2S), a smartphone application that utilizes explainable AI to provide fine-grained feedback on location, movement, orientation and hand-shape for ASL learners. L2S is built using a waterfall combination of three non-parametric models as seen in Figure1.3 to ensure extendibility to new vocabulary. Learners can use L2S with any smartphone or computer with a front-facing camera. L2S utilizes a bone localization technique proposed by Papandreou *et al.* (2017) and various carefully chosen similarity measures for movement and location-based feedback and a light-weight pre-trained Convolutional Neural Network (CNN) as a feature extractor for hand-shape feedback.

Having an automated system for new students to learn, practice and obtain feedback in a self-paced manner using their own devices will be paramount to scaling the beginner level classes as well as speeding up learning.

1.3 Components

There is a benefit to breaking down the task of sign language recognition into conceptual modules. In this study, we consider the location, movement, and hand-shape modules separately. This not only helps recognition by allowing the usage of specialized recognition modules, but also allows for the recognition algorithm to be modular and explainable. For instance, for the tutoring application, an ability to provide detailed feedback is desired. This feedback does not make the learning experience immersive, but also allows focussed repetition and thus enhances learning. The scope of the feedback considered in this study is limited to solving the technical challenges for facilitating feedback-driven language learning for beginner level sign languages. This is equivalent to what a second-language learner would encounter in a single semester where obtaining the necessary vocabulary is the primary goal. Even for this somewhat narrow scope, there are many unsolved technical challenges. Some of these are how to ensure support for growing vocabulary, how to ensure accurate and real-time recognition and feedback, and how to test for appropriate feedback mechanisms. While similar feedback mechanisms can be adapted to aid other gestural language learning or even advanced aspects of sign language learning this is out of the scope of this work. The results of Mayberry et. al. suggest that the concepts of correct location, movement, and handshape are acquired first Mayberry (2007) at the beginner level and other more advanced correctness concepts such as facial expressions and body gaits are acquired as the learner begins to develop fluency. Thus, the feedback provided by Learn2Sign for this work is limited to textual feedback provided as brief sentences for the location, movement, and handshape of the signer.

1.3.1 Recognition of Location

The goal of the recognition module for location is to isolate the primary position of articulation of the sign. This position has to be determined relative to the body of the signer due to variance in frame-size, sitting/standing posture and distance to the camera. Another challenge is to isolate the primary location of each of the articulators in the presence of movement since many sign language signs incorporate some kind of movement The correct location of a sign can either be hard-coded by querying a list of 'supported' signs which is equivalent to learning a model that identifies certain signs, or this can be done during recognition time by comparison to a certain gold standard. The latter approach is employed in this work. After recognition of the correct location, the goal of the tutoring module would be to provide feedback if the location of either of the hands is incorrect. The usage of the signers' own body as a frame of reference rather than the field of visibility of the camera helps to mitigate some of the challenges towards this. The signer's shoulders are taken to be fairly stable for the duration of the execution of a sign and correctness of the location of each hand is based on its relative position of each of the hands to each of the shoulders. This is explained in more detail in Chapter 2. If the location module detects that either of the hands' location was not correct for the duration of the signing, corrective feedback for location can be given. The feedback can be in the form of a text or a color highlight of the part of the screen. If the correctness of the location of the hand can be known, then various types of feedback can be given to the learner based on further usability studies. In this work, we give a detailed treatment for recognition of the location of the signing using a video-feed in Chapter 2. Recognition of location using body-worn sensors can be done accurately using an initial calibration and thus is not discussed in much detail in this work.

1.3.2 Recognition of Movement

Hands are the primary articulators for signed languages. The Movement module is responsible for checking that the movement executed by each hand of the learner is sufficiently close to that of a teacher or a known model of a sign. Training a model of all known signs suffers from the inability to scale to unseen vocabulary. Thus, for scalability reasons and to support a tutor application a similarity-based measure is chosen.

This measure of similarity has to be sufficiently robust to delays in starting and the speed of signing. It also has to account for the variance in distance between the signer and the camera being used as well as the distortions introduced by the difference in frame-size and resolutions of the camera. The use of a 2D camera that is available in most devices today instead of a 3D camera or multiple cameras for depth inference further complicates the comparison of movement trajectories. Thus, a distance estimation technique that is reasonably robust to these conditions is used along with a smoothing and normalization procedure that will be discussed in Chapter 3. In addition to the requirement of an appropriate similarity measure, detailed feedback also requires the computation of an alignment between the movement trajectories. This additionally allows us to reasonably estimate which frames to utilize for handshape comparisons. The distance measure can be used as a threshold for deciding if the learner's trajectory was close enough to that of the teacher. The algorithm for comparison results also in a point-by-point optimal alignment for the two trajectories. This can be used to provide much more detailed feedback to the learner if desired.

In this work, I present a body-worn armband based approach as well as a webcam-based approach to recognize signs that differ in movement. Details on the approach, methodology and results can be found in Chapter 3.

1.3.3 Recognition of Hand-Shape

Signs in sign languages can differ in meaning due to variation in the shape of the hand. Indeed other forms of gesturing, dances and body language also use handshapes to convey meaning. In this work, I present a novel algorithm to identify handshape based on Electromyogram (EMG) sensors coupled with Inertial Measurement Unit (IMU) sensors that are included in a wireless armband. I also present a fast and efficient technique of comparing handshapes between videos by using deep feature extraction. Details on both of these approaches as well as evaluation results are presented in Chapter 4. To provide handshape based feedback, the shape that the learner's hands assume must be compared to that of the teachers. This comparison must be done at several stages of a sign since a single ASL sign can have various handshapes that transition from one to another. Also, the execution of the teacher and that of the student cannot be assumed to be in perfectly aligned in time. This misalignment can be caused by several factors such as delayed start, or difference in speed of signing. After correct alignment, a well-measured crop of the hands being compared is also required to avoid noisy comparisons and to increase accuracy. Additionally, the handshape comparison should be robust to changes in brightness conditions, skin-tone, orientation differences etc. After all this information is available, a detailed feedback based on which frame the handshape was not correct for can be used in the feedback provided. Many other types of feedback mechanisms can be designed. For the user study, to avoid any confounding factors, a simple text message of whether the handshape was correct or not is provided for each hand. Then the replay of the correct handshape as executed by the teacher is shown.

1.3.4 Learner Feedback

The various recognition modules can be combined to determine if a particular sign execution was correct or incorrect. Appropriate comparison techniques and thresholding mechanisms can be used to determine the correctness of an execution compared to a gold-standard. At this point, these results can be either used to compose a recognition algorithm that simply outputs correct if all of the modules' results were positive or outputs negative if either one of the modules were not correct. Since recognition on all the modules is dependent on a threshold that can be tuned during training, the amount of False negatives can be balanced with False positives to provide a desired level of 'Feedback Sensitivity'. An appropriate level of feedback to be given to the learner can thus be determined. A waterfall approach can be taken as seen in Figure 1.3 where only Location Feedback is provided first, then if the location of the execution is correct, a movement feedback can be provided. Finally, the handshape feedback is provided if and only if both the location and the movement were already correct. Conversely, all the possible feedback types can also be shown concurrently as seen in Figure 1.1. This feedback mechanism can also be personalized to cater to different learning patterns of learners.

Chapter 2

RECOGNITION OF CORRECT LOCATION

2.1 Introduction

For SLRs, it is important to keep track of the location of both the hands as they are the primary articulators in most sign languages. For isolated sign recognition, the user‘s body remains relatively stationary throughout the video. Thus, the location of the primary articulators can be computed relative to the signer‘s own body. This type of relative localization is preferrable to absolute localization with respect to pixel coordinates in the frame, because no assumptions on the size of the frames or the position of the learner in the frame have to be made. In other words, using a location relative to the signer’s body abstracts away the need to account for differences due to camera resolutions or the distance of the learner from the camera. This technique works well for one-handed signs as well as for two-handed signs. However, for this to work, other parts of the learner’s body have to be localized as well. This is done effectively by using landmark detection algorithms for body pose which are also known as pose detection algorithms. After the relative location of each of the hands is known for each of the frames, a comparison between the positionings for different executions can be performed using vector comparison methods like cosine similarity to get a measure of execution similarity with respect to the location.

2.2 Problem Statement

The problem of determination of the correct location of a sign is that of 1) localizing the primary articulators i.e. the hands 2) tracking them throughout the duration of the signing video and 3) determining the primary location of signing. During the duration of the execution of a sign language word, the position of one or both of the hands can be stationary or mobile. The movement during a sign can be part of the sign or it may be the setup process of moving the hands to the correct positions.

2.3 Significance

The signs in sign languages are composed of a finite number of discrete, meaningless and contrastive sub-units: 1) Location 2) Movement and c) Handshape Stokoe *et al.* (1970). Morphemes, or units of meaning, are formed by a combination, of such sub-units and any substitution may result in a different morpheme. What this means practically is that if the only the location of a particular word was modified, while keeping the movement and handshape the same, it can result in the change of meaning. This is exemplified in the case of the pairs MOTHER vs. FATHER or STOMACH-ACHE vs. HEAD-ACHE in ASL. In sign languages, each free morpheme is restricted to a single major body area- such as the head, torso, or the dominant and non-dominant hands Battison (1978). In signs that have a movement component to them, a sign may consist of up to two settings for a single sign. For some signs such as DEAF, both times the contact is in the head area, first near the ear and then near the mouth. For other signs, that do not involve contact, researchers are divided whether there are two distinct major locations or only one Sandler (2012). Nonetheless, the location of signing plays an important role in distinguishing otherwise similar signs. Thus, the identification of the correct location is paramount to correctly recognizing a sign.

2.4 Related Work

Identifying the position of hands at any given instance can be achieved by various techniques. In this section I review two such approaches and some important related work for them.

2.4.1 Hand Tracking

Traditionally, Hand tracking has been done via the usage of wrist-worn sensors Paudyal *et al.* (2016), data-gloves Dipietro *et al.* (2008), 3D cameras or wrist-worn cameras Kim *et al.* (2012). Hand tracking using either 3D or 2D videos can be done using a hand kinematic model approach or a model-free approach Oikonomidis *et al.* (2011); Frati and Prattichizzo (2011). The advantage of a hand-model based approach is the increase in accuracy due to the inclusion of body joint constraints, while model-free approaches are robust are more useful for reinitialization. If ease of usage is considered, smaller smartwatch-like sensors or pervasive webcam like devices are preferrable Paudyal *et al.* (2019c).

2.4.2 Pose Estimation

For this work, we require an estimation of human pose, specifically the estimates on the location of various joints throughout a video, known as keypoints. There have been several works towards this goal Tome *et al.* (2017); Chen and Ramanan (2017); Bogo *et al.* (2016); Sarafianos *et al.* (2016); Tompson *et al.* (2014); Chen and Yuille (2014). Some of these works first detect the keypoints in 2D and then attempt to 'lift' that set to 3D space while others return the 2D coordinates of the various keypoints relative to the image. To fulfill the requirement to use pervasive cameras, we did not focus on the approaches that utilize depth information such as Microsoft

Kinect Pedersoli *et al.* (2014). Thus, we utilized the pose estimates from a Tensorflow JS implementation of a model proposed by Papandreou et al. Papandreou *et al.* (2017) which can run on devices with or without GPUs (Graphical Processing Units).

2.5 Technical Challenges

While utilizing existing work on Hand Tracking or Pose Detection is straightforward, creating a robust and usable system to account for all the variations possible in real-world usage is challenging. In this section, I outline the technical challenges associated with determining the correctness for the location of signing with the end-goal to develop systems that are scalable, real-time, and robust to variations in the real world.

2.5.1 Challenge 1: Determination of Frame of Reference

Due to differences in the size of the frame, the distance from the camera, resolution of the camera feed etc, it is difficult to determine how to compare a location in one video to that of the other. Unless environmental restrictions are in effect, the only way to mitigate these issues is to determine the location of the articulators relative to other body parts. In camera-based approaches, this means localizing other joints or keypoints in the body. For sensor-based approaches, in the absence of other markers either an initialization phase is required or a rigid position for the sensor relative to the body part has to be assumed.

2.5.2 Challenge 2: Granularity of Identification

The phonetic realization of signs is often affected by co-articulation among other factors. Even in a teacher-student scenario, a perfect replication is not expected. Thus, the division of the signing space around the body has to be done in a way to allow some variation for the correct location, without accepting an altered morphology.

2.5.3 Challenge 3: Determination of Similarity Measure

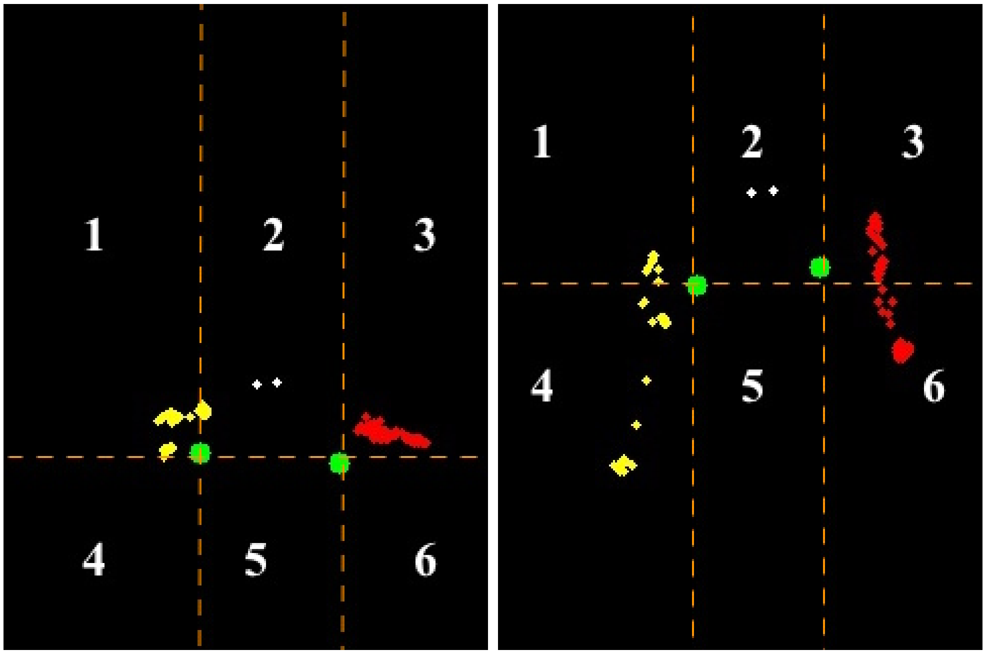
Traditional distance measures like Euclidean distance assume proximity in scale. However, for practical applications where the length of signing may differ due to the speed of the signer as well as the frame-rate of the captured signals, this proximity is often violated. Thus an appropriate distance measure should be utilized that can take this into consideration.

2.5.4 Challenge 4: Movement in Signs

Signs in most sign languages consist of movement within a sign. This becomes even more paramount when continuous signing is considered. Thus, determining the primary location(s) for a sign in the presence of movement is challenging.

2.6 Approach and Methodology

Based on the key-point coordinates of both the left and right shoulders, six location buckets are established. This is indicated by the two vertical lines passing through both the shoulder points in Figure 2.3. These lines along with a horizontal line connecting both the points separating the upper and lower buckets serve as bucket boundaries. For every recorded frame in the video, the location of both the left and right hands are tracked and the count for the corresponding bucket is incremented. Finally, we obtain a 1x6 vector for both the hands. A similar process is followed for the tutor‘s videos and the correctness of location for each hand is determined by thresholding on a cosine similarity score.



(a) TIGER mostly in Bucket 1 for Left Hand.

(b) DECIDE in Buckets 3 and 6 for Right Hand.

Figure 2.1: Automatic Bucketing for Location Identification for Varying Distances from the Camera. Left Wrist(Yellow), Right Wrist(Red), Eyes(White), Shoulders(Green).

2.6.1 Keypoint Estimation

Pose estimation is generally done through estimating the location of various salient points in the body such as the hands, the shoulders, or the torso as seen in Table 2.1. These salient body parts that are tracked frame-by-frame throughout a video are generally referred to as keypoints. Robust estimation of keypoint is a necessary step for localizing a sign relative to a signers‘ body and determining the type of movement. The keypoint locations and the confidence levels for them are also used for initial calibration, starting the countdown for recording and for determination of handshape crops as will be discussed in Chapter 4. We utilize a variant of Residual

Neural Network Tai *et al.* (2017) from Tensorflow.js Smilkov *et al.* (2019) called PoseNet Papandreou *et al.* (2017). PoseNet can be used for single as well as multiple pose detection, which means that it can detect many people in the same frame. However, for our purposes, the single pose detection module was sufficient. The ResNet50 powered model was configured with optimum default values considering the performance speed for Output stride(32), image input resolution(257) and enough quant bytes(2) to perform weight quantization. For each frame in a video or continuous stream, PoseNet returns 17 different keypoints throughout the body. A sample of this is presented in Table 2.1. Among these, the keypoints for eyes, nose, shoulder, elbows, and wrists were utilized. The eyes and the nose keypoints along with those for the wrists are used initially before the learner begins to record their execution. Along with the 'x' and 'y' coordinates, PoseNet also returns a confidence score between 0 and 1 for each keypoint for every frame. A threshold of 0.6 was utilized to calibrate the webcam positioning and to verify the learner was in range and visible. After verification of about 60 frames, a 3 s timer to begin recording is triggered. As a learner is recording their execution the keypoints for every frame are recorded to be used for a comparison to the tutor for that sign. The keypoints estimated for 5 different body parts throughout the videos for signs 'Spanish' and 'Nice to meet you' are plotted in Figures 2.3 and 2.2 respectively.

2.6.2 Bucketing

Based on the key point coordinates of both the left and right shoulders, six location buckets are established. This is indicated by the two vertical lines passing through both the shoulder points in Figure 2.3. These lines along with a horizontal line connecting both the points separating the upper and lower buckets serve as bucket boundaries. For every recorded frame in the video, the location of both the left and

Table 2.1: Confidence Score and Frame (x,y) Coordinates for some Keypoints

Part	Score	x	y
Nose	0.9907	138.7879	147.7989
Left Eye	0.9958	142.4354	136.857
Right Eye	0.9991	126.7588	135.5591
Left Shoulder	0.8325	180.0198	196.2646
Right Shoulder	0.7876	138.7879	195.5847
Left Wrist	0.6844	198.9818197	223.4856
Right Wrist	0.1564	1.9084	211.0473

right hands are tracked and the count for the corresponding bucket is incremented. Finally, we obtain a 1x6 vector for both the hands. A similar process is followed for the tutor‘s videos and the correctness of location for each hand is determined by thresholding on a cosine similarity score. Here a cosine similarity measure is preferred over other measures like euclidean distance since we do not make any assumptions on the number of frames recorded for the tutor as well as the learner. A value of 0.6 was determined to be an appropriate threshold by experimentation.

2.6.3 Similarity Measure

The cosine similarity measure was selected for location-based similarity comparison of two videos. Cosine similarity is preferable to other common measures such as manhattan similarity or euclidean similarity since a relative count in the different buckets has meaning rather than absolute counts. In other words, if two videos had a different number of frames due to differences in signing speed or frame-rate of

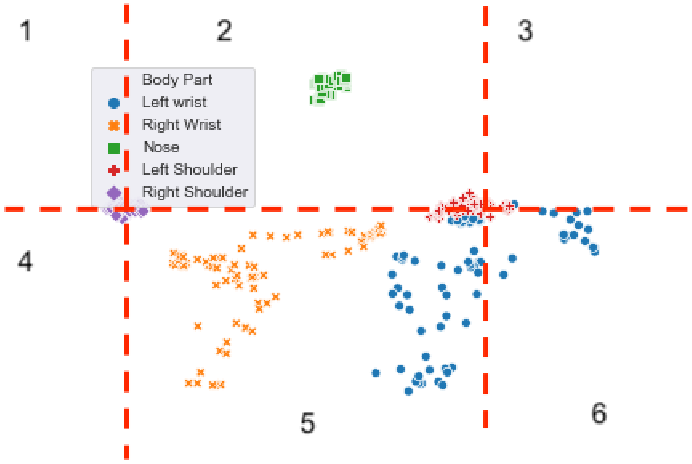


Figure 2.2: The Right Wrist stays in Bucket 5 for Sign ‘Nice to meet you’.

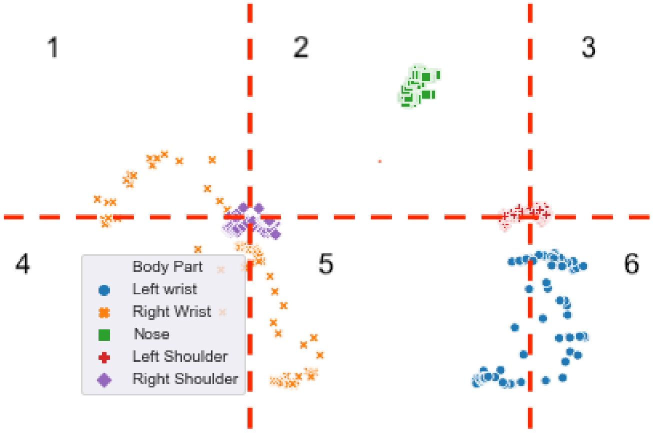


Figure 2.3: The Right Wrist goes between Buckets 1 and 5 for Sign ‘Spanish’.

the capturing camera, but both videos had the right-hand location in bucket 3, the counts for bucket 3 would still be relatively high. This case is handled well by the cosine similarity measure. In other words, a cosine similarity measure is preferred over other measures like euclidean distance since we do not make any assumptions on the number of frames recorded for the tutor as well as the learner. A value of 0.7 was determined to be an appropriate threshold by experimentation.

2.7 Results and Evaluation

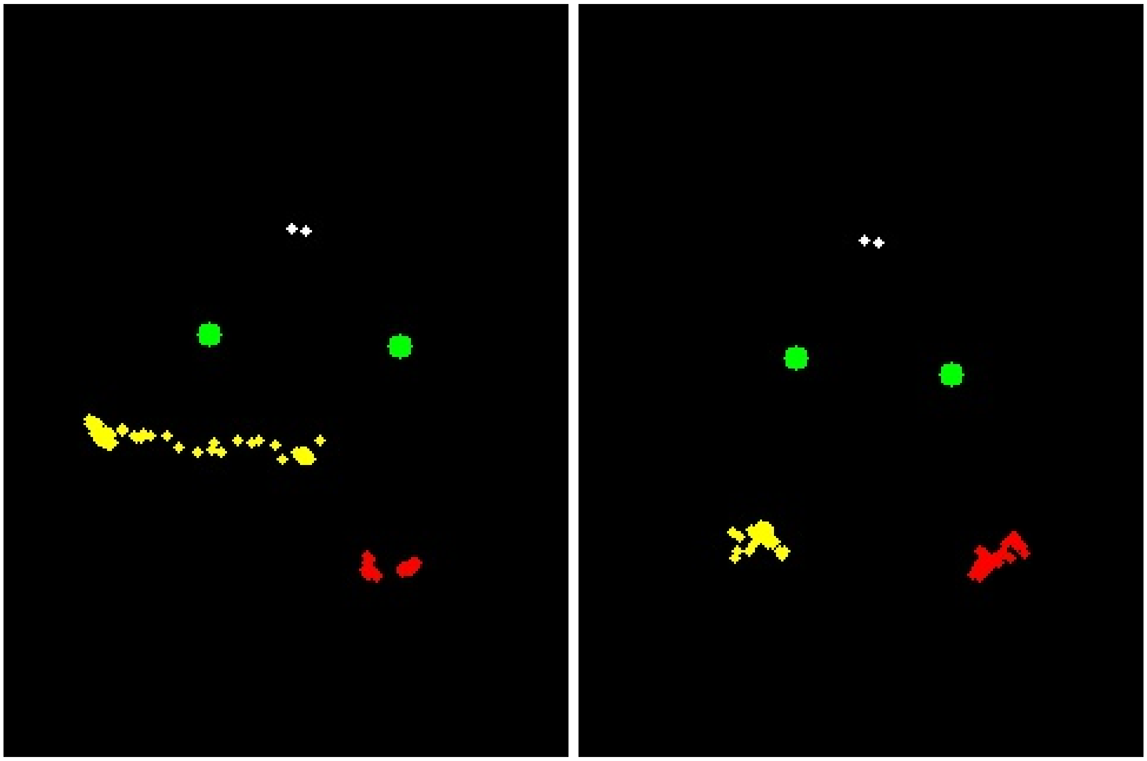
After preprocessing for normalization and outlier removal, the location-based identification was executed. The precision and recall rates for each of the 25 signs in the dataset are shown in Table 4.5. An average true positive rate of 91.40 % across all users was achieved.

To obtain the results, data collected from one learner was selected at random and served as the 'tutorial' dataset. Then for each sign for each user in the test dataset, we compared against the corresponding sign in the tutorial dataset. The ideal behavior of the application when provided with a sign that is correct in the location modality to a tutorial video, should be accepted. The high recall rate for location-based identification reflects this.

A subset of signs will have considerable overlap in the locations where the signs were executed. For instance in Figure 2.4 the signs for 'AND' and 'HERE' share much of the location space while they will differ in other modalities like movement and handshape. Thus, the evaluation is done jointly with the handshape and movement components discussed in Chapters 3 and 4.

The combined results are summarized in Table 4.5. To obtain the results, data collected from one learner was selected at random and served as the 'tutorial' dataset. Then for each sign for each user in the test dataset, we make comparisons against the corresponding sign in the tutorial dataset. The location module had an overall recall of 96.4% and a precision of 24.3%. The lower precision is because many signs in the test dataset had similar locations. We performed a test comparing only the signs 'LARGE' to the sign 'FATHER' and both the precision and recall were 100 %.

Another experiment was done for signs that differ considerably in the location where the signs are executed as seen in Figure 2.1. The Table 2.2 shows that the



(a) AND

(b) HERE

Figure 2.4: Eyes(White), Shoulders(Green), Left Wrist(Yellow) and Right Wrist(Red) Locations.

location-based identification of disparate signs is indeed effective when the two signs being compared vary on location. The high precision rate shows that when a sign executed in the incorrect location the location model identifies it as being incorrect by the location model.

Table 2.2: Signs Differentiable by Location

Sign1	Sign2	Precision%
Here	Tiger	100.0
And	Large	90.70
About	Here	83.33
Hospital	Father	100.0

Chapter 3

RECOGNITION OF CORRECT MOVEMENT

3.1 Introduction

Similar to location, ASL Signs are also differentiated by their movement. Capturing the movement of the hands with respect to time is required for making comparisons. It is not meaningful to utilize euclidean distance between frame by frame keypoint locations as a measure of similarity because no assumptions are made on the size of the frames or on the speed of execution by the signer. A frame-by-frame distance would also not account for a delayed start or an early finish. Thus, a technique called dynamic time warping is utilized to account for issues with synchronization, difference in speed and delayed start/stop Berndt and Clifford (1994). The performance of dynamic time warping is boosted by the usage of z-normalization on the time-series Ratanamahatana and Keogh (2004) as seen in Eq 3.1. Z-normalization also accounts for the difference in the size of the frame, the distance of the learner from the camera and the size of the learner relative to the tutor to some extent. DTW tries to get an optimal match for every data point in the sequence with any data point of the corresponding sequence. A data point may have more than one match as long as the index mapping between elements of both the sequences must be monotonically increasing. For ensuring real-time performance Learn2Sign uses an alternate version of DTW called Fast DTW Salvador and Chan (2007). In this approach, the traditional brute force dynamic programming approach is avoided. In place, the sequence is sampled down to a much lower resolution and a warp path is decided. After this, the warp path is projected to higher resolution time-series incrementally

Table 3.1: Summary of ASL Users' Survey on a Continuum from Very-unlikely (0) to Very-likely (5).

Question	**1**	**2**	**3**	**4**	**5**
Interest in trying solutions for gesture based HCI	0	0	1	3	9
Interest in using ASL for gesture based HCI	0	0	3	4	6
Likeliness to use a system that requires 3 trainings instances	0	0	3	1	9
Importance of non-invasivenss	0	1	3	2	7
Likeliness to use if wait time 0-3 s	0	1	2	4	6
Likeliness to use if wait time 3-7 s	1	1	4	5	2
Likeliness to use if wait time 7-10 s	4	4	5	0	0
Likeliness to try a prototype	0	1	3	4	5
Likeliness of the use of this system by a native speaker for communication	0	1	1	5	6

until it reaches the full resolution. Finally, a distance measure is computed based on this optimal warp path. There is a trade-off between accuracy and performance time when selecting the warping window size. For our purposes, a warping window size of 5 was selected for the user-studies. This selection also has other implications such as how much lag or differences in signing speed is tolerated and thus must be chosen carefully Chen *et al.* (2012).

$$z - normalized = \frac{x - \mathrm{mean}(X)}{\mathrm{sd}(X)} \tag{3.1}$$

3.2 Significance

3.3 Related Work

Most existing work on sign language recognition utilize image/video-based recognition techniques while some other utilize data gloves Liang and Ouhyoung (1998), Kuroda *et al.* (2004), Fang *et al.* (2004), Kim *et al.* (2015), Praveen *et al.* (2014).

Although a high accuracy is achieved in the aforementioned data-glove based approaches, one major drawback is that the system is wired and invasive as it interferes with the day-to-day activities of the user. According to Fang et al. Fang *et al.* (2004), commercially existing data-gloves are also too expensive for large scale deployment. The system that we propose, utilizes Bluetooth enabled sensors that appear in wearables such as smart-watches, and are thus pervasive and non-invasive. They can also be easily worn underneath a shirt to make them less conspicuous.

The image and video-based recognition systems Starner *et al.* (1998); Mitra and Acharya (2007); Kelly *et al.* (2009); Koller *et al.* (2015); Bilal *et al.* (2013) use either annotated fingers, color bands etc, or just plain raw images and video from users to translate gesture-based communication. Some studies also utilized multiple cameras Vogler and Metaxas (1998) to enhance the accuracy by some margin. However, image and video-based systems are dependent on the presence of cameras in the right positions and are affected by background colors. Another drawback to these systems is that image and video processing is computationally expensive which makes the recognition system's response time slower. Also, heavy dependence on Hidden Markov Model (HMM) based methods makes training the system cumbersome and data-intensive. Unlike these approaches, using portable wrist-watch like sensors to detect gestures is far less invasive and a template-based recognition approach does not require many training data-sets. The computational complexity also decreases as there is no need to deal with data-intensive images and videos, this allows the system to be real-time as well as environment independent.

Li et al. Li *et al.* (2010) has done a study on combining accelerometer and EMG sensors to recognize sub-word level gestures for Chinese Sign Language and Chen et al. Chen *et al.* (2007); Zhang *et al.* (2011) show that combination of multiple sensors helps to increase the recognition accuracy. Following this path, we add the

EMG measurements from eight built-in pods in the Myo device to get information that can be leveraged to detect subtle finger movements and configurations which are essential for detecting certain signs and distinguishing them from others. The orientation sensors help to distinguish between signs that have a similar movement and muscle patterns but different rotational primitives. Accelerometer sensors detect movement and position of the hands. The combination of all these sensors into one commercial-grade, wireless sensor has given the ability to deploy gesture and sign-language recognition abilities in a far more practical and user-friendly way. Bluetooth technology for the transfer of data makes the system wireless and easy to use. The overall accuracy is also increased due to the use of all three types of signals.

Thus by coupling pervasive technologies with simple algorithms, we achieve high accuracy rates and great mobility. Starner et al. propose a solution that might be user and language-independent as possibilities, however, an implementation is not provided Starner *et al.* (1998).

3.4 Problem Statement

The main challenge of movement recognition for this purpose is to develop a method to store, retrieve and most importantly match gestures effectively, conclusively and in real-time.

There are various known methods for gesture recognition and time-series analysis, such as Support Vector Machines, Linear Correlation, Hidden Markov Models, and Dynamic Time Warping Morency *et al.* (2007), Huang *et al.* (2009), Chen *et al.* (2003), Corradini (2001). Two classes of methods are usually used for time-series analysis and data-matching: One is the learning-based, in which models are developed and stored and the other one is the template-based, in which the new time-series data is compared to a saved template. Although learning-based models were tried, the

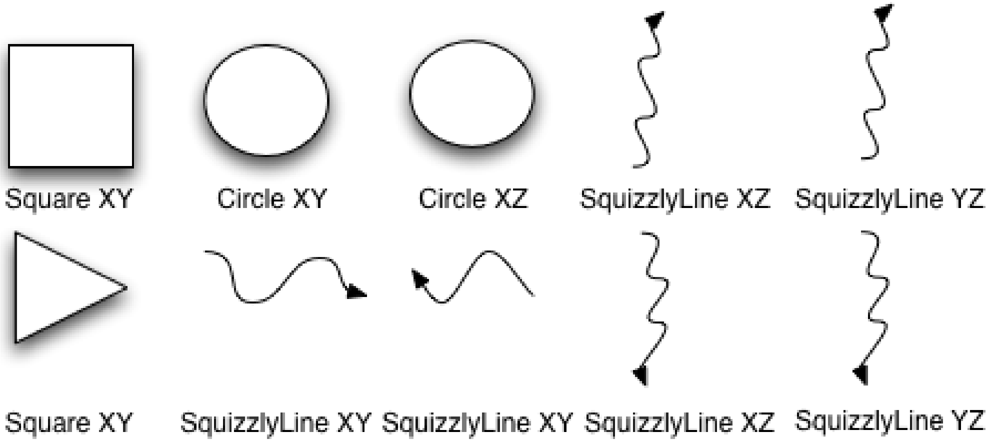


Figure 3.1: User Trained Gestures.

template-based models were preferred to allow user-driven training with the least number of repetitions and to make sure the system is usable in real-time. A very compelling advantage of utilizing a user-trained approach is that the gesture space can start small and can gradually grow as it is being used. This will also give the users an ability to discover a pattern of expression, train their system and use that to communicate seamlessly even with users of other systems or languages. Due to the flexibility of the system, a user who is well versed in a particular gesture-based system such as ASL, can choose to train the system according to the specifications of that language. She can then add custom gestures or shortcuts such as seen in Table 3.1 to that system if desired.

Dynamic Time Warping Morency *et al.* (2007), Huang *et al.* (2009), Chen *et al.* (2003), Corradini (2001), Paudyal *et al.* (2018). Two classes of methods are usually used for time-series analysis and data-matching: One is the learning-based, in which models are developed and stored and the other one is the template-based, in which

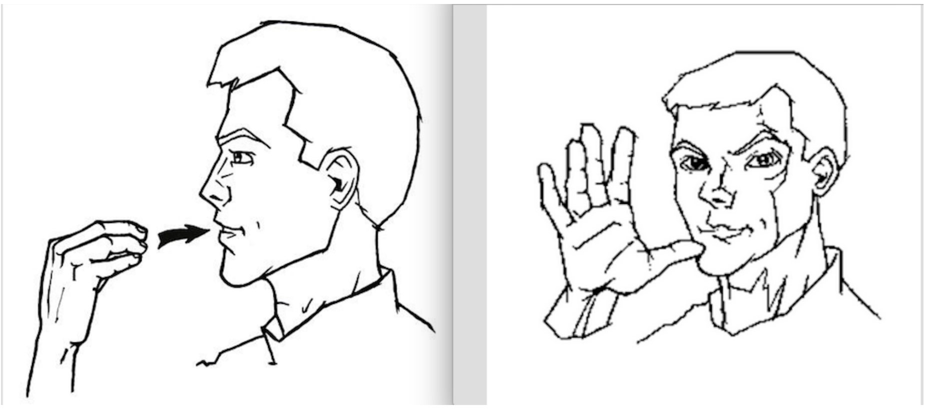


Figure 3.2: Gesture for ‘Mom’ Vs. ‘Eat’.

the new time-series data is compared to a saved template. Although learning-based models were tried, the template-based models were preferred to allow user-driven training with the least number of repetitions and to make sure the system is usable in real-time. A very compelling advantage of utilizing a user-trained approach is that the gesture space can start small and can gradually grow as it is being used. This will also give the users an ability to discover a pattern of expression, train their system and use that to communicate seamlessly even with users of other systems or languages. Due to the flexibility of the system, a user who is well versed in a particular gesture-based system such as ASL, can choose to train the system according to the specifications of that language. She can then add custom gestures or shortcuts to add to that system if desired.

3.4.1 American Sign Language Specifics

Signs in ASL are composed of several components: the shape the hands assume, the orientation of the hands, the movement they perform and their location relative to other parts of the body. One or both hands may be used to perform a sign, and facial expressions and body tilts also contribute to the meaning. Changing any one of these may change the meaning of a sign Johnston and Schembri (2007), Bahan (1996), Stokoe *et al.* (1970), Costello (2008b). This is what gives sign-language the immense power of expressibility, however this also means that multi-modal information has to be processed for sign-language recognition.

Orientation is the degree of rotation of a hand when signing. This can appear in a sign with or without movement. The gyroscope sensors provide rotation data in three axes, which is utilized in recognizing orientation Lee *et al.* (????). Data from EMG sensors is utilized in detecting muscle tensions to distinguish hand-shapes, which is another very important feature of the signs.

Location, or *tab*, refers to specific places that the hands occupy as they are used to form words. ASL is known to use 12 locations excluding the hands Tennant and Brown (1998). They are concentrated around different parts of the body. The use of a passive hand and different shapes of the passive hand is also considered valid locations. Movement or sig, refers to some form of hand action to form words. ASL is known to use about twenty movements, which include lateral, twist, flex, opening or closing Stokoe *et al.* (1976). A combination of accelerometer and gyroscope data is instrumental in distinguishing signs based on movement and signing space.

Sign language is a very visual medium of communication, the system that this paper proposes however tries to capture this visual information, in a non-visual manner to ensure pervasiveness. Thus, multi-modal signals are considered to get as close of an estimate as possible.

3.4.2 Problem Description

The primary task of the system is to match gestures. Since multi-modal signals are being considered, they are compared individually to the signals stored in the database and the results are combined. The form of the input data consists of an

Algorithm 1 Problem Description.

```
procedure MATCH GESTURES(test)
    for (i in 1 to database_size) do
        $template \leftarrow next\_template$
        for (j in 1 to number_of_sensors) do
            $distance[j] \leftarrow dist(test[j], template[j])$
        end for
        Overall_dist = combine(distance)
    end for
    Output = min(Overall_dist)
end procedure
```

array of time-series data. The gesture database has pre-processed iterations of the same gesture along with other gesture data. The problem that the system is solving is comparing this test gesture data to each of the existing sample data using DTW and Energy based techniques as appropriate and then combining the results at the end to give a distance score. Then, at the very end, the task is to recognize the gesture based on the least distance. Algorithm 1 provides a high-level overview of this approach.

3.5 Technical Challenges

The need/demand for sign language recognition software for accessibility has been previously established, but accessing if there is a need or demand for systems that can utilize an established natural language as a human-computer-interface has not been studied (to our knowledge). As part of this research, a survey was taken where 13 users of ASL expressed their opinions on these topics that included their willingness

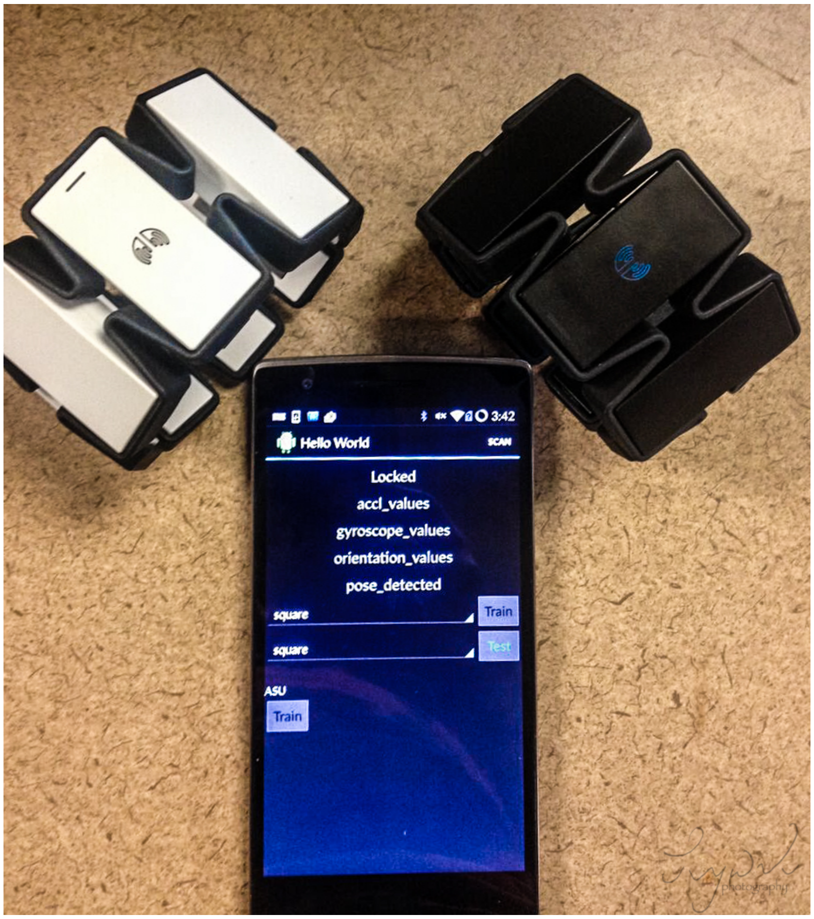


Figure 3.3: Myo Devices and Android Smartphone with the Sceptre App running.

to use a gesture-based system, the perceived practicality of using ASL for HCI, the acceptable time constraints for user to user and user to computer communications, etc. The results of the survey are summarized in Table 3.1.

3.5.1 Extendability

With more and more systems with voice command interfaces, it only seems plausible that a system that understands sign language will be desirable. To the question 2 in the survey, more than 75% of the ASL users agreed that they were very likely to use ASL for HCI, and the rest would be somewhat likely to use such a system.

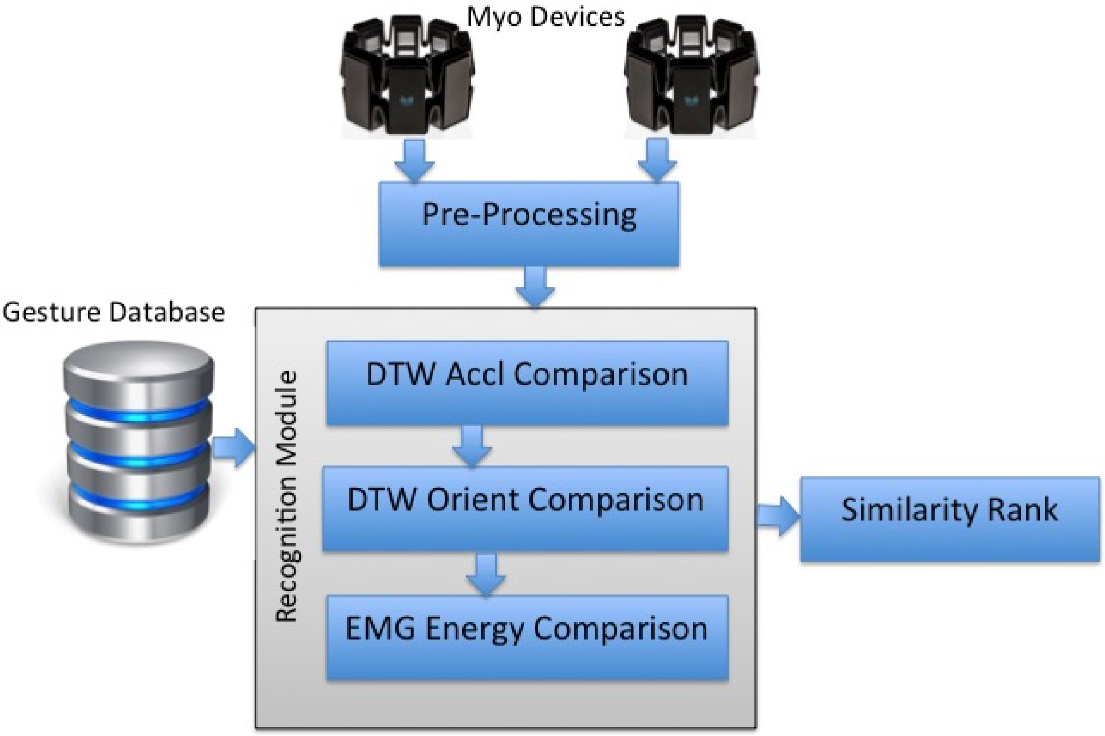


Figure 3.4: Gesture Comparison and Ranking: New Data is Collected, Processed and Compared With Existing Gesture Data to get a Ranked-list.

More than 75% people also responded that it was very important that the system be easily trainable, and another 15% thought it was somewhat important. This follows from the knowledge Johnston and Schembri (2007) that ASL (and sign language in general) varies a lot over geographical, racial and cultural bounds. Thus, the ability to add unrecognized signs or new signs easily is very important. When ASL users were asked if they were likely to use the system if it could be trained using 3 training instances, more than 76% said very likely and the rest responded they were somewhat likely to spend the time training. This requirement puts the constraint on the system that it has to be relatively easy to train. This system can be trained using only three instances of training per sign. The relationship between training instances and accuracy is summarized in Section 3.7.

3.5.2 Time Constraints

Communication is a real-time activity and the need for fast response times in both person-to-person communication as well as HCI is well established. A timing delay of 0.1 s is considered 'instantaneous' while a delay of 0.2 s to 1 s is noticeable to the user, but generally acceptable. If a system takes more than 1 s, then some indication needs to be given to the user about the 'processing' aspect Miller (1968). The survey question regarding timing backs this up: while more than 76% of ASL users said they were very likely and an additional 15% said they were somewhat likely to use the system if the response time was under 3 s, this number falls to 53% and 0% very likely responses when the response times were 3-7 s, and more than 7 s respectively. This establishes a strict constraint for recognition time for the system. To meet these standards the system has to be light-weight and should not rely on computationally expensive methods. Also, if the processing is delayed due to unpredictable issues like client network sluggishness, a proper indication has to be given to the user of the wait.

3.5.3 Non-Invasiveness

Historically there has been a certain stigma associated with signing. It was only in 2013 that the White House published a statement in support of using sign language as an official medium of instruction for children, although it has been known to be a developmentally important for deaf children for years Aaron J. Newman (2002). Given this stigma, and due to other more obvious reasons like usability, an interpretation system should be as non-invasive as possible. The survey results confirm this, as more than 90% thought that it is important or very important for the system to be non-invasive.

3.5.4 Pervasiveness

The other disadvantage of using more invasive technology like data-gloves or multiple-cameras is that it makes the system either tethered to a certain lab-based environment settings or it requires bulky equipment that discourages its use in an everyday setting Liang and Ouhyoung (1998); Kuroda *et al.* (2004); Fang *et al.* (2004). This makes the use of smart-watch like wireless equipment that can work on the streets as well as it works in a lab more desirable.

3.6 Approach and Methodology

Recognition of body movement is commonly achieved by computer vision techniques such as object detection or by directly tracking the movement of one or more body-worn sensors. Since I am presenting separate but related work for recognizing movement using both of these approaches, I will present the approach used separately.

3.6.1 Sensor based Approaches

The system consists of two Myo Devices connected via Bluetooth to an Android device which acts as a hub as shown in Figures 3.3 and 3.5. For the EMG part, a Bluetooth enabled computer is used to obtain the data since the Myo framework does not yet facilitate streaming EMG data to smartphones. For tests that do not use EMG data, Android device collects and does some pre-processing on the data of the gesture performed and stores this as time-series data. All this is sent to the server for processing. Three to four instances for each gesture data are collected, processed and stored. A high-level overview of the pre-processing that is done to the data before it

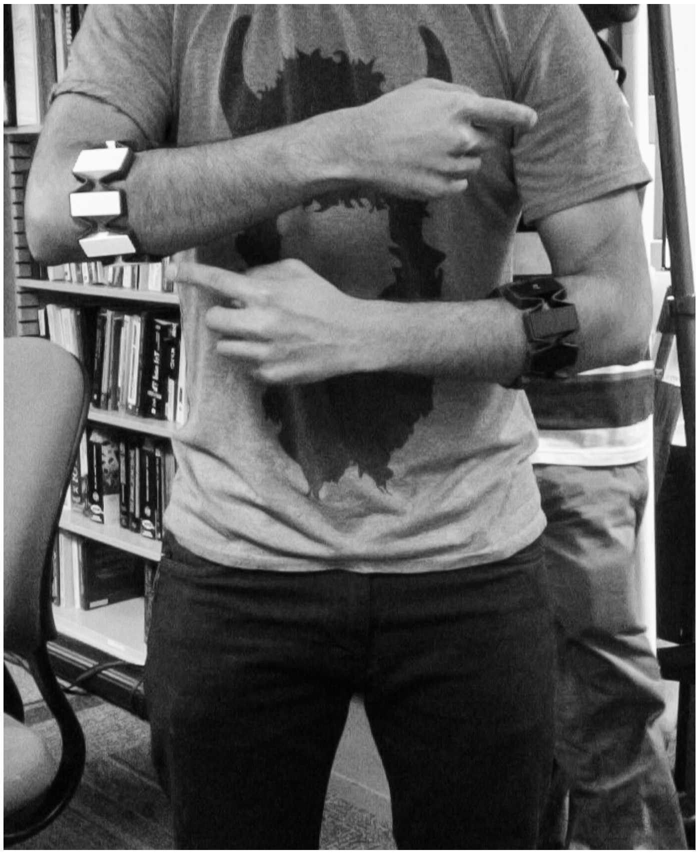


Figure 3.5: Deployment: A User with Wrist-band Devices on Each Hand Performing a Gesture.

is stored in the database is summarized in Figure 3.6.

At test time when a gesture is performed, the system similarly processes the gesture-data as described above. In the Architecture Figure 3.4, this is encapsulated in the 'Preprocessing' block. After this is done, the system compares this data with the other gesture data stored in the database using a specialized comparison method. The comparison method comprises testing accelerometer data, orientation data, and EMG data by different methods and later combining the results before calculating a 'nearness' score for each pair. After this step, a ranking algorithm is employed in the server to compute a list of stored gestures that are closest to the test gesture and the best one is returned.

A smart ranking architecture is used when the size of the database grows to ensure that the results come back in an acceptable time to be a real-time application. This is discussed in more detail in Subsection 3.6.6. The system also periodically uploads all recognized gesture data and other meta-data like user-specific accuracy, new signs trained etc. to the server. This information can be used to make future predictive methods more efficient which is not part of this work. The overall methodology of gesture recognition consists of the following steps as shown in Figure 3.4.

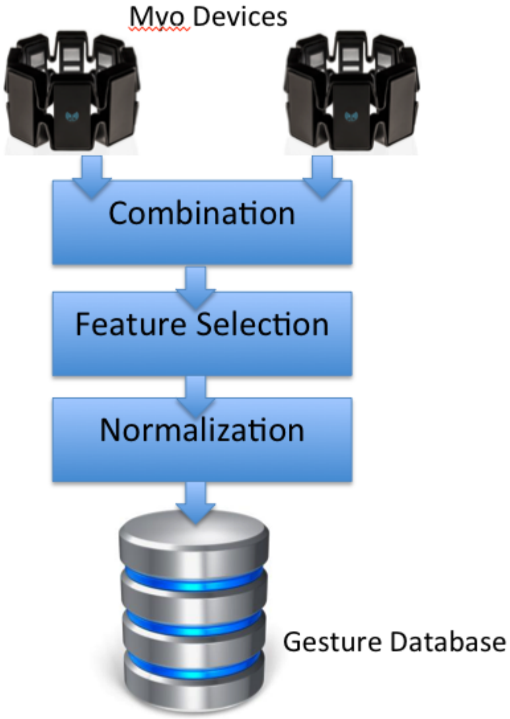


Figure 3.6: Pre-processing: Data is collected from the Myo devices, processed and then stored in Database.

3.6.2 Pre-processing

Data is collected from two Myo devices while the gesture is being performed (Figure 3.6). As soon as the end is detected or signaled, the pre-processing begins. The data collected from two hands is aggregated into one data-table then stored into a file as an array of time-series data. At 50 Hz. of sampling rate a 5 s. gesture data will consist of: 6 accelerometer Vectors of length 250 each, 6 gyroscope Vectors of Length 250 each, 6 orientation Vectors of Length 250 each, 16 EMG Vectors, Length 250 each. We combine all this for simplicity into a 34 × 250 matrix. Each time-series is transformed to make sure the initial value is 0, by subtracting this value from all values in the time-series. This will help prevent errors when the user performs the sign in different starting positions. Normalization is then done by representing all values as floats between 0 and 1 by a min-max method.

Orientation values are received in the form of three time-series in terms of unit quaternions. The pitch, yaw and roll values are obtained from the quaternion values w, x, y and z by using the following equations:

$$roll = tan^{-1}(\tfrac{2(wx+yz)}{-x^2y^2}) \tag{3.2}$$

$$pitch = sin^{-1}(max(-1, min(1, 2(wy - zx))))$$

$$yaw = tan^{-1}(\tfrac{2(wz+xy)}{-y^2z^2}).$$

3.6.3 EMG Energy

After correctly identifying the location of each of the individual pods of the two Myo devices, data is stored and shuffled in such a way that the final stored data is aligned from EMG pod-1 to EMG pod-8. This gives flexibility to the end-user as she no longer has to remember 'how' to put on the devices, and can thus randomly put either Myo device on either hand and expect good results. This is a great leap

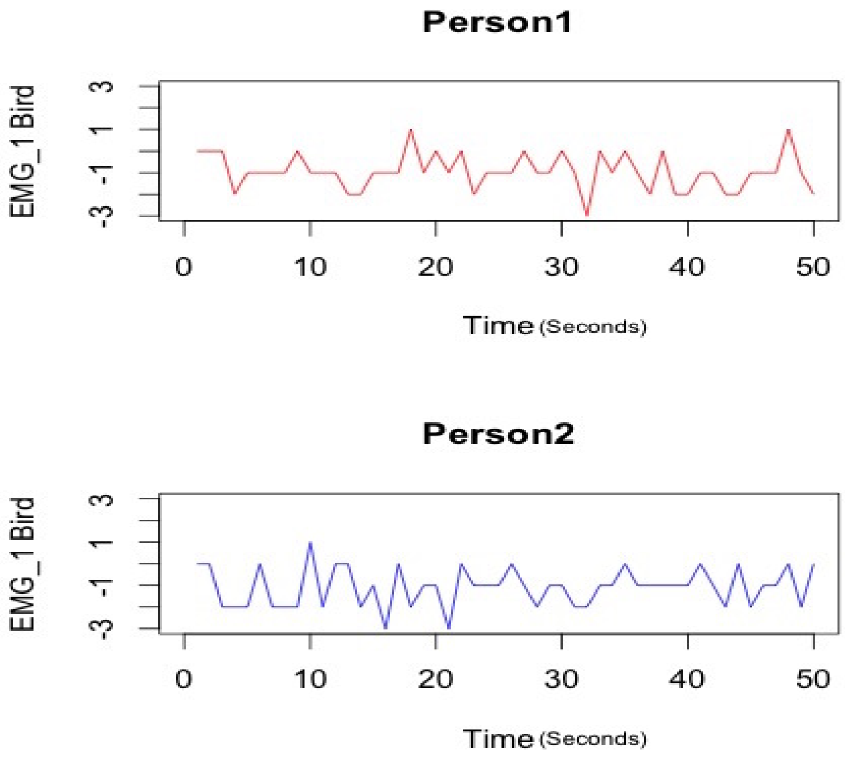


Figure 3.7: Variation in EMG signals between two users.

towards the pervasiveness and ease of use of this technology.

While we found that the DTW based testing worked very well for accelerometer and orientation comparisons, the EMG comparison numbers were not satisfactory. This is because EMG activations seem to be much more stochastic and a ’shape’ based comparison did not yield good results. The shapes formed by performing the same gesture for two different people are drastically different as seen in Figure 3.7, but they also differ in shape significantly enough between two repetitions by the same person to not give a small DTW distance. However, it was found by experimentation that gestures tend to activate the same EMG pods when repeated. Thus an energy-based comparison was tried:

EMG energy ’E’ on each pod is calculated as the sum of squares of x[n] , the value of the time-series at point 'n' .

$$E = sum(x[n]^2). \tag{3.3}$$

3.6.4 Dynamic Time Warping

We used four different approaches for comparing accelerometer and orientation data: a. Euclidian distance b. Regression c. Principal Component Analysis (PCA) and d. DTW. Euclidian distance simply compares the two time-series using mean-squared error. Regression analysis fits a model to the time-series and uses this model to compare the best fit for the test gesture. Given a set of features from the time-series for a gesture, PCA derives the optimal set of features for comparison. According to Müller (2007), DTW is a well-known technique to find an optimal alignment between two given (time-dependent) sequences under certain restrictions. It was found that DTW based methods gave the best results. DTW based approached proved ideal in our use-case since it could gracefully handle the situations when the sign was performed slower or faster than the stored training data, or performed with a small lag. Traditionally, DTW has been used extensively for speech recognition, and it is finding increasing use in the fields of gesture recognition as well Rabiner (1989); Lee *et al.* (2019), especially when combined with Hidden Markov Models Wilson and Bobick (1999). SCEPTRE takes a simpler approach by randomly re-sampling the training and test datasets based on the least number of points in either one, then doing a DTW based distance analysis on them. First, a DTW analysis of Accelerometer Data is run and a ranked list of 'probable' signs is passed on for DTW based analysis of orientation Data which in turn creates an even shorter ranked list to be processed by the final EMG algorithm.

On another approach, the normalized distances from each of the outputs are taken and the sum of squares of the final output is taken as an indication of 'closeness' of

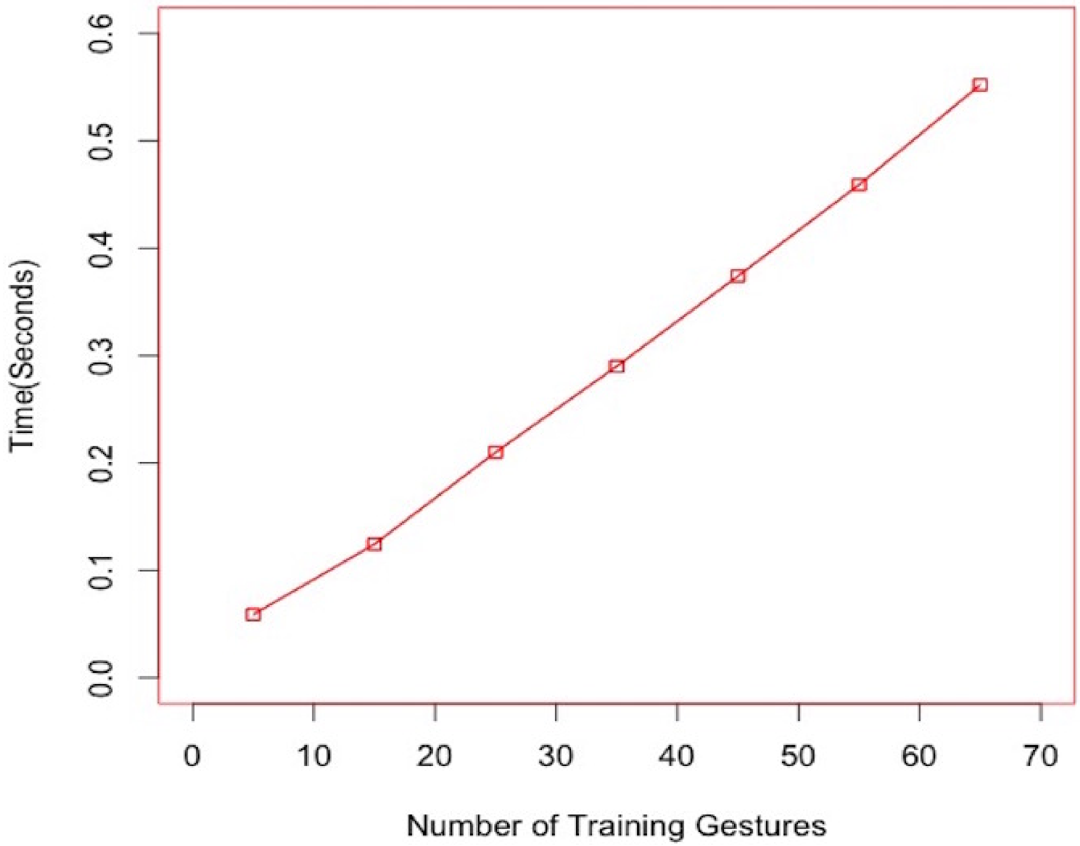


Figure 3.8: Database Size vs. Recognition Time.

a test sign to a training sign. This is the approach that was chosen due to its computational simplicity and speed.

3.6.5 Combination

The overall 'nearness' of two signs is computed to be the total distance between those signs which is obtained by adding up the scaled distances for accelerometer, orientation and EMG as discussed above. An extra step of scaling the distance values between (0,1) is performed to give equal weight to each of the features. Also, since we have 8 EMG pods and only 3 each of accelerometer and orientation sensors, we use the following formula for the combination. The formula is for combining the accelerometer

Algorithm 2 Overall Gesture recognition method.

```
procedure GestureRecognition
    $t \leftarrow$ elapsed time from start of the experiment
    $test\_gesture \leftarrow$ get gesture data from the user
    $S_G \leftarrow$ query list of recognized gestures from database
    $normalized\_accl \leftarrow accel\_values - first\_accl\_value$
    $normalized\_orient \leftarrow orientation - first\_orientation$
    $normalized\_emg \leftarrow emg\_values - first\_emg\_value$
    for each trained gesture $T_G \in S_G$ do
        $dtw\_accl \leftarrow dtw(training\_accl, normalized\_accl)$
        $dtw\_orient \leftarrow dtw(training\_orient, normalized\_orient)$
        $T_G^E \leftarrow$ calculated energy of each pod for the training gesture
        $emg\_energy\_diff \leftarrow T_G^E$ - test_emg_energy
        $scal\_lim \leftarrow \{0,\ 1\}$
        $scaled\_emg\_diff \leftarrow$ scale(emg_energy_diff,scal_lim)
        $scaled\_accl\_dist \leftarrow$ scale(dtw_accl, scal_lim)
        $scaled\_orient\_dist \leftarrow$ scale(dtw_orient, scal_lim)
        $combined\_nearness \leftarrow$ compute nearness from Eq. 3.4
    end for
    Sort $T_G$ with respect to combined nearness.
    $recognized\_gesture \leftarrow$ top $T_G$ in the sorted list
end procedure
```

sum of distances and EMG sum of distances. Similar techniques were applied for the other combinations. An algorithmic summary can be found in Equation 3.4.

$$dist = (8cs(sc\_accl\_comb) + 3cs(sc\_emg\_comb))/24. \tag{3.4}$$

where $cs()$ is a function that returns the sum of columns, sc_accl_comb is a data frame that holds the combined accelerometer DTW distances for both hands for all trained signs, and sc_emg_comb is a data frame that holds the combined EMG energy distances for both hands for all trained signs. The overall algorithm that is employed for recognition is summarized in Algorithm 2.

Algorithm 3 Optimization to meet time constraints.

```
$databasesize \leftarrow sizeofgesturedatabaseforuser$
$time\_to\_compute \leftarrow estimatedtimefrompreviousiterations$
if $time\_to\_compute < time\_contraint$ then
    Compute Similarity Score with one template per gesture
    $top\_gestures \leftarrow list(top\_gestures, time\_to\_compute, time\_constrant)$
    Utilize Algorithm 2
    END
end if
```

3.6.6 Ranking and Optimization

Timing constraints due to the real-time nature of the application requires us to optimize the recognition algorithm to be efficient, especially as the gesture space increases. As the number of gestures in the database increases to beyond 60, as we can see in Figure 3.8, the recognition time for identifying one gesture goes beyond the .5 s mark, which we have chosen to represent a real-time timing constraint. Thus, to deal with this situation, we modify our comparison algorithm to first compare to one stored instance of each gesture, then choose the top 'n' number of gestures which when compared to 'k' of each, still allowing the time-constraint to be fulfilled. We then, proceed with the normal gesture comparison routine on only these gesture instances and thus keep the recognition time within our constraints. All the variables for this method viz. the 'n', 'k' are calculated dynamically by what is allowed by the timing constraint 'tc', thus making this approach fluid and adaptable to more vigorous time constraints if required. The overall optimization algorithm is summarized in Algorithm 3.

3.6.7 Video based Approaches

We utilize a variant of Residual Neural Network Tai *et al.* (2017) from Tensorflow.js Smilkov *et al.* (2019) called PoseNet Papandreou *et al.* (2017). PoseNet can be used for single as well as multiple pose detection, which means that it can detect many people in the same frame. However, for our purposes, the single pose detection module was sufficient. The ResNet50 powered model was configured with optimum default values considering the performance speed for Output stride(32), image input resolution(257) and enough quant bytes(2) to perform weight quantization. For each frame in a video or continuous stream, PoseNet returns 17 different keypoints throughout the body. Among these, the keypoints for eyes, nose, shoulder, elbows, and wrists were utilized. The eyes and the nose keypoints along with those for the wrists are used initially before the learner begins to record their execution. Along with the 'x' and 'y' coordinates, PoseNet also returns a confidence score between 0 and 1 for each keypoint for every frame. A threshold of 0.6 was utilized to calibrate the webcam positioning and to verify the learner was in range and clearly visible. After verification of about 60 frames, a 3 s timer to begin recording is triggered. As a learner is recording their execution the keypoints for every frame are recorded to be used for a comparison to the tutor for that sign. The keypoints estimated for 5 different body parts throughout the videos for signs 'Spanish' and 'Nice to meet you' are plotted in Figures 2.3 and 2.2 respectively.

3.7 Results and Evaluation

The choice of performance metrics and the test set heavily influences the performance results. In this section, I provide details on how the experiements were designed and executed as well as the results obtained.

3.7.1 Design of experiments

Experiments were designed to test the application with scalability in mind. The system begins on an ”empty slate” and the gestures are trained interactively. Since a template-based comparison is being utilized, we store the entire data set plus some features. After all the data is collected, the system is ready to be trained for another gesture. A sample application screen can be found in Figure 3.3. It is advised to provide 3-4 training instances of the same gesture to improve accuracy of recognition but that is not required. We test the system with 1, 2, 3, and 4 instances of each gesture to determine the correlation between the number of instances saved and the recognition accuracy rate. That is all that is needed at the training phase. During the testing phase, we evaluate our system based on the time it takes for recognition and on the combination of features that produce the best results. Then we evaluate how recognition time increases with the increase in the number of gestures.

The way gestures are performed by the same person are generally similar but they tend to vary slightly over time. Although, considering signals generated by a portable device that the user can put on and off by herself, provides the system a lot of portability and pervasiveness, it comes with a drawback that the signals we receive might not always be the same. To account for this experiments were performed that allowed users to put on the devices themselves, they were allowed to face any direction, and perform the gesture as long as they stayed in the Bluetooth range of the hub. Also, samples were collected over multiple sessions to account for variability in signing over time. The test subjects were 10 University students between the ages of 20 and 32 who performed the gestures in multiple sessions while also varying direction they were facing and if they were standing up/sitting down. The subjects viewed a video recording of the gestures being performed by ASL instructors before performing them with the system.

3.7.2 Prototyping and Choice of Gestures

20 ASL signs were chosen to prototype the system. They were carefully chosen such that a. A good mix of the various ASL components as discussed in Section 3.4.1 and b. To include signs that are very close in some components but different in others. The choice of signs and the break-down of components for 10 of the 20 chosen signs is summarized in Table 3.2. The other 10 signs that were chosen are hurt, horse, pizza, large, happy, home, mom, goodnight, wash. Detailed analysis of system performance is given in Section 3.7.

3.7.3 Performance Metrics

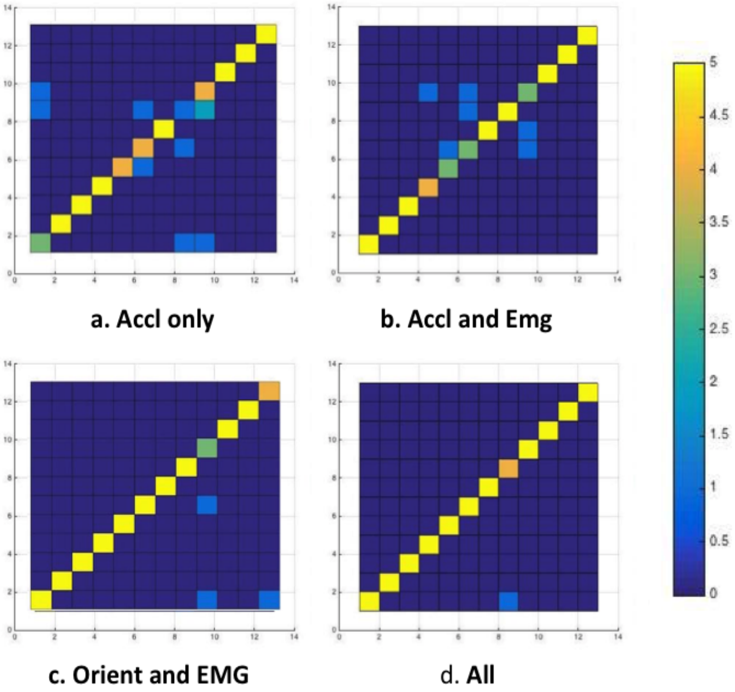
The system is evaluated on each of the requirements discussed in Section 3.5. Pervasiveness of the system is justified since it is wireless and can work in Bluetooth range of the hub device which can either do the processing itself or offload it to a cloud server. Invasiveness is a harder metric to evaluate, as this seems to be more subjective. However, with wearable devices like smartwatches, wristbands such as FitBit, Myo becoming increasingly compact, wireless and even stylish Gupta *et al.* (2013), the proposed system is much less invasive then any of the alternatives. Accuracy is undoubtedly one of the fundamental performance metrics for any recognition system. This basic function of the system is to facilitate real-life conversations, thus the system has to be able to function in real-time. Another aspect of the system is that it is extendable, thus the ease in scalability to a larger number of signs is very important. With the increase in the gesture space, the recognition time will also increase, the system should be able to scale up with a reasonable recognition time. Although there is a slight dip in recognition rate, the potential for scalability showcases the system's flexibility. Thus the experiments focus on these three metrics to evaluate the system:

Table 3.2: Component Breakdown of 10 of the Chosen Signs.

#	Sign	Location(Tab)	Movement	Orientation	Hands
1	Blue	Around the Face	Wrist-Twist	Away from signer	1
2	Cat	Around the face	Outward and closing	Towards center	2
3	Cost	Trunk	Wrist-Flex	One towards signer one towards center	2
4	Dollar	Trunk	Two Wrist-Flexes	One towards signer one towards center	2
5	Gold	Side of the head	Away from center, change handshape twist	Towards the signer	1
6	Orange	Mouth	Open and close twice	Towards user	1
7	Bird	Mouth	Pinch with Two fingers	Away from signer	1
8	Shirt	Shoulder	Twice Wrist Flex	Facing down	2
9	Large	Trunk	Both Hands Away from Center	Facing away from signer	2
10	Please	Trunk/Chest	Form a circle	Both facing towards signer	1

1. Recognition time
2. Extensibility
3. Recognition rate (Accuracy)

It was determined through experiments that three is a good choice for the number of training instances for each gesture. This formed a good compromise between usability and results as shown in Figure 3.10. Thus, each gesture was performed three times at varying times in the day. Then all data was aggregated, and annotated according to the gesture performed. The testing comprised of taking one dataset and comparing it with all other data sets and then estimating the correct gesture. With


a. Accl only

b. Accl and Emg

c. Orient and EMG

d. All

Figure 3.9: Gesture Recognition vs. Features Used.

the 'guided mode' turned on, a recognition rate of 100% was achieved as expected. Then 20 randomly selected ASL Signs were used. The best accuracy was obtained by a tiered-combination method of all three features. The relative accuracy of other methods can be seen in Figure 3.11. This helps confirm that the combination of all three features produces the highest results.

Figure 3.8 shows how the recognition time increases with the increase in the number of gestures that are stored in the system. With 65 gestures in the database, a recognition time of 0.552 s was achieved. (processed on a 3.06 GHz Intel Core i3 8 GB RAM Mac). This fast response time means that the system qualifies for use for real-time communications. This time can be further improved upon by utilizing a more powerful server and optimizing the data queries, which will be part of future work.

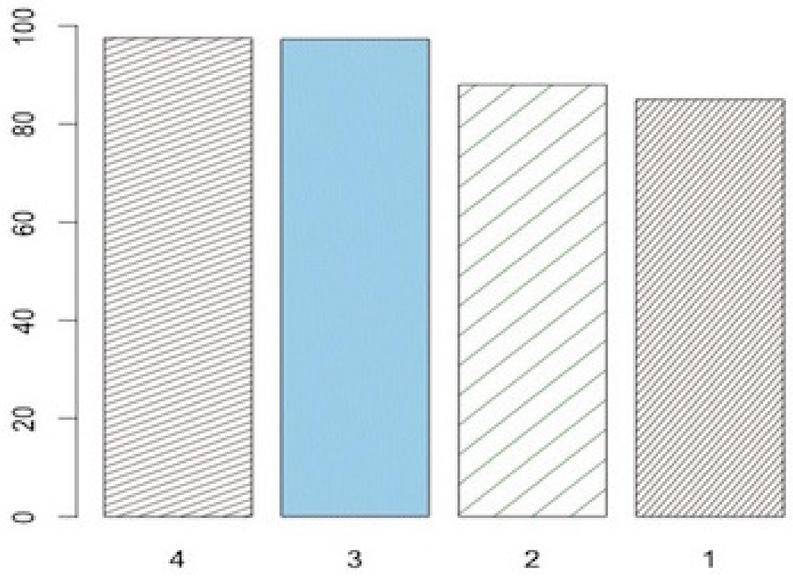


Figure 3.10: Training Data vs. Accuracy.

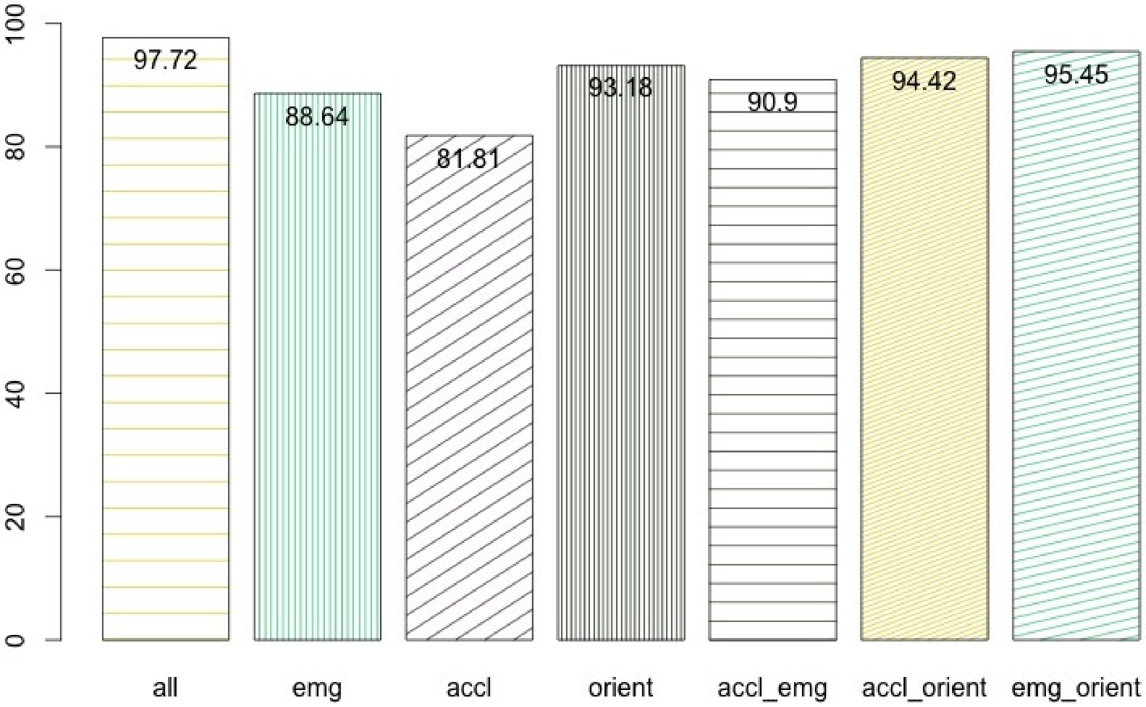


Figure 3.11: Features vs. Accuracy.

Figure 3.9 shows the performance of the system based on recognition results for a combination of the features. The gestures tested were 'day', 'enjoy', 'eat', 'happy', 'home', 'hot', 'large', 'mom', 'pizza', 'shirt', 'still', 'wash' and 'wave'. The different sub-figures show that although comparing solely based on accelerometer performs good for some gestures like 'happy', it is ineffective in distinguishing between others like 'day'. This can be explained because the nature of performing this gesture is very similar to other gestures in overall hand movements, but are different when it comes to finger configurations. Thus the performance gets better as shown in Figure 3.9c when EMG sensors are brought into the equation. Like we discussed earlier, the least number of recognition errors overall occur when all three of the sensor data are fused as seen in Figure 3.9d. This gives an insight into where the system is error-prone and thus can be optimized. Server level Optimization based on such input will be part of future work. We envision a server that continuously monitors success-failure data and becomes better by implementing machine learning algorithms to give weights to the different features.

These results can be understood better in context with the breakdown information given for each gesture in Table 3.2. For instance, Figure 3.2 shows screenshots of a person performing the gesture 'mom' and another one performing 'eat'. These two gestures are very similar in 'Location' and 'Movement' but are distinctive when it comes to 'HandShape' and 'Orientation of Hand'. Thus, while the system isn't able to distinguish between these two gestures based on Accelerometer information only as seen in Figure 3.9a, the system does well when EMG information is included as seen in Figure 3.9c.

The results in Figure 3.11 show an accuracy of 97.72% for 20 ASL signs when all accelerometer, orientation, and EMG are used. The database consisted of signs from 10 different people, and each person performed each sign 3 times. This table also lists the accuracy of other variations in the combination of these three input signals. The accuracy varied when individual databases were separated, however, it stayed in the (97-100)% bracket when all signals were used on a dataset consisting of template gestures from the test user.

3.8 Discussion

Continuous Processing: The gestures that were recognized in this test were isolated in nature. A start and stop button, or a timer was utilized to gather data. This was done to test the algorithms in isolation first. However, when the system is deployed to the public, a continuous monitoring algorithm has to be utilized. This can be done by using windowing methods and optimizing based on the best results. However, we expect that the system will require a lot more training.

Dictionary Size: Another thing that can be improved upon is the dictionary size and the support for Signed Alphabet. Signed alphabet gives an ASL user the ability to use the English Language Alphabet to visually spell out a word. This is an important aspect of sign language communication since not all English words or ideas have sign language counterparts. The dictionary size used is currently 20 ASL words and 10 users invented gestures. The video-based sign language dictionary website signasl.org currently has over 30,000 videos for sign languages in use, so there is a lot of room to grow into.

Framework Limitations: Currently due to limitations of the Myo framework for Android, final tests involving EMG data signals had to be done using laptop computers. Two Mac computers were used to gather the data simultaneously via Bluetooth channels and then combine them and store them to a cloud-based data storage system. A separate script was triggered to process the stored data at test time. With the anticipated release of the new framework for Myo Devices, this computation can be done at the mobile phone level and data can be sent to the server only at designated intervals.

User Independence: EMG data fluctuates a lot between people. This is apparent from Figure 3.7. More research can be done on data gathering and feature selection of the EMG data pods to come up with a 'unifying' framework that works for everyone. This will be a definitive direction in attaining user-independence while using EMG signals.

Search Algorithm: The search algorithm that is implemented can be improved upon to decrease the recognition time. Hierarchal searching can be done, in which training gestures are clustered according to the energies in each time-series data, and only those gestures are compared which have comparable energies.

SCEPTRE in collaboration with other HCI techniques such as brain-driven control Oskooyee *et al.* (2015); Pore *et al.* (2015); Sohankar *et al.* (2015) can revolutionize the way people interact with computers. In addition to the uses of HCI or Sign Language Communication, this technology can also be extended to uses in the domains of activity recognition, food intake estimation or even physiotherapy.

3.9 Conclusion

DTW and energy-based comparison methods are fast and highly effective. Template-based comparison techniques have a great advantage of lossless information storing. Comparing accelerometer and orientation data between gestures is best done by using Dynamic Time Warping methods in which muscle movement (EMG) data is best compared by comparing total energies in each pod. The structure present in Sign Languages like ASL can be divided into components that explain the success of certain signals in recognizing them. The overall accuracy of the system is increased by a smart combination of these signals without compromising on speed, recognition rate, or usability. The sensors that are used are non-invasive and versatile thus allowing people to effectively utilize them in day-to-day situations without drawing much attention. The future direction of this research is in incorporating continuous sign processing, user independence in the system and increasing the dictionary size.

Chapter 4

RECOGNITION OF CORRECT HANDSHAPE

4.1 Introduction

Handshape is also one of the most important attributes of American Sign Language. ASL signs which are otherwise similar may differ semantically only by the shape or orientation of the hands. Similar to location or movement recognition, handshape recognition can be performed using cameras or body-sensors. In this chapter, I will present the work I have done towards performing handshape recognition using both of these approaches.

For video-based techniques, the first step for handshape based recognition is to obtain a tight crop of each of the hands. This helps to ensure that the comparison being done is focused on the shape of the hand and not on external factors such as background objects. Many techniques exist in the literature that attempt to localize known objects in an image or video. In this case, the known object are the hands. Classification techniques that output the presence or absence of a particular category of images can be utilized to detect if an image or a frame in a video has hands. However, to obtain a crop of the handshape for comparison each of the hands must also be localized within the frame. Also, the localization for the right and left hands should be separable since they have distinct semantic significance. However, the apparent similarity between the two hands of a user as it appears to a camera makes the usage of object detection and tracking algorithms infeasible. Also, the crops obtained for the hands must be tight enough for image comparison algorithms to have good performance. This means that in addition to using landmark detection algorithms that track salient points in the body, a bounding box that includes the signers' hands must be obtained. Further, this tracking and localization algorithm must perform well in real-world conditions that might have variance in lighting, distance, resolution conditions. Finally, after a crop of the hand has been isolated, comparing all of the crops from a video to those from another video for similarity-based comparison is also challenging. Firstly, there is the problem of alignment, since both videos may have varying frame rates, sign execution speeds, delays in start-stop, differences in frame-sizes or other such challenges. Then, there are issues related to differences in size, skin-tone, resolution that make the one-to-one comparison part challenging, even after proper alignment has been achieved. If a model-based classification technique for video recognition is used instead of a similarity-based approach, then the model has to be re-trained to support new vocabulary. Also, for a model-based recognition algorithm to be effective, the alignment and classification have to be learned jointly as the shape of the hand can change during the execution of a sign. Thus, for video-based handshape recognition part, first, a landmark detection algorithm is utilized to detect key body points including the wrists of the signer Pugeault and Bowden (2011). Then, the position for the crops for the handshapes can be estimated roughly by using the location of the wrist as reference Paudyal *et al.* (2019c,b). However, when handshape crops were done using the wrist position for different videos two issues were encountered: 1) Depending upon the orientation of the hands, the crops did not always contain the entire hand unless the size of the crop was made very large relative to the signer‘s body and 2) The distance of the signer from the camera impacted the quality of the crop as signers that were closer to the camera needed bigger crops. The first issue was accounted for by translating the crop position using a straight-line projection between the location of the elbow keypoint and that of the wrist seen in Figure 4.2. The second issue was mitigated by using the distance between the shoulders' of the signer to compute a relative size of the crop rather than an absolute one. After a reasonably accurate crop of the handshape was derived, the next step is to compare this to that of a tutor. For this, a cosine distance-based comparison was used on the penultimate activation output of a convolutional neural network. An inception-v3 model pre-trained on Imagenet was retrained using an open dataset for ASL fingerspelling recognition Pugeault and Bowden (2011). In addition to account for differences in orientations, several techniques for data augmentation such as random cropping, rotations, and hue-translations were applied Wu *et al.* (2015). This process is called transfer learning and is instrumental when training examples are scarse Kornblith *et al.* (2019). The intuition here is that the CNN utilized is trained to recognize shapes of hands and activation values for the final layers are representative of how close they are to certain shapes of hands present in the finger-spelling dataset. This process is summarized in Figure 4.3. After each cropped image was run through the pre-trained CNN a vector of length 26 is obtained corresponding to the classes in the fingerspelling dataset. Then, each of these vectors obtained for a learner was concatenated and was compared to the concatenated vector obtained for the tutor using cosine similarity. The correctness of the handshape is determined by thresholding on the final score. For the user-studies, a score of 0.75 was chosen by experimentation. This threshold like the thresholds for movement and location should be chosen experimentally depending on the domain and the verboseness of feedback desired.

The advent of wearable electronics has opened up a new era of pervasive human-centered computing which when applied to disability research results in technology of significant impact. Sign language recognition (SLR) systems is such an example, where wearable technology can be used to drastically improve communication between the sign language cognizant deaf and hard of hearing and the agnostic mass. These

SLRs collect data from wearables to recognize gestures, which can be then converted into text or speech. Wearable sensors such as Myo Paudyal *et al.* (2017); Lee *et al.* (2018) contain electromyogram (EMG) sensors that allow measurements of muscle tension as well as Intertial Measurement Units (IMU). These sensors can be utilized to recognize the shape of a learner's hands. The advantage of using wearable sensors for handshape and sign language recognition is the freedom of movement for the signer since they are no longer restricted to having to be in front of a camera or other specialized setup. The problem of finger-spelling detection is analogous to the detection of handshape since letters in the singed alphabet are differentiated by the shape of the hands. Thus, effective solutions for handshape recognition are not only useful for sign recognition but are also useful for finger-spelling recognition.

American Sign Language (ASL) (potentially true for many other SLs) has four important aspects (Figure 4.1): a) word gestures for commonly used words, b) finger-spelling, through which unique hand gestures are associated to a letter in the English (or another language) alphabet Stokoe (2005), c) emotion or expression, through which abstract concepts such as intensity and finer expressions are depicted, and d) individual or customized variations. Gesture recognition research so far has mostly focused on the first two aspects of ASL. Although there are several non-invasive wearable systems available for word recognition, accurate finger-spelling recognition is still a challenge. This is because ASL signs for words use significant arm movements that can be easily captured using sensors such as accelerometers, or gyroscope available in state-of-the-art wearables. However in finger-spelling, arm movements are restricted, rather relative positioning of the fingers are important distinguishing factors amongst gestures for different words. Such nuances are difficult to capture using wearables and may require other infrastructure such as video monitoring Bossard *et al.* (2003). Electromyogram (EMG) signals, which measure the tension in muscles are available in state-of-the-art wristbands, which can be potentially used for recognizing relative finger positioning. However, EMG sensors have very low signal to noise ratio, and vary not only from one individual to others but also during different attempts by the same individual. I proposed DyFAV (Dynamic Feature Selection and Voting) that exploits the fact that fingerspelling has a finite corpus (26 letters for ASL) and uses an independent multiple agent voting approach to identify letters with high accuracy. The custom approach is shown to perform better than standard machine learning approaches of Support Vector Machines, Naive Bayes and Multiple Layer Perceptron.

4.2 Significance

An ASL user would use the American Fingerspelled Alphabet (AFA), (also called the American Manual Alphabet). The AFA consists of 22 handshapes (4.1) that when held in certain positions and/or are produced with certain movements represent the 26 letters of the American alphabet Lifeprint (2016). There has been some work recently Paudyal *et al.* (2016); Zhang *et al.* (2011); Kim *et al.* (2008) that use armbands or other sensors on the arm to get information about sign language gestures, however, there is not any work that aims to incorporate fingerspelling along with it.

The use of fingerspelling in ASL has been studied extensively and researchers agree that fingerspelling is integrated into ASL in very systematic ways Brennan (2001); Padden and Perlmutter (1987). The most obvious usage is primarily for representing proper nouns or for English words without equivalents in sign language called neutral fingerspelling Battison (1978); Wilcox (1992); Padden (2006). Besides that, Lexicalized fingerspelling, abbreviations for longer signs, two-word compounds, initialized signs and singed-fingerspelled compounds Baker (2010) all utilize fingerspelling. Fingerspelling also helps to bridge the gap created due to the variation of ASL lexicon across geography and cultures Brentari and Padden (2001). This tight integration

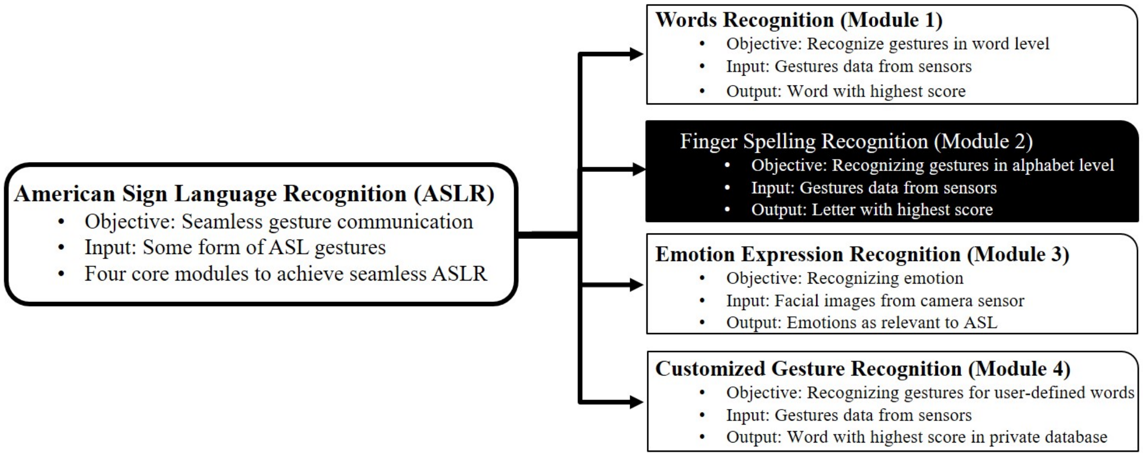

American Sign Language Recognition (ASLR)
Objective: Seamless gesture communication
Input: Some form of ASL gestures
Four core modules to achieve seamless ASLR
Words Recognition (Module 1)
Objective: Recognize gestures in word level
Input: Gestures data from sensors
Output: Word with highest score
Finger Spelling Recognition (Module 2)
Objective: Recognizing gestures in alphabet level
Input: Gestures data from sensors
Output: Letter with highest score
Emotion Expression Recognition (Module 3)
Objective: Recognizing emotion
Input: Facial images from camera sensor
Output: Emotions as relevant to ASL
Customized Gesture Recognition (Module 4)
Objective: Recognizing gestures for user-defined words
Input: Gestures data from sensors
Output: Word with highest score in private database

Figure 4.1: Modules of American Sign Language Recognition

with ASL means that an ASLR system that aims to translate conversations in ASL to English must incorporate fingerspelling recognition as an integral part as seen in Figure 4.1.

Incorporation of fingerspelling in ASLR may also result in a more convenient and user-friendly system. For instance, during the course of a translated communication session in a critical scenario such as a medical emergency room, an ASL user may want to communicate a specialized medical condition, history of drug use or operative procedures to a caregiver, which do not have ASL word equivalents. This is very much a possibility as Random House Sign Language Dictionary has 4500-word listings in it, while any ordinary English Language Dictionary has 200,000 words in it Costello (2008a). In such a case, if the SLR system being used does not support fingerspelling recognition, the user must stop the system and change modalities to a pen/paper or other mediums to convey that word/idea which will hamper the overall usability. In the case of using such a system for HCI, the system will not be truly independent of other input modalities unless it can support fingerspelling.

The same techniques used for fingerspelling can be modified to be used for handshape recognition. Signs in ASL are composed of several components: the shape the hands assume, the orientation of the hands, the movement they perform and their location relative to other parts of the body. A variation in any one of these components can alter the meaning of the word. Thus accurate recognition of handshape is needed for sign language recognition.

4.3 Related Work

There are numerous ways to recognize the movement of a hand being tracked. Approaches vary according to the devices used, the dimensions being tracked as well as the precision required. Hands can be localized in space either by having body sensors attached to them, or by cameras monitoring the scene of action. In this section, I provide details on some influential work that is related to this work.

Data glove based approaches

Data glove based approaches have been used extensively to recognize sign language alphabet as well as words Fels and Hinton (1993, 1998); Khan and Ibraheem (2012); Sole and Tsoeu (2011); Liang and Ouhyoung (1998); El Hayek *et al.* (2014). Due to the presence of finger joint information, it becomes relatively easy to classify the various finger configurations. Contrasting to the image and video-based approaches, the data glove based approaches don't suffer from problems of occlusion, lighting changes, and background variations. However, these approaches are invasive as wearing a data glove prevents the user from participating in day-to-day activities as well as singles the sign language user out. The proposed work in this paper attempts to solve this problem by providing an alternative that uses inconspicuous armbands (Myo) to gather the information.

Table 4.1: ASL symbols and their hand configurations

Name	Symbol	Description
Fist	A, S , T	The hand clasped with thumb
Flat hand	B	The open or spread hand, thumb out or in
Curved hand	C, O	Fingers connected with thumb or not
Retracted hand	E	The fingers clenched to palm
F-hand	F	Thumb and forefinger touch, other fingers spread
Index	G	Allocheric forms: g, d, l
H-hand	H	First two fingers extended and joined
Pinkie or I-hand	I	The little finger projects from the closed hand
K-hand	K	The index, second and thumb make k
L-hand	L	The thumb and index make right angle
Bent-hand	M	The hand makes a dihedral angle
R-hand	R	The first two fingers crossed; r
V-hand	V	The index and second extended and spread
W-hand	W	The first three fingers extended and spread
Y-hand	Y	The thumb and little finger are spread out from fist

Image, video and depth based approaches

Fingerspelling forms distinct shapes thus it is very intuitive to utilize image and video-based approaches to classify the various hand configurations that relate to the various letters (and numbers). Image and Video-based approaches have achieved high accuracy rates Singha and Das (2013); Pugeault and Bowden (2011); Paulino da Silva *et al.* (2014); Starner *et al.* (1998). There have been some approaches to classifying continuous gestures using video with high accuracy Tubaiz *et al.* (2015). Furthermore, there have been approaches that use the depth information provided by the Kinect sensor to improve classification accuracy. Image and video-based approaches need special equipment setup before usage which is a major limitation. Also, they face the unavoidable problems of occlusion, sensitivity to ambiance light and background changes. Furthermore, classification between visually similar letters like *S*, *M* or *N* is difficult with images. To get around this problem, some researchers have evaluated their systems on subsets of the alphabet Pugeault and Bowden (2011). These limitations prevent image, video, and depth-based systems from being practical for day-to-day usage.

Sensors on the arm based approaches

Using armband or sensors on the arm is a relatively new and promising approach to SLR that utilizes various combinations of sensors like accelerometer, gyroscope, and EMG Paudyal *et al.* (2016); Savur (2015); Yun *et al.* (2012). Some other works have focused on providing algorithms for gesture recognition for control of devices Lu *et al.* (2014). The accelerometer and gyroscope sensors provide location and movement information while the EMG sensors help to classify the finger configurations. Recognizing fingerspelling, however, provides a unique challenge since there is little or no movement involved in most of the letters in ASL and EMG signals can be noisy. Savur et al. have proposed a real-time and offline system to translate ASL fingerspelling to letters. The authors report an accuracy of 82.3% on a real-time system and a 91.1% on an offline system using all 26 letters repeated 20 times each for training Savur (2015). The approach we discuss in this paper achieves a higher real-time average accuracy of 95.36%. Also, only 5 training instances are required which is much smaller than other works Savur and Sahin (2015); Pugeault and Bow-

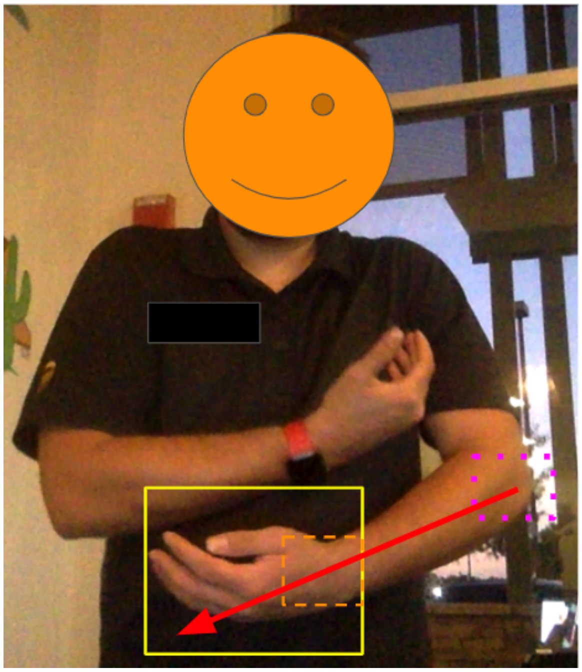

[redacted]

Figure 4.2: Using Elbow(dotted pink) and Wrist(dashed orange) Location to obtain a Handshape Crop(Solid Yellow)

den (2011). Further, we use commodity wireless sensors in place of the medical-grade sensors, and perform all testing and training using a smartphone app (to be available in play store). To increase accuracy we utilize accelerometer, orientation and gyroscope sensors in addition to the EMG sensors.

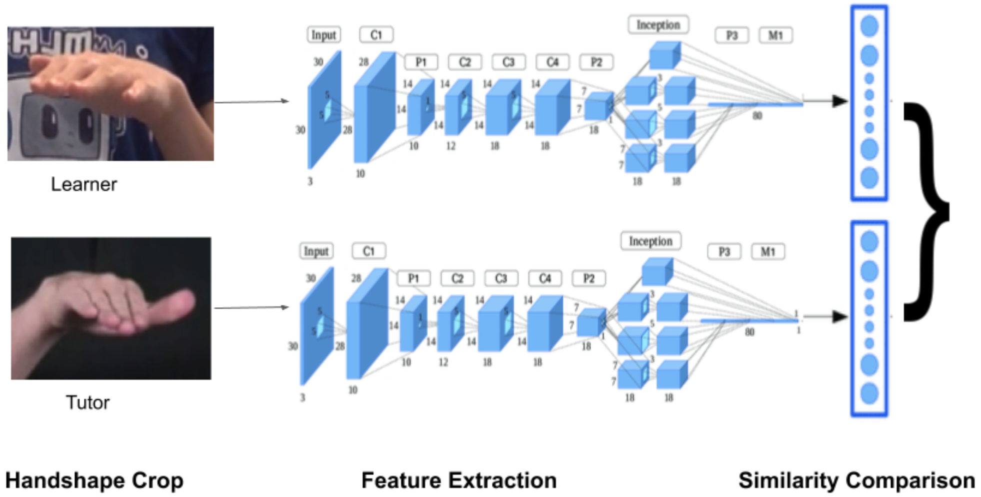


Figure 4.3: Using a CNN as a Feature Extractor to measure Handshape Similarity.

4.4 Problem Statement

4.4.1 Sensor Based

Input: Discrete signals for accelerometer, gyroscope, orientation and EMG from Myo Devices.

Output: Recognized alphabet.

Challenges:

1. *Low SNR signal:* The signals, especially the EMG has a very low signal to noise ratio. Thus, it makes it difficult to design an algorithm that performs consistently.

2. *Difference between people:* The signals, especially the EMG varies considerably between people even when performing the same alphabet

3. *Diversity of significant features:* A different combination of significant features appears that helps to identify different alphabets.

4. *Lack of significant movement: Fingerspelling as opposed to other ASL gestures do not involved much arm movement*

Constraints:

1. *Real-time constraint:* To facilitate recognition in real-time, the recognition algorithm must be light-weight.

2. *Limited processing:* Recognition is done in the smartphone to remove network dependence and lag. Smartphones don't have as much processing power as servers.

3. *Limited Training Data:* To make the usability of the system high, training sessions are repeated only 5 times per letter per person. This puts a constraint on the amount of data available for training.

The core problem we solve in this research is a fixed-class classification problem of identifying a letter performed using just the signals captured by a wearable armband. The signals captured are noisy and fluctuate between people. Some of the features extracted are critical in helping for some classes, but not so much for others. For instance, features from gyroscope and orientation sensors are critical for letters ‘J’ and ‘Z’ but they are not so useful for other letters.

Also, the recognition has to be done under many constraints. The system requires a very small number of training instances(5) which contributes to high usability. However, this also means that we only have limited instances of data to learn from with 510 features each. Thus, using out of the box machine learning techniques becomes less feasible. For the same reasons, common dimensionality reduction techniques like Principal Component Analysis (PCA) are not very helpful. Careful selection of features has to be done by relying heavily on the differences among the various signs and how people use them.

The aim of facilitating real-time usage puts a constraint on the complexity of the recognition algorithm and justifies the design choices for parallelization. Although an end-to-end system with server-side recognition is provided, the system’s usability will increase if it is pervasive and ubiquitous. Thus, a lightweight recognition algorithm that can run on a smartphone is desirable. However, processing powers of regular smartphones are limited and thus recognition is much slower than on a server (Figure 4.10). Consequently, a smart and flexible architecture has to be implemented that can still perform in real-time by making appropriate trade-offs between network lag and processing power. This constraint also means that features that are computationally complex to extract, like frequency domain features, could be used.

4.4.2 Video Based

Input: Video from a web cam or smartphone camera

Output: Deep feature extraction output for each hand for each frame.

Challenges:

1. *Localization:* Each of the hands have be to correctly localized for each frame in the video despite differences in frame size, camera resolution, etc..
2. *Comparison:* These images have to be compared to compared efficiently with another image with emphasis on handshape.
3. *Alignment:* Due to differences in signing speed, delays in start or stop times, a correct alignment is needed before the actual comparison.

Constraints:

1. *Real-time constraint:* To facilitate recognition in real-time, the recognition algorithm must be light-weight.
2. *Lack of datasets:* Extensive Datasets that account for all possible handshapes in a sign language are not avaialble.
3. *Phonetic Realization:* The practical realization of handshapes can be somewhat different than the phonetic description of handshape.

Depending on the type of camera system utilized there are different technical challenges associated with the recognition of handshapes. While a computer vision classification algorithm can be trained for each of the possible handshapes in particular sign language, this necessitates the presence of a labeled dataset of all possible handshapes for that sign language. Besides, the phonetic handshapes may not be fully realized in practical usage. For recognition of fingerspelling, the handshape recognition can be confined to the fingerspelling alphabet. In the case of the sign recognition, a dynamic model is needed to account for the changing handshape during the execution of a sign.

4.5 Usage and System Architecture

4.5.1 Sensor Based

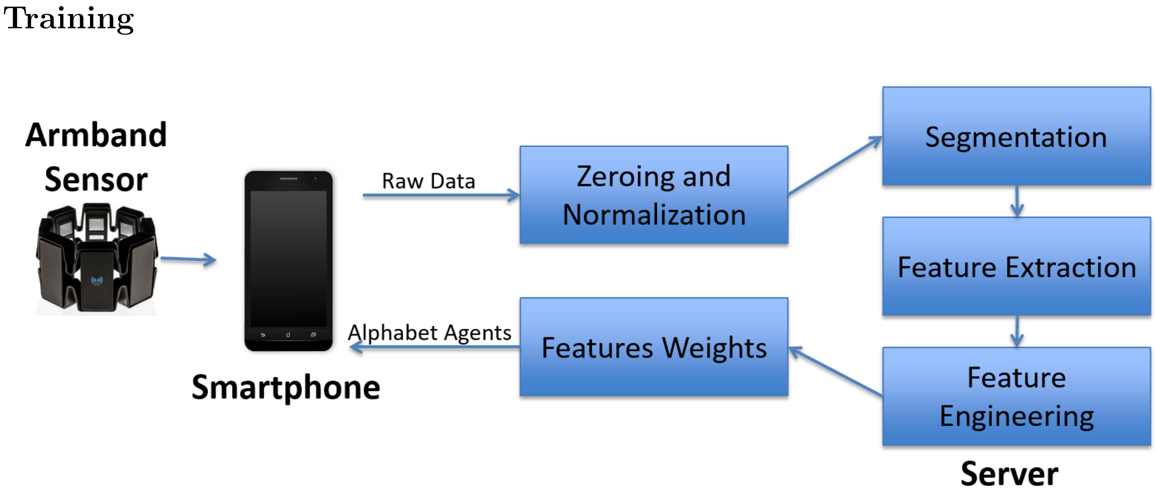


Figure 4.4: System Architecture: Training.

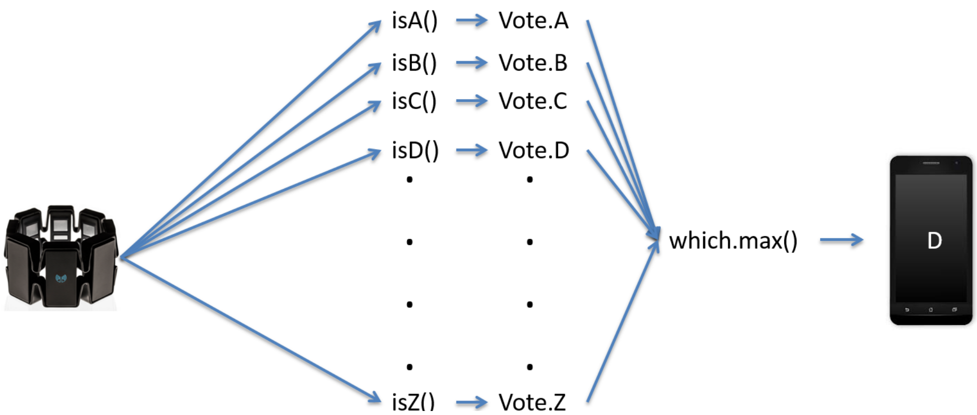


Figure 4.5: System Architecture: Testing.

During training time, the user performs each of the 26 alphabets a total of 5 times while wearing a Myo armband as shown in Figure 4.6. The average total training time for the entire corpus across 9 users was about 10 minutes. While the entire training session could easily be completed in a single session without much fatigue, for best results training should be done in multiple sessions to capture some variations. The training screen (Figure 4.8) has an indicator of connection to the sensor, a progress bar, and a dropdown to select the letter and a guidance animation to assist in training.

All data collected during training is stored in the smartphone and sent to the server via a socket connection. The server performs preprocessing, segmentation, feature extraction, feature engineering and assigns weights to features based on how useful they were in classifying the letters in the training set. Each letter receives a different set of features and each feature, in turn, receives a different weight for each letter. Details on the algorithm are explained in Section 4.7.1. A single file with all relevant information for all of the alphabet agents is sent back to the phone to be used during recognition as seen in Figure 4.4. This is the trained model for the user. The training was done with 9 users(5 females and 4 males) between the ages of 20 and 35 who were taught to sign the letters as part of the experiments.

Recognition

During usage, the test module of the app is brought up to begin translation. Data is collected for 5 s. for each letter and recognition is done on the phone. The trained model that is obtained from the server is saved locally and used by the 26 alphabet agents to get votes from the various top features as explained in Section 4.6.1. These votes are added up to compose the total score and the alphabet with the maximum score is shown as the correct alphabet. This is summarized in Figure 4.5. A simple wave out option is provided for the user. The recognized alphabet is then shown on the screen and spoken out (Figure 4.6). In case of misclassification, the user can offer a correction. This feedback data is recorded and sent to the server for improvements. This correction and feedback feature is optional and can be turned off. For a user who is learning to sign this module can be used as practice.

An end-to-end architecture for offloading the decision is also provided as suggested by Pore *et al.* (2015). The recognition on the server is very fast as can be seen in Figure 4.10. However, there is the added time due to network latency. The decision to offload the computation to the server depends on the availability of network, speed of network, processing power of the smartphone, battery level as well as user preferences.

After recognition, a text-to-speech library is utilized to speak out the letter to facilitate conversation with hearing users. The recognized letter can also be utilized by an application that supports input by dictation.

4.5.2 Video Based

For video-based handshape recognition, a similarity-based technique was used with a pre-trained convolutional neural network as a feature extractor. This technique was utilized jointly with the other recognition modules in the sign language tutor application which is discussed in detail in Chapter 5.

4.6 Technical Challenges

I have discussed the two major approaches to recognizing the shape of the hand: a) Body Sensor-based approaches and b) Camera-based approaches. Attempting to evaluate each of these techniques in real-world systems presents its own set of challenges. Thus, in this section, I outline the technical challenges for the two approaches separately.

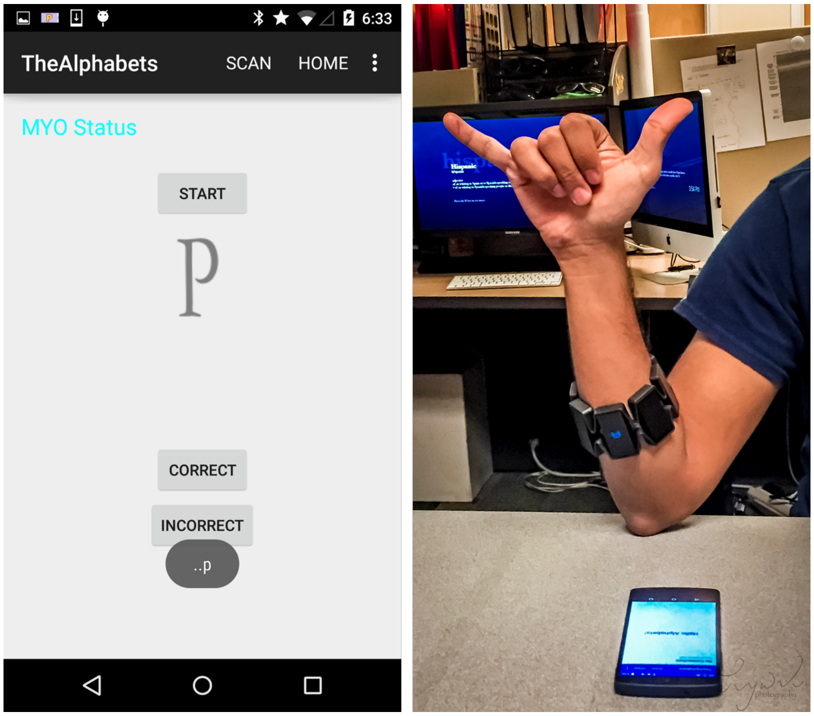


Figure 4.6: Left: Recognition for ‘P’; Right: Testing for ‘Y’.

4.6.1 Sensor based

High accuracy in identifying fingerspelled words has been demonstrated in various systems as outlined in section 4.3, however, most of these systems utilize video, series of still photos or depth cameras Kuznetsova *et al.* (2013). Savur et al. use surface EMG signals to classify letters but the Bio Radio 150 system they utilize has many wires and is not usable in daily use Savur and Sahin (2015). In this work, we propose a novel algorithm that can recognize fingerspelled words using just a smartphone and commercially available wearable armband in real-time and with high accuracy. We envision that contributions from this work will be directly applicable in existing armband based SLR systems that wish to integrate fingerspelling.

An ASLR system should be real-time, non-invasive and pervasive. Fingerspelling has been mostly tackled using image, video, depth camera, data gloves. Some work (summarized in Table 4.2) do use sensors on the arm and armbands as discussed in Section 4.3, however, their accuracy is around 75%. Two inferences can be made from Table 4.2: 1. Fingerspelling recognition has been restricted mostly to image/video or data-glove based approaches 2. The approaches that do use armbands or sensors on the arm either do not support all the letters, are not real-time or are too invasive for daily usage. This work functions to bridge this gap.

Fingerspelling recognition is a very different problem than word recognition and has the following challenges:

Diversity of significant features: Each sign in the AFA utilizes a unique hand configuration. While letters like 'J' and 'Z' show considerable hand movements and others like 'Q' and 'R' have some amount of orientation changes, this is not true for all letters of the alphabet. This means that there usually isn't enough information that can be collected from the accelerometer and gyroscope sensors alone that can help classify between the 26 classes. Hence, a recognition algorithm that focuses only on one modality such as accelerometer, gyroscope, orientation or EMG will have low accuracy Abreu *et al.* (2016). Although the overall structure for signing a letter is specified (Table 4.1), each person has subtle nuances incorrectly executing the sign. Hence, an AFA recognition system should not only be able to fuse multiple modalities but also learn the unique signing patterns of an individual much like handwriting recognition does.

In this paper, we propose Dynamic Feature selection And Voting (DyFAV) algorithm that extracts significant features from accelerometer, gyroscope, orientation and EMG signals and assigns a weight for each of these features such that it is fine-tuned not to the specific patterns that help distinguish between letters, but also to specific nuances in how an individual signs them.

Lack of significant arm movement:

There is a considerable lack of arm movement while performing most of the AFA signs. Thus, classification has to depend on hand configurations, which can be captured using EMG data Abreu *et al.* (2016) Some letters of the ASL alphabet are very closely related to others and differ only very slightly in finger placement. For instance, the letters ‘M’ and ‘N’ only differ from each other by whether the thumb is in between the first and second or the second and third fingers (Figure 4.7).

Further, EMG data can vary a lot between people Kortelainen *et al.* (2012) and feature-selection on this data is a difficult problem since different hand configurations activate different portions of muscles in varying ways. Hence, an AFA recognition system should be able to pick up fine differences at the different portions of the arm to distinguish between these closely related signs. To account for this, the voting mechanism of DyFAV ranks the features such that the ones that were most instrumental in distinguishing a particular sign during training get the highest weights.

A notable difference between recognizing words versus letters is that fingerspelling has a finite corpus i.e. for ASL we have only 26 possible classes. DyFAV takes advantage of this to select a dynamic list of salient features for each letter and lets each of those salient features vote. These votes are adjusted by a dynamic weight for each feature that is learned during training. Details on the DyFAV algorithm can be found in Section 4.7.1 and the recognition accuracy and time analysis can be found in Section 4.8.3.

A study that was performed using traditional machine learning approaches of Adaboost, Multi-Layer Perceptron, Naive Bayes, Random Forest, Support Vector Machine (SVM) with polynomial kernels and Radial Basis Function Kernels gave respective average accuracies of 7.41 %, 85.95%, 80.56%, 91.83 %, 88.47 %, and 18.70. DyFAV on the other hand gave us an average accuracy of 95.36 %.

The contributions of this study can be summarized as the following: (1) DyFAV

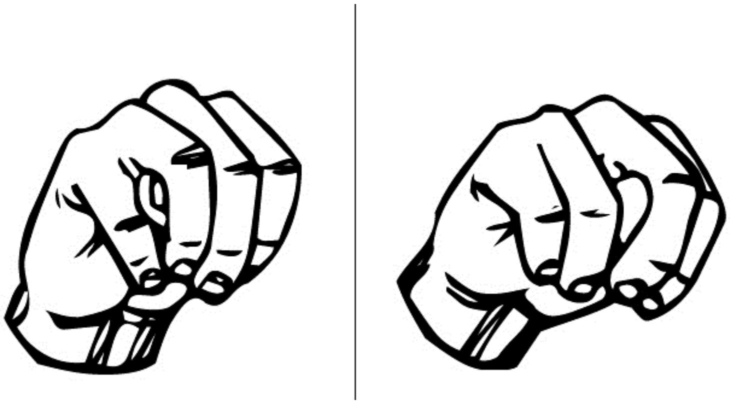


Figure 4.7: Left: ASL sign for ‘M’. Right: ASL sign for ‘N’.

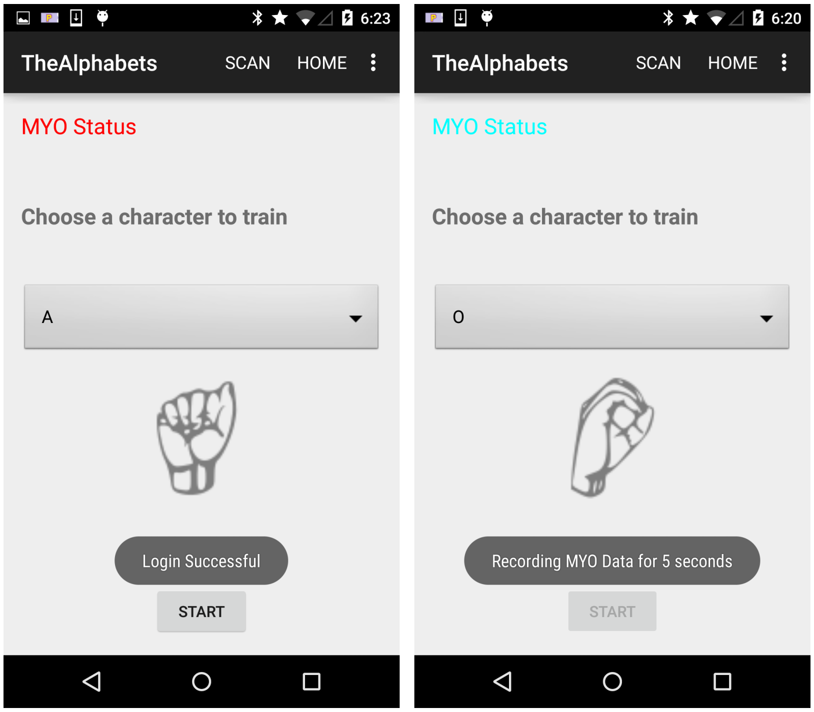
algorithm which uses dynamic feature selection and voting to solve a fixed class classification problem (2) Highly efficient recognition algorithm capable of real-time recognition of fingerspelling using a smartphone (3) A mobile app user-interface that allows training, testing, user-driven correction and collection of user feedback (4) Evaluation of the accuracy and execution time on 9 users using a computer and different types of smartphones.

4.7 Approach and Methodology

In the Section 4.6 I outlined the technical challenges separately for body-sensor and camera based approaches separately. In this section, I present my approach to solving these challenges using a wrist-worn body sensor and then using web-cams.

4.7.1 Sensor based Approach

There are various types of body-sensors that can be used to identify handshape. The choice of a particular sensor usually results in a trade-off between precison and usability. In this work, I provide solutions to technical challenges of trying to recognize a handshape using the Myo sensor which is an armband sensor worn slightly above



MYO Status

Choose a character to train

A

Login Successful

MYO Status

Choose a character to train

O

Recording MYO Data for 5 seconds

Figure 4.8: Left: Training for ‘A’; Right: Training for ‘O’.

the wrist.

Data Collection

Myo device provides four types of raw data: 1. EMG 2. Accelerometer 3. Gyroscope and 4. Orientation.

The raw data for all signals is is obtained by using the ‘Myolib’ Library by darken Darken (2015). Orientation values are received in the form of unit quaternions. The pitch, yaw and roll values are obtained from the quaternion values w, x, y and z by using the following equations:

$$\begin{aligned} roll &= tan^{-1}(\tfrac{2(wx+yz)}{-x^2y^2}) \\ pitch &= sin^{-1}(max(-1, min(1, 2(wy-zx)))) \\ yaw &= tan^{-1}(\tfrac{2(wz+xy)}{-y^2z^2}). \end{aligned} \tag{4.1}$$

Zeroing and Normalization

To account for different starting positions, orientations and rest-level activations of EMG sensors, all sensor readings are zeroed at initialization point. This is done by differencing the average of the first three sensor readings from the rest of the signal. Three readings are used in place of one to minimize noise. In addition, normalization is performed across the entire alphabet corpus and abnormal sensor readings are smoothed. Normalization does not play a role in increasing the accuracy for an individual but does have an impact if it is used with data from other people.

Feature Selection

For each one of the channels, we select five features: 1. Max 2. Min 3. Mean 4. Standard Deviation and 5. Total Energy. Features were chosen to incorporate the variation in data while keeping computation to obtain the features minimal. The EMG has 8 channels and each of the other signals have 3 channels. This gives us a total of 85 features. We then compute the same features, but after segmenting the data into 5 segments. We do this to preserve some time-domain information after feature selection as suggested by Venkateswara *et al.* (2013). Thus, we end up with a total of 510 features for each instance.

Algorithm 4 Feature Engineering

```
procedure Sort by Features(F_D)
    for (i in 1 to 26) do
        for (j in 1 to number_of_features) do
            $sort(F\_D, feature[j])$
            $range[j] \leftarrow range(letters[i], feature\_data)$
            $thres\_lower[j] \leftarrow range[0]$
            $thres\_upper[j] \leftarrow range[1]$
            $weight[j] \leftarrow abs(ln(dif(range) - 4)/(130 - 4))$
        end for
        $n\_min \leftarrow min(weight)$
        $n\_max \leftarrow max(weight)$
        $norm\_weight \leftarrow (weight - n\_min)/(n\_max - n\_min)$
        $write\_feature\_file(feature, range, norm\_weight)$
    end for
end procedure
```

Feature Engineering

Feature data from all instances in the previous step is loaded in a matrix. This gives us a 130 X 510 matrix F_D . This matrix is sorted in ascending order based on the values of each of the 510 features. Each feature then receives a weight based on how well it can perform in sorting each letter in the vicinity of other instances of the same letter. This process is performed iteratively for each feature to get a different weight $wieght[j]$ for each feature for each letter. After completing the entire process, all the weights are normalized. These normalized weights are used during recognition as voting factors (Algorithm 4). After this step, the lower and upper

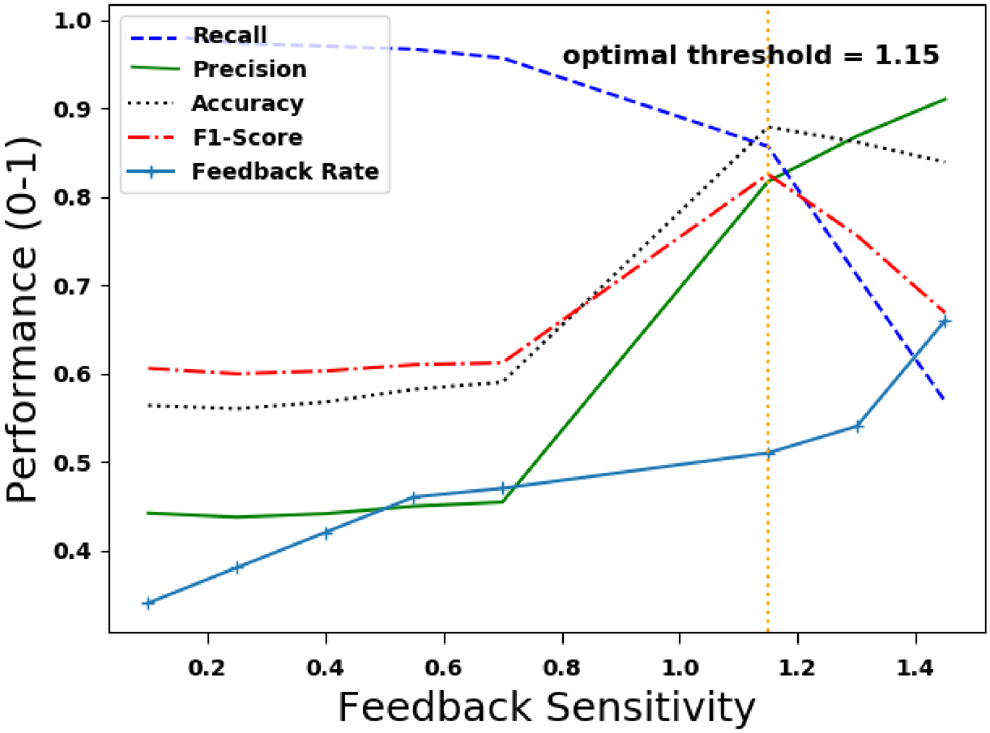


Figure 4.9: Feedback Sensitivity vs. Performance.

thresholds corresponding to each feature is expanded so that values that test features that lie sufficiently close to the actual threshold are still counted towards final voting. This is done to prevent over-fitting to training data. The output of this step is the 'Feature Model' as seen in Table 4.6. It can be seen that the mean standard deviation across the training instance for EMG pod 2 (EMG2_STD0) was the most instrumental feature that could contain all five training instances within a range of 10, thus this particular feature has one of the highest voting weights.

$$adj\_thres\_l = thres\_lower - (thres\_range)/range.$$

Algorithm 5 Test Agent

```
procedure isAlphabet_A(test_features)
    $sum \leftarrow 0$
    $Model\_A \leftarrow read\_model\_A$ from training
    $this\_feature \leftarrow Model\_A[feature]$
    $A\_lower \leftarrow Model\_A[thres\_lower\_adj]$
    $A\_upper \leftarrow Model\_A[thres\_upper\_adj]$
    $norm\_weight \leftarrow Model\_A[norm\_weight]$
    for (this_feature in trained_model_A) do
        if this_feature in range(A_lower, A_upper) then
            $sum \leftarrow sum + norm\_weight$
        end if
    end forreturn $sum$
end procedure
```

4.7.2 Video based Approach

Most signs in ASL have 1-2 distinct handshapes during the duration of the sign. The left and right hands may assume related or independent handshapes. Thus, the recognition of the correct handshape during the execution of a sign has to be done dynamically throughout the sign. Either a dynamic handshape model must be learned for every sign, or an effective way for frame-wise comparison to a gold-standard must be established. In this work, the later is preferred due to its advantages of scalability in vocabulary. The approach is comprised of first localizing various 'keypoints' of the body by using a landmark detection algorithm, then utilizing these landmarks to obtain a crop for each hand for every frame.

4.7.3 Keypoint Estimation

Pose estimation is generally done using estimating the location of various salient points in the body such as the hands, the shoulders, or the torso. These salient body parts that are tracked frame-by-frame throughout a video are generally referred to as keypoints. Robust estimation of keypoint is a necessary step for localizing a sign relative to a signer's body and determining the type of movement. L2S also utilizes the keypoint locations and the confidence level for initial calibration, starting the countdown for recording and determination of handshape crops. We utilize a variant of Residual Neural Network Tai *et al.* (2017) from Tensorflow.js Smilkov *et al.* (2019) called PoseNet Papandreou *et al.* (2017). PoseNet can be used for single as well as multiple pose detection, which means that it can detect many people in the same frame. However, for our purposes, the single pose detection module was sufficient. The ResNet50 powered model was configured with optimum default values considering the performance speed for Output stride(32), image input resolution(257) and enough quant bytes(2) to perform weight quantization. For each frame in a video or continuous stream, PoseNet returns 17 different keypoints throughout the body. Among these, the keypoints for eyes, nose, shoulder, elbows, and wrists were utilized. The eyes and the nose keypoints along with those for the wrists are used initially before the learner begins to record their execution. Along with the 'x' and 'y' coordinates, PoseNet also returns a confidence score between 0 and 1 for each keypoint for every frame. A threshold of 0.6 was utilized to calibrate the webcam positioning and to verify the learner was in range and clearly visible. After verification of about 60 frames, a 3 s timer to begin recording is triggered. As a learner is recording their execution the keypoints for every frame are recorded to be used for a comparison to the tutor for that sign. The keypoints estimated for 5 different body parts throughout the videos for signs 'Spanish' and 'Nice to meet you' are plotted in Figures 2.3 and 2.2 respectively.

Hand Localization and Cropping

The frame-by-frame landmark detection algorithm returns 17 landmarks or keypoints for each frame of the video. The keypoint location for the wrist for each hand is utilized to help obtain a crop for the hands. Due to the differences in alignment of the hand, it is not always possible to use a crop around the wrist location. Thus, a trajectory is calculated that joins the elbow and wrist and the final crop is made along this trajectory as seen in Figure 4.2. Other object localization such as YOLO could also be utilized for that purpose Redmon *et al.* (2016). YOLO is a joint object localization and classification approach which could be trained to find the position of hands and get a bounding box around it jointly. However, the simpler approach of inferring the handshape crop using the localized landmarks of elbow and wrist is preferable since this computation has already been performed. The other reason is that this approach does not require a separate object detection and localization model to be trained which would require manual labeling.

Although there are techniques like Intersection of Union scores that evaluate the performance of object localization algorithms, a separate evaluation for the quality of handshape crops was not performed in this work. This is because algorithms like IOU require a hand-labeled test set for comparisons. Instead, a sample of the training videos was run through the handshape cropping algorithm and the resulting crops were visually inspected.

Alignment

After crops of handshape have been obtained, the next step is to compare the crops to corresponding crops in the other videos. In the case of learner vs. tutor for sign language learning, this would be comparing the student's handshape crops to those of the teacher. For each hand, the dynamic time warping algorithm used for movement comparisons using the wrist bones produces not only an alignment score, but also an alignment path. This algorithm is described in more detail in Section 3.6.1. This alignment path is utilized to align the handshape crops that should be compared to each other.

Transfer Learning

The intuition behind transfer learning is that the lower-level representations that are learned for one computer vision task can be useful to aid the training in another task. It is generally well known that computer vision neural network architectures need a lot of data to train. These approaches are generally not suitable for problems in the absence of a large amount of data. However, the lower to mid-level features which such as edges, shapes, etc. that help recognize classes for a problem that has a lot of training examples can be reused for another classification problem. In the video-based handshape classification approach, I utilize this by first obtaining an Inception-v3 CNN that is trained on the million images in the Imagenet database. Then, I retrained this network on a custom ASL fingerspelling dataset keeping all but the last inception block and all of the fully connected blocks trainable. This means that all of the shallower layer weights were frozen from the Imagenet training. The ASL fingerspelling dataset used for retraining has about 1000 images per class Pugeault and Bowden (2011).

Data Augmentation

Many kinds of data augmentation strategies are proposed in the computer vision community. There has even been some work on automating data augmentation for maximal efficiency Ho *et al.* (2019). For this application, data augmentation in terms of skin-tone variation and random-cropping were applicable, however techniques for horizontal, or vertical flip and random rotations would hurt performance. After the application of gray-scaling, skin-tone variation and random-cropping, about 3000 additional images were obtained per class. This added to the 1000 images gave us a total of 4000 images per class for training the inception model with transfer learning.

Deep Feature Extraction

Computational Image comparison is challenging. The purpose of this step is to get a reasonable approximation of the similarities and differences in the shape of the hand between the two videos by comparing each of the aligned frames. For each of these images, a convolutional neural network(CNN) is used as a feature extractor as seen in Figure 4.3. The CNN utilized here is trained to be able to differentiate between the various handshapes that are found in the ASL vocabulary and is trained on the ASL fingerspelling dataset with a classification accuracy of 96% on that dataset. The output layer of this algorithm is chopped off to obtain a 26-dimensional feature vector for each of these images.

Similarity Measure

Then these feature vectors are compared to each other using a cosine similarity metric and final output is thresholded to determine a match. Although the entire videos could be used, in practice some salient frames are selected from the beginning, middle and end of the videos to extract the crop to compare with the corresponding aligned crop with the second video for a similarity matching. For instance, if we utilized 10 frames for comparison, then we would run a cosine similarity on a 260-dimensional feature vector.

4.8 Results and Evaluation

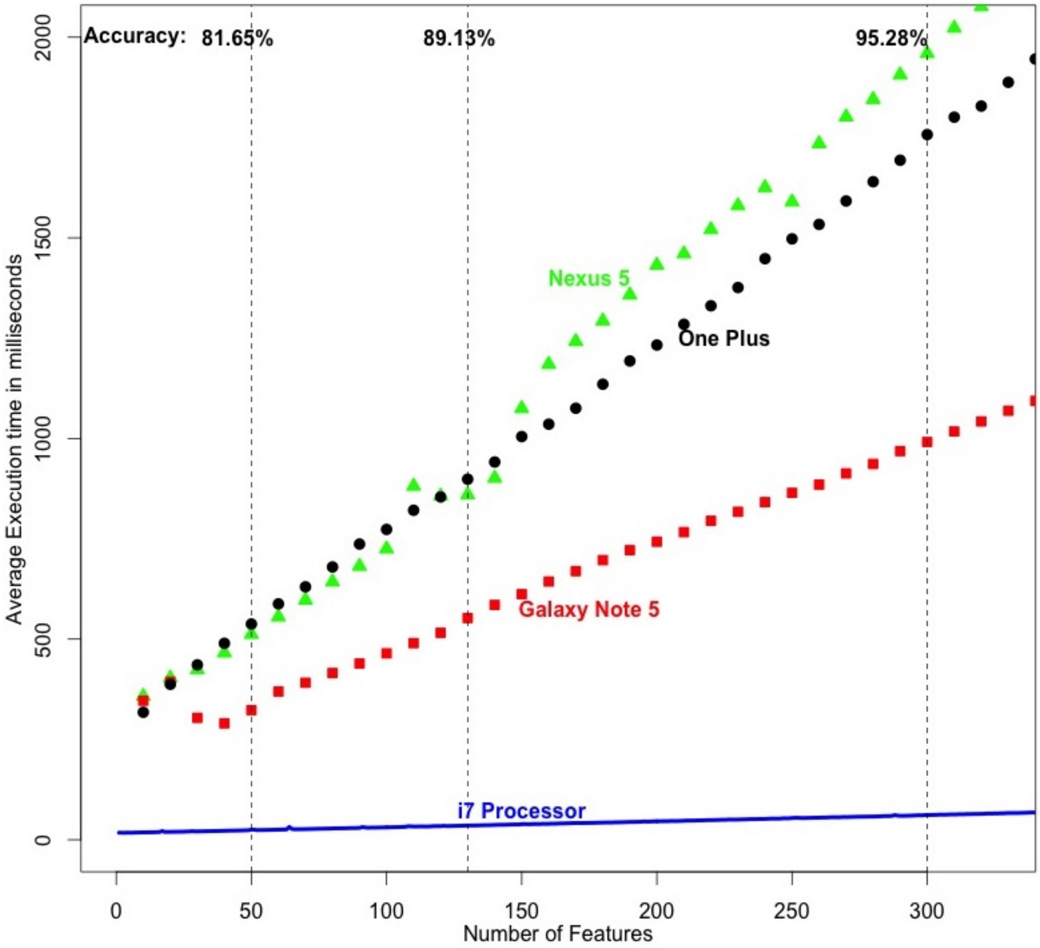


Figure 4.10: Number of features vs. Time: Blue line for Computer and Green Triangles for Android phone.

For the video-based approach is evaluated together with the location and movement to differentiate between the various sign language words and phrases considered. The results are summarized in Table 4.5. To obtain the results, data collected from

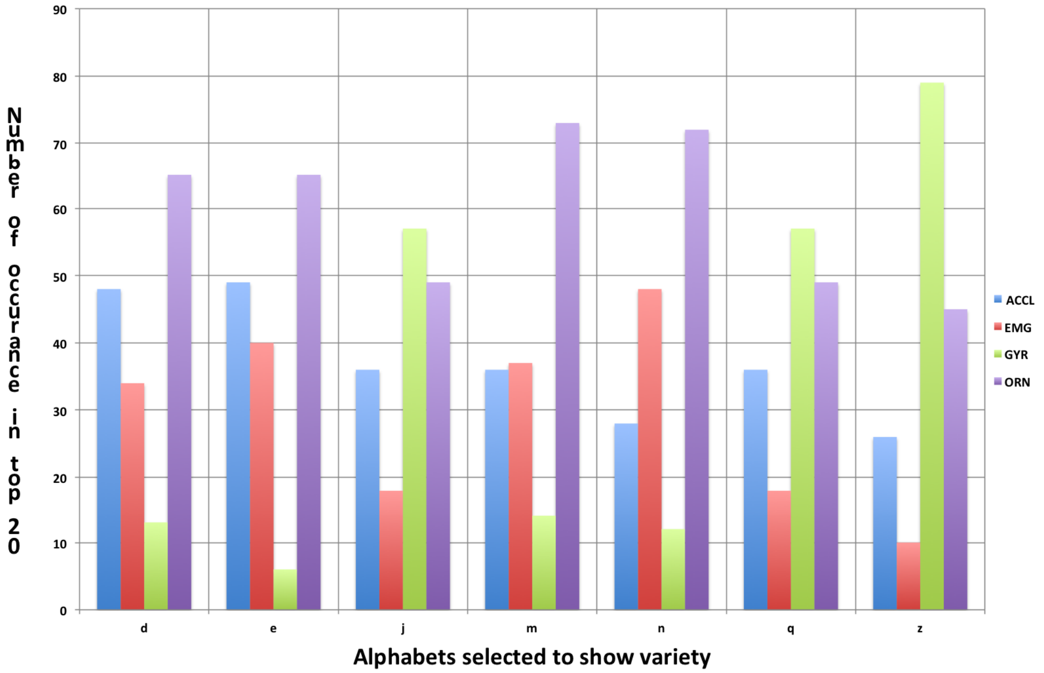


Figure 4.11: Importance of Different Kinds of features for Classifying Various Letters.

one learner was selected at random and served as the ’tutorial’ dataset. Then for each sign for each user in the test dataset, we make comparisons against the corresponding sign in the tutorial dataset. The location module had an overall recall of 96.4% and a precision of 24.3%. The lower precision is because many signs in the test dataset had similar locations. We performed a test comparing only the signs ’LARGE’ to the sign ’FATHER’ and both the precision and recall were 100 %. The movement module had an overall recall of 93.2 % and a precision of 52.4%. The handshape module had a recall of 89 % and a precision of 74 %. The results for only the handshape part is summarized in Table 4.4. The choice of the optimal decision threshold is done to balance the Precision and Recall as seen in Figure 4.9.

For the body sensor based approach, the most important metrics that the system is tested against are recognition speed, usability and accuracy. Experiments were designed with these metrics in mind. The following sections outline the details:

4.8.1 Design of Experiments

Training data was collected in 3 sessions. In each session, the users performed the entire alphabet corpus (A-Z) twice. All training was done in a lab setting using a smartphone app. Users were seated in a chair and were free to rotate or move their hands. The training was done using the first five samples when the remaining one was used for testing. Results were computed using cross-validation. The smartphone app played a GIF image of the alphabet being signed during training to help keep the signs consistent among the users. Video data was collected and later analyzed to make sure the users were using the signs correctly. The device specifications are as follows: Smartphone 1: Nexus 5: 2.26 GHz quad-core Snapdragon 800 processor Smartphone 2: Qualcomm MSM8974AC Snapdragon 801 One Plus One: Quad-core 2.5 GHz Krait 400 Smartphone 3: Samsung Galaxy Note 5: Exynos 7420 Octa: Quad-core 1.5 GHz Cortex-A53 & Quad-core 2.1 GHz Cortex-A57 Computer Specifications(Server): Intel i7, 32 GB RAM

4.8.2 Evaluation

To evaluate the effectiveness of the feature ranking algorithms we performed a test on data collected for all 26 letters for 9 people. Figure 4.17 shows how the accuracy changes as more features are used. The average accuracy across all people is shown in blue. The highest maximum of 100% accuracy across all people is reached using 241 features, the highest average accuracy of 95.36 % is reached using 327 features. It can be seen that the maximum accuracy stalls after a certain number of features and sometimes even reduces. This is because some of the lesser important features that aren't good classifiers slowly contribute to decrease the accuracy. The optimal number of features to be used based on these experiments is determined to be around

327 across users as the accuracy at this number stays the highest.

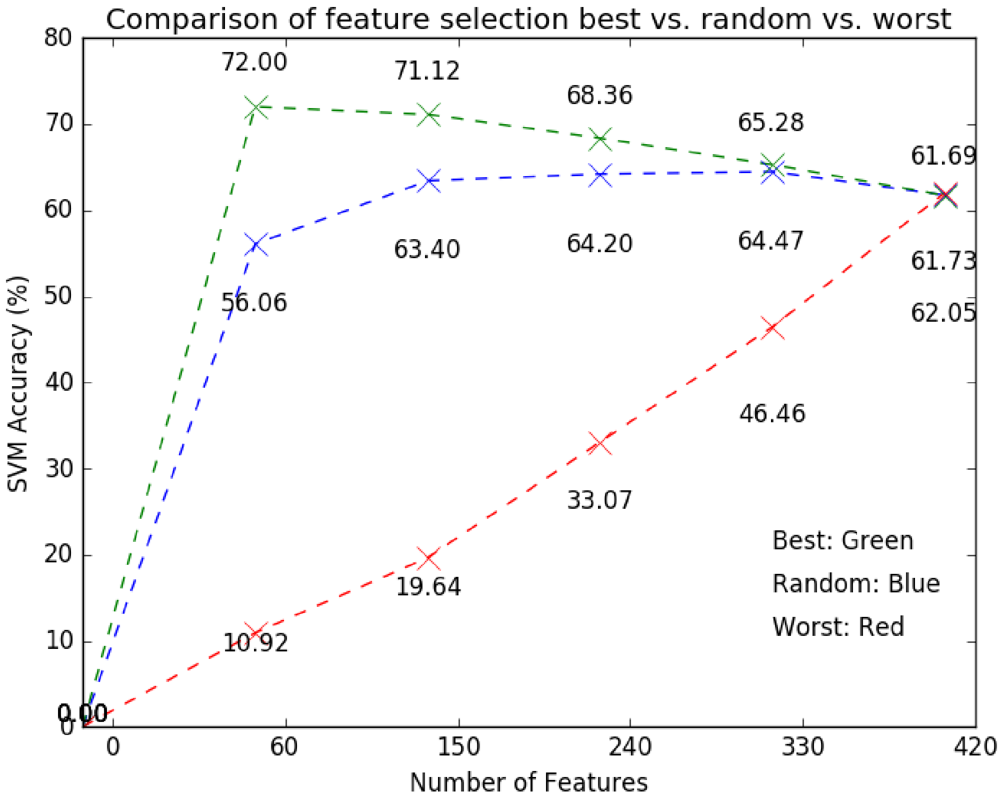


Figure 4.12: Comparison of Feature Selection Best vs. Random vs. Worst.

Figure 4.10 shows how the recognition time varies with the increasing number of features. The blue line represents the average recognition time across users on the server, which has an i7 processor. Three different Android smartphones as specified in Section 4.8.1 were tested with the algorithm for increments of 10 features. The average total execution time on the server for all 510 features was still under 100 ms. and that for 327 features (optimal number) is 67 ms. However, on the smartphone the execution time for 510 features was over 3 s. for Nexus 5 and One Plus while it was just over 2 s. for a Galaxy Note 5. The Note 5 could reach the optimal average accuracy using 327 features in just under 1 s. while it required close to 2 s.

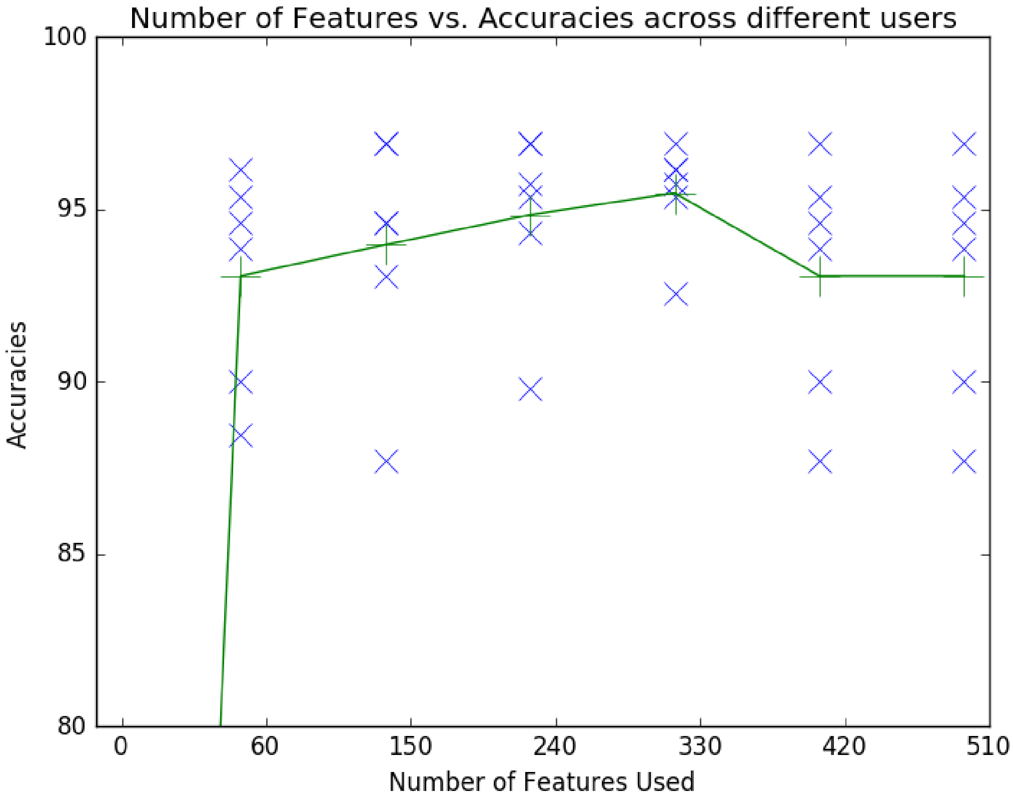


Figure 4.13: Number of Features vs. Accuracy for New Users.

for the Nexus 5 and One Plus phone. Thus, a different optimal number of features should be computed and used for different smartphones based on their computational capabilities while ensuring execution time remains acceptable.

Another way to increase the usability of the system when there are strict timing constraints was to offer close matches in ranked order to the user. She can then do a wave out gesture in the 1 s. window between gestures to select the next one in line instead. The next in line alphabet would appear from the right of the smartphone in gray color before being brought into focus. To test whether such an approach would increase accuracy we tested the probability of the correct letter being returned in a set of 1, 2, 3, 4 and 5. The maximum average accuracies that were achieved were

95.19 %, 98.82%, 99.41 %, 99.84 % and 99.92% respectively. This is summarized in Figure 4.15. The biggest jump in accuracy between the return set of 1 vs. 2 justifies to have this feature present even in a very capable smartphone to ensure the highest performance. This feature will be especially important in smartphones with lower processing speeds. Using just 100 features, the accuracy at a return set of 2 is over 96 % while it is only 87 % for a return set of 1. Also, it should be noted that the accuracy for a return set of 4 quickly converges to practically 100 % when the number of features is 170. Thus, the return set of 5 is not needed. An aggregated confusion matrix across the 9 users is shown in Figure 4.16. It can been seen from this confusion matrix that letters like 'J' and 'Z' have done extremely well due to the inclusion of hand rotation and movement while letters like 'S', 'T' or 'N' are not affected. Intuitively, this makes sense because the letters 'A', 'S' and 'O' are very similar to each other and are thus frequently misclassified.

Figure 4.11 shows how important the different signals were in classifying the letters. The y-axis lists the number of times features related to each of the four signals of accelerometer, EMG, gyroscope, and Orientation appeared in the top 20 features for each of the letters shown. It can be seen that orientation signals are high across the board and accelerometer related features are consistent across all letters but EMG related signals and Gyroscope related signals have more variation. Gyroscope related features are effective to help classify the letters 'J' and 'Z' which is expected due to the high amount of movement that is required for them. On the other hand, letters 'M', 'N, and 'E' or 'A' have few top features that are gyroscope related, but many more that are EMG related. This again makes intuitive sense since these should be classified more by hand configuration rather than rotation information.

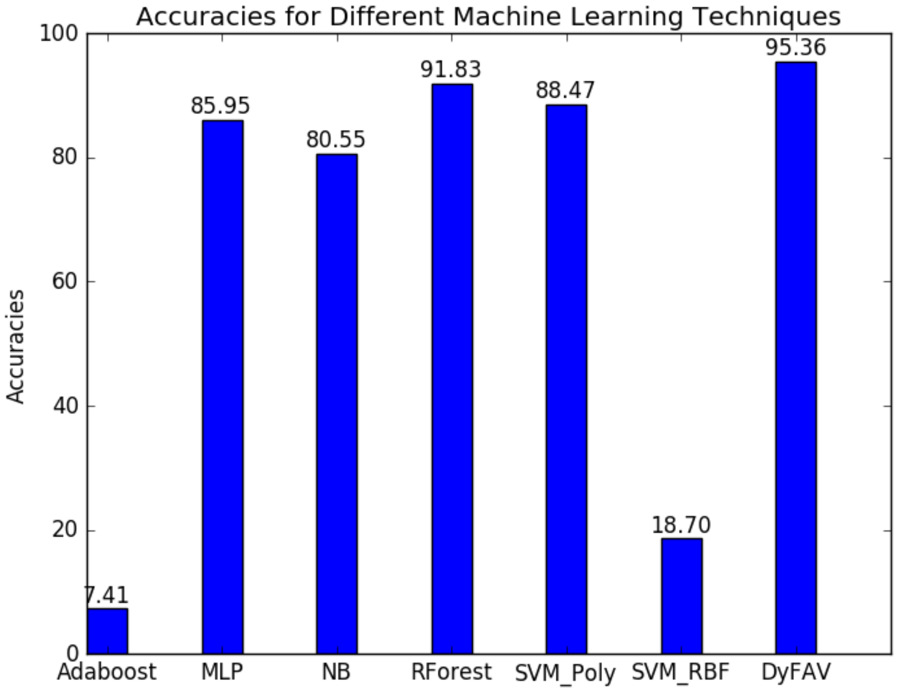


Figure 4.14: Accuracies of different ML techniques on the test dataset.

4.8.3 Algorithm Comparison with other Machine Learning Methods

Figure 4.14 shows the results obtained when comparing the classification accuracies of various machine learning techniques. It is seen that Adaboost does the worst with the average classification accuracy of 7.41% while DyFAV performs the best with the total average accuracy of 95.36%. Among other algorithms compared, Random Forest outperforms other algorithms with a total average accuracy of 91.83 %. The different accuracies across the various test users are also summarized in Table 4.6. It can be seen here that DyFAV performs better than all other algorithms considered except in the case of User 4 and User 6. In the case of User 4, the accuracy is almost the same while in the case of User 6 there is a 4 point difference. A detailed discussion of these results can be found in Section 5.10

4.8.4 Results on New Data

As part of work for this paper, 6 new users were added and similar experiments were designed to acquire training and test data for them. Then following the same algorithms the classification accuracy was computed. Accuracy for the top-5 recall for this dataset across the new users using DyFAV algorithm is summarized in Figure 4.13 it can be seen that the best accuracy is achieved for this users when 330 of the top features are used. The various cross marks on the Figure indicate the individual accuracy of the different test users while the line denotes the average accuracy. The different number of features that were considered were 510, 420, 330, 240, 150 and 60. These features were selected by the DyFAV algorithm to be best suited for classification. It can be noted that similar to results on the prior dataset, the accuracy seems to increase until a certain number of features and then starts to decrease. This is most likely because when too many features are considered that are not important for classification, the model overfits and the test accuracy starts

4.8.5 Feature Engineering Comparison

To evaluate the effectiveness of only the feature selection portion of DyFAV we used the feature engineering part of DyFAV and built n number of linear SVMs to perform a multi-class SVM classification using k number of 'top-features' as determined using DyFAV. A similar experiment was performed using the worst k features as determined by DyFAV. Another experiment was performed using a random subset of k features. The results are summarized in Figure 4.12. It can be seen that the DyFAV feature selection does significantly better than the worst 'k' features as well as does better than the random features up to a certain point. It can also be noted that the overall accuracy increases as more features are included up to a certain point,

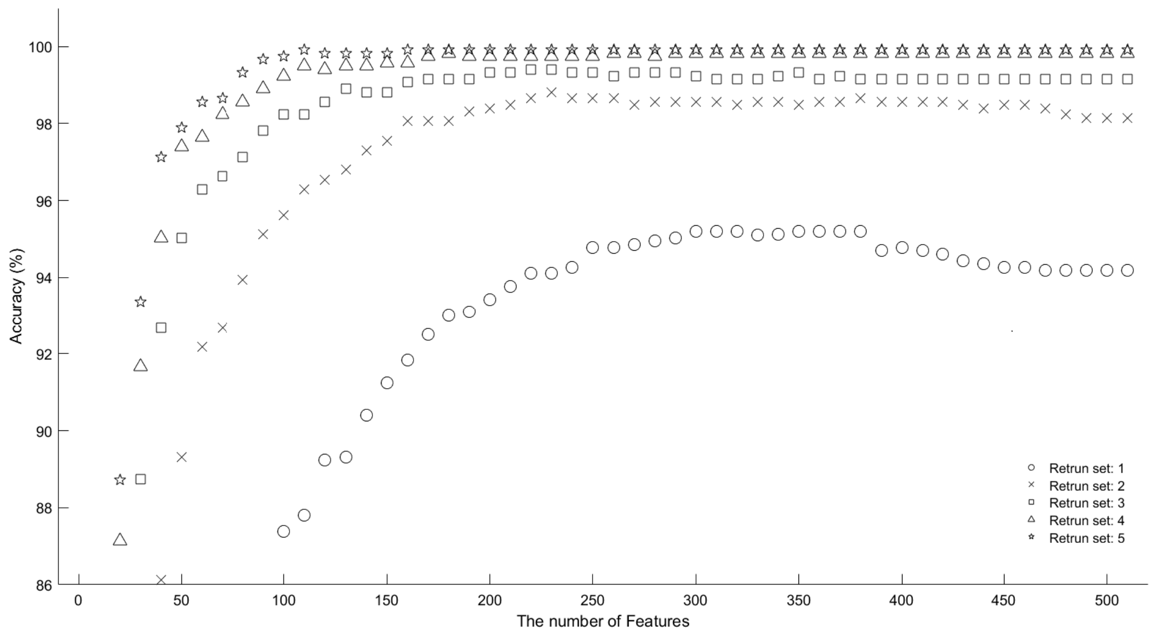


Figure 4.15: Number of Features vs. Accuracy for Different Numbers of Return-sets.

after which it starts to decrease. A sample of the selected features for DyFAV selected features for ‘Q’ is listed in Table 4.3.

Although EMG data varies a lot between people, the voting mechanism of DyFAV ranked the features such that the features that were most instrumental in distinguishing a particular sign during training get the highest weights which helped increase recognition. DyFAV algorithm extracts significant features from the accelerometer, gyroscope, orientation and EMG signals and assigns a weight for each of these features such that it is fine-tuned not only on the specific ways an individual signs the letters, but also takes into account the specific patterns that help identify one alphabet from the other. Although this work contributes a fast and efficient algorithm to recognize fingerspelling, the need for individualized training although very little, is still a hindrance. The data collection is collected in isolation for letters and is thus not very close to suggested levels of fluent ASL signing speeds Quinto-Pozos (2010). The uses of fingerspelling in isolation might not be very significant as it could be replaced by a smartphone notebook or pen/paper, however, in the context of an SLR system as well as a system for ASL based dictation or HCI control, the research becomes highly relevant. Not all features were equally relevant in doing the recognition for all people, thus, custom wearables of the future should focus on the creation of wearables that assist most in SLR if they are to be used for that purpose. DyFAV based gesture recognition algorithm can be utilized in other applications such as diet monitoring application Lee *et al.* (2016) for identifying eating action recognition.

4.8.6 Comparison with other machine learning techniques

Comparison results with other very common machine learning techniques were summarized in the 4.8.3. The machine learning techniques that were chosen to be compared with DyFAV were selected to provide an overview of comparison with other common techniques in the literature as well as to provide insight into the working of DyFAV and why it is successful in this case. Following is a brief overview of each of the techniques that were evaluated and how they compare to DyFAV.

Adaboost

Adaboost short for "Adaptive Boosting" is a machine learning algorithm that has solved many of the practical difficulties of earlier boosting algorithms. It calls a series of given weak or base learning algorithms repeatedly in a series of rounds to give higher weights to misclassified instances to increase the overall accuracy. Freund *et al.* (1999)

Out of all the different machine learning techniques used, Adaboost seems to perform the worst in our dataset. We attribute this to the fact that the training set does not contain enough examples for Adaboost to successfully create many weak classifiers and iterate over them with variable weightage to improve the overall accuracy.

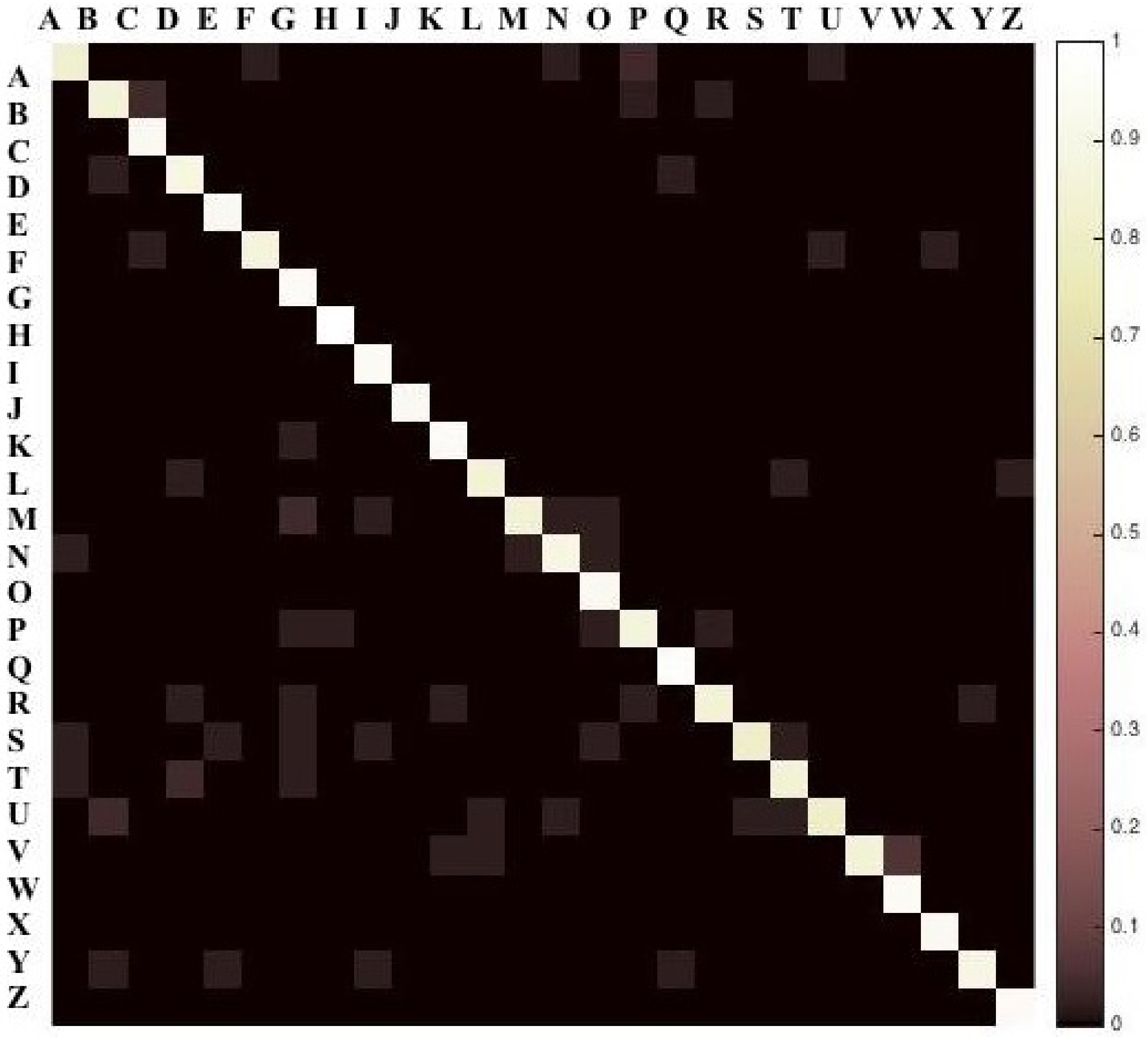


Figure 4.16: Combined Confusion matrix for Alphabet Recognition for 9 Users.

The accuracy that was obtained by Adaboost was just 3.5 % points above that what would have been obtained by random chance.

Multi Layer Perceptron

Multilayer Perceptron (MLP) is a feed-forward artificial neural network that maps a set of input data into a set of output results. An MLP consists of an input layer, an output layer, and at least one hidden layer. For simplicity and to avoid overfitting, we

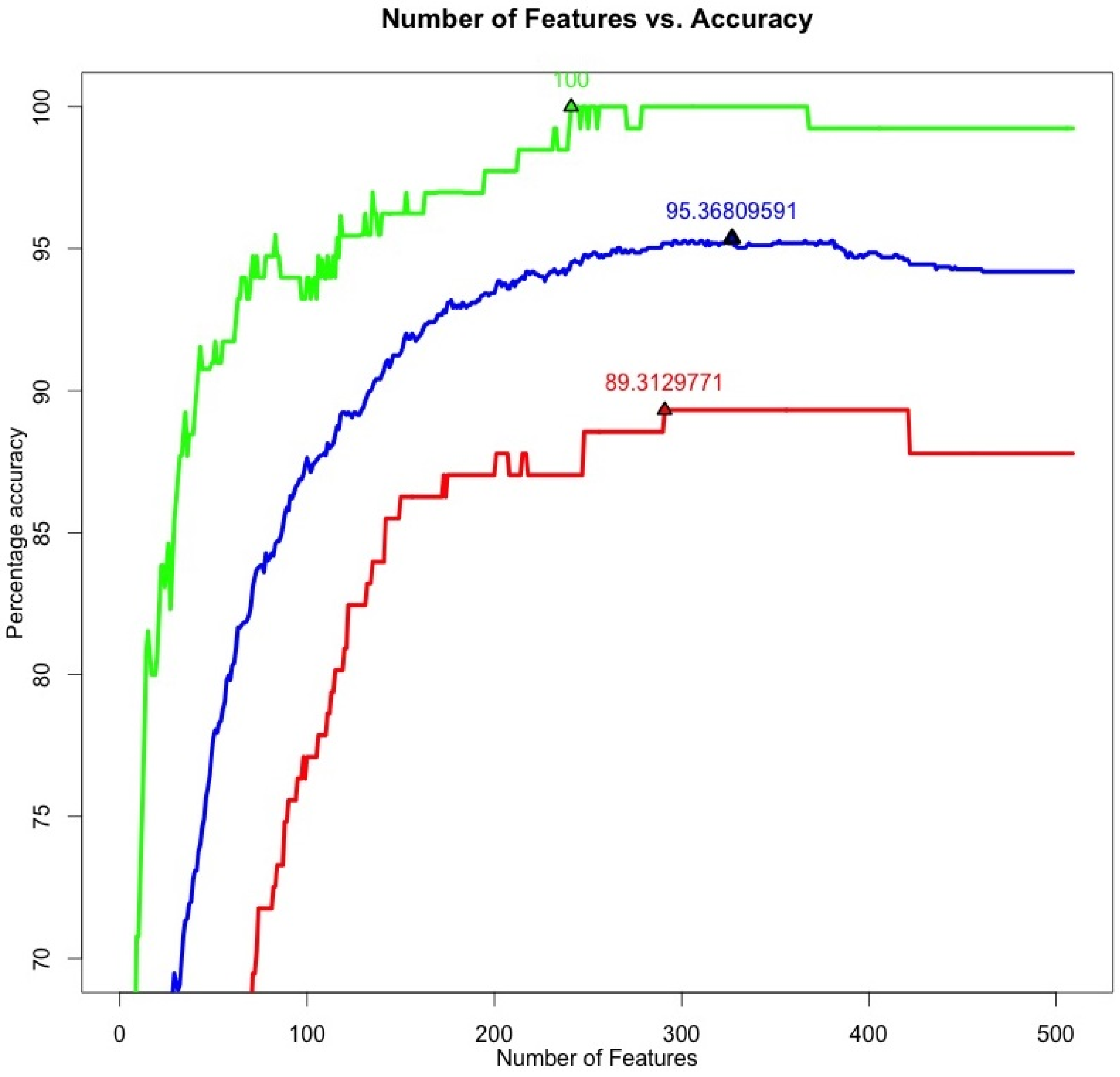


Figure 4.17: Features vs. Accuracy: Green is Maximum; Blue is Average and Red is Minimum.

used an MLP implementation with only 1 hidden layer and 25 neurons. When larger or wider MLPs were tried, the test accuracy suffered most likely due to overfitting.

The accuracy obtained by MLP was 85.96 % which is a very good result.

Naive Bayes

Naive Bayes is one of the most widely used machine learning classification algorithms perhaps due to its simplicity. This classification technique uses a framework that is provided by a simple theorem of probability known as the Bayes' rule. Lewis (1998)

When we utilized Naive Bayes to our dataset the accuracy was just over 80 %. The dataset that we deal with consists of multiple time series data from various sensors and not all of the data is truly independent of each other. For example, there is a very high correlation between neighboring data-points in a time series and sensor information like gyroscope and orientation are inherently inter-related. We postulate that this might be a reason that the NB algorithm does not give us very good performance in our dataset.

Random Forest

Random Forests or random decision forests is a type of ensemble learning method for classification or regression. It operates by constructing multiple decision trees during training and outputting the mode of the output of the various decision trees. Random forests put an additional layer of randomness to traditional methods like bagging because, in addition to the construction of the various trees using different subsamples of the training data, random forests also vary the construction of the classification or regression trees. Liaw and Wiener (2002)

The results we got using Random Forest was the best among all of the other machine learning techniques considered. The ensemble nature of Random Forest ensures that if some features are not optimal as might be the case, some of the trees will ignore those features and the mode of the results still ends up being good. In DyFAV however, we can identify the features that are not instrumental in increasing accuracy and give them lower weights at training time.

SVM Poly and SVM RBF

Support Vector machines are based on the Structural Risk Minimization principle from computational learning theory. Joachims (1998) The goal of Support Vector Machines is to find a hypothesis h for which we can guarantee the lowest true error. The first type of polynomial kernels tried was the polynomial kernel SVM which represents the similarity of vectors (training data) in a feature space that is a polynomial function of the original feature values. Thus, this method not only looks at the values of the features but also the some polynomial combination of these. The second type of kernel considered for the SVM implementation was the RBF kernel which stands for the Radial Basis Function. The RBF kernel has a nice interpretation as a similarity measure and the feature space has an infinite number of dimensions Alpaydin (2014)

The results we obtained for the two different types of kernels for SVM point that the RBF kernel SVM was significantly outperformed by the Poly Kernel. In theory, the accuracy of the SVM RBF kernel could be improved by searching for the correct 'c' and 'gamma' parameters. However, in our experiments, we could not find values that converged for higher accuracy. Nonetheless, both of these methods didn't do as well as Random Forest or DyFAV.

4.8.7 Feature Engineering Comparison

Feature engineering(selection) is an important aspect of any Machine Learning Algorithm. Many algorithms provide feature selection as part of the algorithm and others can be used for feature selection. Learning can be subdivided into two subtasks: deciding which features to use in the presence of irrelevant and extraneous features and deciding how to combine the relevant features to create a model Blum and Langley

(1997). In the experiments that we performed to evaluate the efficacy of the feature engineering part of DyFAV, we decoupled the classification portion of DyFAV from the Feature Selection part. We built 26 different models for the dataset of the 26 different classes. Then from the model, we looked that the top k features and built multiple ploy SVM models for them. Then we plotted them against the number of features selected. We got results for using 60, 150, 240, 330 and 420 features. The average results for the smaller feature subsets for DyFAV-selected features vs. worst features according to DyFAV and a random subset selection of features indicate that DyFAV feature selection does significantly better than the two other scenarios. This shows that the DyFAV algorithm can be used as a feature selector when it is desirable to use only a subset of the available features. This may happen due to concerns regarding execution time or to reduce the burden of data collection and preprocessing.

Fingerspelling is an intrinsic part of sign language and thus systems that aim to be usable in day to day communication must support it. For systems with a limited dictionary of supported words and phrases, fingerspelling recognition can bridge the expressibility gap. Fingerspelling recognition using wearables is a difficult and a unique problem that can benefit from specialized algorithms. We used the proposed algorithm to classify all 26 letters with high accuracy and provide analysis on features used and recognition. The algorithm suggested worked well for this application and we foresee that this can be applied to any other classification problem as well. In the general case, static fingerspelling recognition is equivalent to approaches for handshape recognition, although some specifics will vary. In this work, I presented two very different approaches to recognizing the shape of the hand: one a model-based approach that utilizes armband sensors and the second a model-free approach that utilizes transfer learning, landmark detection, deep feature extraction, and cosine similarity. The utility of each of these approaches will depend on the particular circumstance and the specifics of the application and the intelligent system that the approach is part of.

Table 4.2: Existing work in sign language recognition.

Device	**Work**	**Acc (%)**	**Lexicon**	**FR support**	**Real-Time**	**Invasive**	**Continuous**	**Mobile**
Data Glove	Karthikeyan and Muthulakshmi (2014)	94	NA	5	No	H	No	No
	Sole and Tsoeu (2011)	95	NA	6	NA	M	No	No
	El Hayek *et al.* (2014)	94	NA	26	NA	M	No	No
	Oz and Leu (2011)	81-94	300	26	NA	M	Yes	No
	Singha and Das (2013)	82	300	NA	No	H	Yes	No
Camera	Pugeault and Bowden (2011)	69-75	NA	26	Yes	M	Yes	No
	Kuznetsova *et al.* (2013)	85-97.8	NA	24	No	M	No	No
	Paulino da Silva *et al.* (2014)	99.04	NA	26	No	H	No	No
Armband	Savur and Sahin (2015)(2015)	82.3	NA	26	Yes	L	No	No
	Paudyal *et al.* (2016)	97.72	20	NA	Yes	L	No	Yes
	Abreu *et al.* (2016)	41.95	NA	20	No	L	No	No
	This work (2016)	95.36	NA	26	Yes	Low	No	Yes

Table 4.3: Top 10 Features in Model for Alphabet ‘Q’.

Features	**Range lower**	**Range Upper**	θ **(Lower)**	θ **(Upper)**	**Norm Weight**
EMG5 min3	91	95	-6	-6	1
EMG7 energy5	125	131	1834.8	6883.2	0.7030
EMG0 stdev0	121	129	19.5608	10474.5	0.4435
EMG7 stdev5	124	128	13.2574	63	0.4435
EMG7 energy3	122	131	17.3286	110	0.4435
EMG7 stdev0	120	129	18938.4	121000	0.4435
EMG0 energy0	122	131	14070.2	110	0.4435
EMG7 energy0	114	123	25.2	110	0.4435
EMG0 max2	0	9	-46.4	121000	0.4435
EMG7 min5	116	126	16.2523	23934.5	0.4212

Sign	Recall %	Precision%	Sign	Recall%	Precision%
About	98.03	21.78	After	100.0	20.60
And	94.19	22.70	Goodnight	100.0	17.40
Can	98.82	12.33	Hearing	98.22	13.12
Cop	95.60	06.21	Hello	100.0	07.39
Cost	94.32	15.51	Help	100.0	17.17
Cat	100.0	08.97	Here	100.0	13.63
Day	98.86	11.31	Hospital	98.30	07.89
Deaf	95.51	15.29	Hurt	96.56	15.22
Decide	100.0	28.73	If	98.28	08.10
Father	93.75	07.99	Large	92.86	20.26
Find	95.06	15.67	Sorry	89.29	24.46
Go Out	97.40	14.20	Tiger	100.0	18.30
Gold	100.0	28.27	**Average**	**97.40**	**15.70**

Table 4.4: True Positive and False Negative Rates for Handshape and Orientation Identification

Table 4.5: Precision(P), Recall(R), F-1 Score (F1) and Accuracy(A) for 25 ASL Tokens using Learn2Sign

Sign	**P**	**R**	**F1**	**A**	**Sign**	**P**	**R**	**F1**	**A**	**Sign**	**P**	**R**	**F1**	**A**
About	0.92	0.71	0.80	0.85	**Decide**	0.91	0.55	0.69	0.74	**Here**	0.92	0.96	0.94	0.96
After	0.85	0.92	0.88	0.92	**Father**	0.86	0.86	0.86	0.92	**Hospital**	0.96	0.86	0.91	0.93
And	0.86	0.86	0.86	0.92	**Find**	0.54	0.81	0.65	0.81	**Hurt**	0.81	0.92	0.86	0.91
Can	0.96	0.77	0.85	0.89	**Gold**	0.88	0.81	0.84	0.89	**If**	0.79	0.90	0.84	0.91
Cat	0.96	0.59	0.73	0.78	**Goodnight**	0.96	0.59	0.73	0.78	**Large**	0.96	0.63	0.76	0.79
Cop	0.91	0.91	0.91	0.95	**goout**	0.88	0.85	0.87	0.91	**Sorry**	0.85	1.00	0.92	0.95
Cost	0.85	0.79	0.81	0.87	**Hearing**	0.85	1.00	0.92	0.95	**Tiger**	0.58	0.64	0.61	0.76
Day	0.96	0.80	0.87	0.91	**Hello**	0.96	0.81	0.88	0.91	**Average**	0.86	0.82	0.83	0.88
Deaf	0.56	0.88	0.68	0.83	**Help**	0.88	1.00	0.93	0.96					

Table 4.6: Comparison of DyFAV and Various Machine Learning Techniques Across Users

User	**Adaboost**	**MLP**	**NB**	**Random Forest**	**SVM Poly**	**SVM**	**DyFAV**
1	7.58	68.94	68.94	90.9	84.09	8.33	**96.43**
2	6.87	92.37	85.5	95.41	91.6	18.32	**98.34**
3	7.69	90.77	87.69	94.61	93.85	63.07	**99.25**
4	7.52	90.23	90.98	95.49	88.72	10.52	**95.41**
5	6.87	79.39	65.65	80.92	80.91	20.61	**89.56**
6	7.46	88.05	81.34	91.05	91.05	12.69	**86.41**
7	9.02	83.46	77.44	88.72	84.21	7.52	**91.23**
8	6.82	91.67	81.06	94.7	92.42	10.61	**96.15**
9	6.82	88.64	86.36	94.7	89.39	16.67	**95.42**
Average	**7.41**	**85.95**	**80.55**	**90.9**	**84.09**	**18.70**	**95.36**

Chapter 5

SIGN LANGAUAGE TUTOR

5.1 Introduction

In this work, I have implemented an interactive Artificially Intelligent(AI) tutor that uses feedback to teach sign language words and phrases to learners. Pre-recorded videos from ASL instructors are used as instructions. The application is built to replicate a typical beginner sign language curriculum. Research postulates that it is generally possible to learn sign languages from videos, or even noisy examples Singleton and Newport (2004). However, most learners that use only video resources are rarely successful. First-time learners of any visual language tend to face issues related to correctly replicating signs using perception alone. First, there is natural variation in hand dominance in the population which might lead to noisy inputs. Secondly, signs in most sign languages including ASL are specified for not for the right and left hand, but dominant and non-dominant hands. In addition to these, practicing, making mistakes, and getting feedback helps to cement the muscle memory, coordination, and linguistic concept to better learn a sign langauge Shield and Meier (2018). Video-based instructions were found to be much more successful if coupled with some feedback mechanism Huenerfauth *et al.* (2015). Many research works point out to the positive relationship between feedback given through interaction and the learning performance of second-language learners Lightbown and Spada (1990); Mackey (2006); Banerjee *et al.* (2019). The ability to practice and receive feedback is also a positive aspect of immersive environments for second language learning such as study abroad programs and even classroom environment to some extent Magnan and Back

(2007). Many software applications for spoken languages incorporate some form of feedback to help improve the pronunciation of learners Stone (2016). Applications like DuoLingo also provide interactive chat-bot like environments with feedback to increase immersion Vesselinov and Grego (2012). However, such applications are not available for learners of sign languages. This is in part due to the inherent technical difficulties for providing feedback to sign language learners as discussed in Section 5.3. The important role of practice and feedback for learners of sign language is also well-established Emmorey *et al.* (2009). However, currently to the best of our knowledge, there are no educational tools that exist that provide automatic analysis and feedback for sign language learners. Huenerfauth et al. performed a Wizard of Oz. study and determined that displaying videos to students of their signing, augmented with corrective feedback messages results in better performances Huenerfauth *et al.* (2015). Other research points out to the fact that second language learners of sign languages usually made mistakes on the location, movement, handshape, and/or facial expressions of the signs Rosen (2004). These also correspond to the phonetic components that make up signs in most sign languages including ASL Stokoe (2005). Of these the location, movement, and handshape are generally acquired first Mayberry (2007), thus we choose these three as the concepts to give feedback on. In theory, state-of-the-art video recognition systems Feichtenhofer *et al.* (2016), or sign language recognition systems Cihan Camgoz *et al.* (2018) could be utilized for recognition and towards providing feedback to sign language learners. However, the algorithms that these techniques use are designed to automatically extract low-level features and attributes and classify a given input among a variety of previously known classes. Although the features and attributes these systems extract are useful for the classification problem itself, they are rarely semantically meaningful for providing an explanation or feedback as to why a given input belonged to a certain class i.e. why a given execution

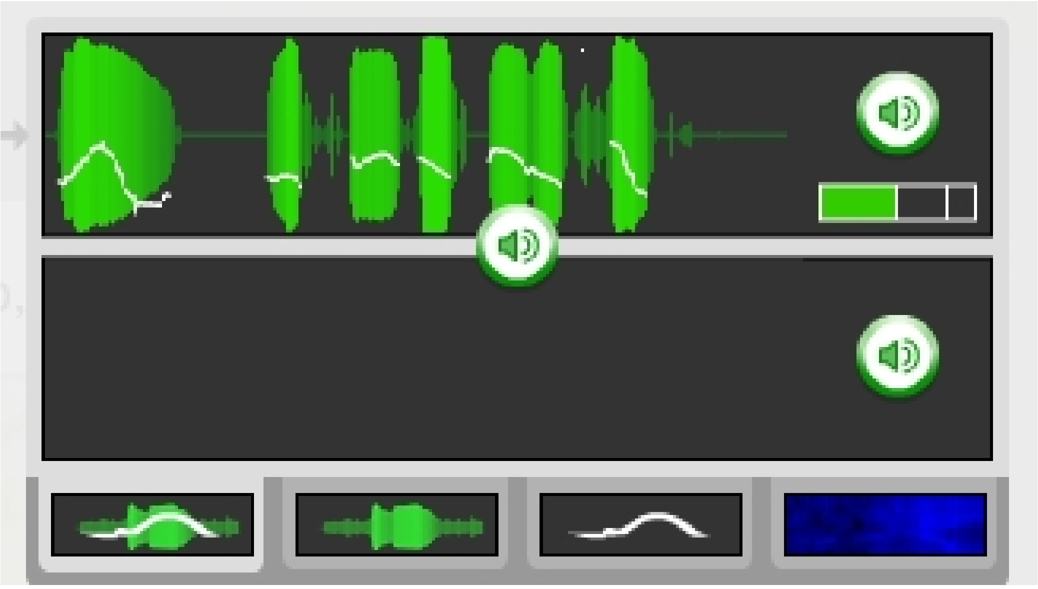


Figure 5.1: Audio feedback for Pronunciation in Rosetta Stone.

was correct or not. Also, these classification techniques do not scale well to unseen vocabularies which would be very important for tutoring applications.

5.2 Significance

World Health Organization estimates that around 466 million people worldwide have disabling hearing loss. This number is estimated to rise to 900 million by 2050 Organization (2018). Signed language are natural languages for deaf or hard of hearing people since only a fraction of the population with disabling hearing loss can benefit from cochlear implants. Other means of communication like lip-reading or writing are not natural for daily life conversations. Many family and friends of deaf or hard of hearing people also benefit from being able to sign. Learning Sign Language is also a very popular choice to fulfill additional language requirements for colleges or high-schools. The Modern Language Association Association (2016) reports that the enrollment in American Sign Language (ASL) courses in the U.S. has increased nearly 6,000 percent since 1990 while that for other languages has been relatively constant as seen in Figure 1.2.

Juan Pablo de Bonet is credited for publishing the first instructional book on sign language in 1620. Since then there have been various books that help people learn Sign Languages Rosen (2010). Unlike spoken languages that have a robust written counterpart, resources on written sign languages are more sparse since learning a visual language from text is not intuitive or practical. For these reasons, most instructional books on sign languages have numerous picture illustrations. Nowadays, most textbooks are also accompanied by video demonstrations as sign languages are best learned through videos or in-person instructions. However, not everybody who wants to learn signed languages has access to a college or in-person classes, thus more and more potential signers are turning to video resources.

There are also numerous tutorials on video-sharing-platforms or websites for learning sign language, especially popular languages like American Sign Language(ASL). Many free or paid smartphone applications provide sign language instructions just like there are for many other languages. However, none of them provide any feedback to the learner as seen in Table 1.2.

One of the essential elements that is missing from video tutorials is feedback. We surveyed 52 learners of American Sign Language and found that 96.2% think that reasonable feedback is important for sign language learning. Thus, meaningful and accurate feedback seems to be a major hurdle for self-learners of sign languages. The age breakdown of the participants is in Table 5.1.

Extended studies show that providing item-based feedback in a computer-based learning environment is very important, especially for language learning Van der Kleij *et al.* (2015). This is the reason that the ideal setting for language learning is immersion in an environment where the primary language is the one being learned, as there is constant positive and negative feedback from the environment Skehan (1998); Paudyal *et al.* (2019a). Classroom settings also offer good environments since feedback can be readily received from the instructor or peers Sheen (2004). This is the reason that extensive language learning applications such as Rosetta Stone or Duolingo support some form of assessment where the learner is allowed to either speak out or write a word or a sentence in the new language, and automatic feedback is provided to them Stone (2016). Both, DuoLingo and Rosetta Stone language learning platforms offer 'Speaking' modules where a learner can record themselves pronouncing various words or sentences in the new language, and they are graded based on them and some feedback is provided as well with some demonstration of how the correct solution looks like as seen in Figure 5.1. These types of feedback if accurate can be important to steer the user to correct their mistakes while making the language learning process more immersive. Studies show that elaborated feedback such as providing meaningful explanations and examples produces a larger effect on learning outcomes than just feedback regarding the correctness of the answer Van der Kleij *et al.* (2015). Sign language learners would also benefit from this type of feedback, however to the best of our knowledge, no such software exists that provides such feedback.

5.3 Technical Challenges

Designing a feedback-driven intelligent tutor has many technical challenges that this work attempts to solve. In this section, I provide these challenges concerning the various aspects of the desired system. These challenges are very much related to the requirements of explainability, ubiquitousness, ability to provide appropriate types of feedback, and the ability to scale to new vocabulary.

5.3.1 Challenge 1: Explainable Systems

A simple notion of the correctness of a sign execution as seen in Figure 5.2 can be computed using existing sign language recognition systems that are discussed in Section 5.4. However, for providing more fine-grained feedback, more details are desirable. This is especially so because sign languages, unlike spoken languages, are multi-modal. Thus, if an error is present in execution, feedback should be given that ties the feedback back to the erroneous articulator(s). For instance, if a student executes the movement part of a sign correctly, and performs the sign in the right position relative to her body, but she fails to articulate the right shape of the hand, then feedback should be given regarding the incorrect handshape. Thus, BlackBox recognition systems are not very useful for feedback, and explainable systems that can recognize the conceptual elements of the language must be developed Lim *et al.* (2019).

5.3.2 Challenge 2: Ubiquitous Recognition

The growing usage of self-paced learning solutions can be attributed to the effect of the economy of scale as well as to their flexibility in schedule. To achieve these desired advantages, the barrier to access must be reduced as much as possible. This implies that requiring the usage of specialized sensors such as 3-D cameras will hinder the utility. Thus, a proposed solution that can truly scale and have the maximum impact as a learning tool must be accessible without the need to purchase special sensors or to attend in special environments. This is challenging because there is a huge variance in the type, quality, and feed of smartphone-based cameras and webcams. Furthermore, assumptions on adequate lighting conditions, orientations, camera facing directions, and other specific configurations cannot be made, and have to either be verified by quality control or accounted for by the recognition and feedback algorithms.

5.3.3 *Challenge 3: Determination of Appropriate Feedback*

Feedback mechanisms for spoken and sign language differ significantly. The differences arise primarily due to the articulators used for speech vs. the ones used for signing. Apart from some research for feedback in rehabilitation for physical therapy, which is conceptually very dissimilar to sign language learning, there are no existing systems in this domain Zhao (2016). Thus, the types of feedback to be given to learners must be determined by referring to the linguistics of sign languages, close work with ASL instructors, and referring to academic studies. Codifying and automating the suggested feedback into a usable system is a challenging process and a worthy research undertaking.

5.3.4 *Challenge 4: Multiple channels and Extension to Unseen Vocabulary*

Sign Language recognition differs from speech recognition in one crucial aspect: the number of articulatory channels. This is partially an artifact of the medium used for recognition, i.e. audio vs video. Audio is usually represented as two-dimensional signals in amplitude and time, while colored videos are four-dimensional signals: three spatial dimensions, one channel dimension for color, and one-time dimension. The consequence of this for Speech CALL systems for spoken language learning such as Rosetta Stone Stone (2016) offers some feedback to a learner based on comparisons between their utterances and those of a native speaker. This one to one comparison to a gold standard is a desirable way for learning systems where the learner is attempting to get close in performance to a tutor. A tutoring system needs to readily extend to new vocabulary as the learner progresses. To extend the capability of a recognition system that is based on a classifier, the entire system will need to be retrained to account for new signs.

5.4 Related Work

There have been many works on providing meaningful feedback for spoken language learners Ehsani and Knodt (1998); Robertson *et al.* (2018); Pennington and Rogerson-Revell (2019). On the practical side, Rosetta Stone provides both waveform and spectrograph feedback for pronunciation mistakes by comparing acoustic waves of a learner to that of a native speaker Stone (2016). There has also been some recent work on design principles for using Automatic Speech Recognition (ASR) techniques to provide feedback for language learners Yu *et al.* (2016). Sign Language Recognition (SLR) is a research field that closely mirrors ASR and can potentially be utilized by systems for sign language learning. However, to the best of our knowledge, no such system exists. This can be explained by the inherent difficulties in SLR as well as the lack of detailed studies on design principles for such systems. In this work, we propose some design principles and an explainable smart system to meet this goal.

5.4.1 Sign Language Recognition

Continuously translating a video recording of a signed language to a spoken language is a very challenging problem and has been tackled recently by various researchers with some success Cihan Camgoz *et al.* (2018). For this application, such complex measures are not desirable, as they mandate extensive datasets for training and large models for translation which decreases their usability. Isolated Sign Language Recognition has the goal of classifying various sign tokens that represent some spoken language words Grobel and Assan (1997); Starner *et al.* (1998); Paudyal *et al.* (2016); Kumar *et al.* (2017). Some researchers have utilized videos Lim *et al.* (2016) while some others have attempted to use wearable sensors Paudyal *et al.* (2016, 2017) with varying performances. In this work, we utilize the insights and advances from such systems to help a new learner acquire the sign language words. To our knowledge, this work is the first attempt at such a practical and much-needed application.

Stokoe proposed that a sign in ASL consists of three parts which combine simultaneously: the tab (location of the sign), the Dez (hand-shape) and the sig (movement) Stokoe (2005). Signs like 'HEADACHE' and 'STOMACH ACHE' that are similar in hand-shape and movement may differ only by the signing location. Similarly, there will be other minimal pairs of signs that differ only by the movement or hand-shape. Following this understanding, L2S is composed of three corresponding recognition and feedback modules.

5.4.2 Explainable Systems

Isolated Sign Language Recognition differs from continuous sign translation Cihan Camgoz *et al.* (2018) in that the latter is meant to recognize and translate continuous sentences. For a sign language tutor for elementary words and phrases, the complexities of continuous translation is not required or desirable. For isolated sign recognition, some researchers have utilized videos Lim *et al.* (2016) while others use wearable sensors Paudyal *et al.* (2016, 2017). These systems, however, are meant to identify particular signs and thus are not appropriate to provide more granular feedback than an overall 'correct' or 'incorrect'. Explainable AI methodologies such as modular composition and explanation interfaces are desirable to provide the kind of finer-grained feedback necessary for an AI tutoring application Paudyal *et al.* (2019c, 2020); Adadi and Berrada (2018); Pieters (2011); Kamzin *et al.* (????). In the case of Learn2Sign, the level of modularity is determined by the concepts of the domain and thus the model explanations can be used directly as feedback to the learner. In this case, the interaction interface for learning is also the explanation interface Pu and Chen (2007) for the underlying model.

5.4.3 Ubiquitous Recognition

Learn2Sign is designed to work across all possible devices. The user studies were performed using multiple laptops without restrictions on lighting or environmental conditions. The application has also been tested to work on smartphone devices or any device with a decent front-facing camera. This requires a robust pose-estimation algorithm that works well across devices. Out of many human pose estimation techniques surveyed Tome *et al.* (2017); Sarafianos *et al.* (2016); Tompson *et al.* (2014), the Tensorflow JS implementation by Papandreou et al Papandreou *et al.* (2017) was selected because it met all these criteria. The recognition in Learn2Sign is a composite outcome of smaller similarity-based comparisons which also contributes to the robustness to changing environmental conditions. For instance, for a system based on a classification model, a stark domain shift such as changes in lighting or background coloration will hurt performance Tzeng *et al.* (2017). However, the performance of Learn2Sign will remain relatively unaffected as long as the confidence levels for pose estimation is maintained. If the pose estimation itself is not confident, Learn2Sign can notify the learner rather than give unwanted results.

5.4.4 Unseen Vocabulary

Sign languages can have thousands of words and even more can be formed to account for special circumstances Paudyal *et al.* (2016). In addition, there are many sign languages in the world. Thus, a system powered by a fixed-class classifier will not scale well and hence will not be very practical. Spoken language learning tools such as Rosetta Stone Stone (2016) utilize a comparative approach to recognition and the feedback is in the form of a visual comparison of the spectrograph. In our prior work Sceptre Paudyal *et al.* (2016), we utilize a similar approach of one-to-one comparison for recognition using dynamically alignment of time series. The movement trajectories of a learner for a particular sign should also be compared to that of a teacher to ensure there is a match. A comparison of movement trajectories is an important research area in object detection and human action understanding. The work by Paudyal et al. deals with comparing movement patterns in ASL and achieves an overall recognition accuracy of 96 % for 20 isolated ASL signs Paudyal *et al.* (2016). While a comparison and path-alignment approach works very well for comparing movement which is comprised of 2D spatial signals, it is generally not applicable for comparing images. Existing object detection techniques could be utilized for frame-by-frame localization of a leaner's hands and this information can be aggregated and compared with that of a teacher. For image to image comparison for similarity, a convolution neural network is utilized as a feature extractor which facilitates a one-to-one comparison as suggested by Chen *et al.* (2016).

5.4.5 Instruments for Assessments

There are many instruments to assess sign language acquisition that have been proposed. These instruments vary manifestly due to the difference in the language being assessed, its specific linguistics, and the level of proficiency. We found that the ASL-PA and the Signed Language Development Checklist can serve as important guidelines for assessments. Some of the other common ones are 1. American Sign Language-Proficiency Assessment (ASL-PA) Maller *et al.* (1999) 2. British Sign Language Receptive Skills Test Haug (2005); Strong and Rudser (1985) 3. Australian Sign Language Receptive Skills TestNiederberger (2008) 4. Signed Language Development Checklist Schembri *et al.* (2002) 5. Assessment for Sign Language of the Netherlands Hermans *et al.* (2009) 6. The Developmental Assessment Checklist for

Sign Language of the Netherlands Marschark and Spencer (2010) 7. Aachen Test for Basic German Sign Language Competence. Haug (2005). Learn2Sign is designed to help new learners of Sign Language acquire vocabulary more efficiently. Currently, Learn2Sign offers the ability to learn and practice individual signs which are analogous to words in spoken languages, whereas, the aforementioned assessment tools go well beyond. Learn2Sign does not currently offer feedback on syntax, semantics, or prosody. Thus the assessments have to be limited to the usage (execution) and retention for the vocabulary items learned. A major takeaway from the survey of the assessment techniques is that production measures and comprehension measures should be tested separately.

Feedback Mechanisms

There are many practical applications as well as academic works Yu *et al.* (2016) on providing meaningful feedback for spoken language learners Robertson *et al.* (2018); Pennington and Rogerson-Revell (2019). Applications like DuoLingo and Rosetta Stone use comparisons with a native speaker to provide waveform and spectrograph feedback for pronunciation Stone (2016). Although, feedback-driven learning applications have been proposed for sign language learning Huenerfauth *et al.* (2015); Paudyal *et al.* (2019c), these systems have not been implemented or tested. The virtual learning environment was proposed by Kelly et. al Kelly *et al.* (2008) is limited in scope since deals only with nine individual letters and requires a colored glove. Our proposed system can give feedback on location, movement, and handshape for any ASL sign for which a tutorial video is available.

5.5 Approach and Methodology

The approach described in this section is to meet the four challenges that were discussed in Section 5.3.

1) **Explanation Systems and Interface**: A concept level explainable AI is proposed that ties the explanations generated to the feedback that is desired to improve learning outcomes. The value from an intelligent tutor application that provides feedback should ultimately be measured by evaluating the impact on learning outcomes. The feedback provided by the learning system is analogous to the model explanations for recognition.

2) **Ubiquitous Recognition and Real-time feedback**: Self-paced learning algorithms should be easily accessible. Learn2Sign is designed to work in mobile or web browsers using only web-cams. One of the implications of this is that only light-weight models are utilized. In addition, the machine learning models are run on the client-side when possible to improve responsiveness and immediate feedback.

3) **Explanation Interface to evaluate feedback mechanisms**: A user-friendly explanation interface is implemented to properly assess good ways of providing feedback while avoiding cognitive overload. The interface is designed as a chatbot for maximum control over interactivity as well as to allow the learner ease in reviewing previously learned material, and tracking their progress as they receive feedback.

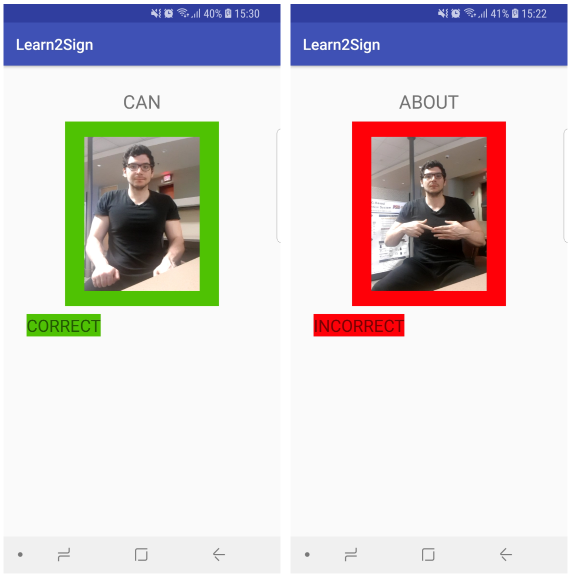
4) **Comparative feedback to a tutor to support Unseen Vocabulary**: This addresses Challenge 4 discussed in Section 5.3 The goals of an intelligent tutor system differs from that of a recognition system. The need for feedback necessitates the need to apply explainable modeling techniques. The other consequence is the underlying approach should not be to classify signs, but to compare them to a ‘gold standard’ that the student is trying to replicate. This ‘gold standard’ is taken to be the execution of

the particular sign by the tutor. This approach has the added benefit of supporting a potentially unlimited vocabulary as the application only compares a student's sign to the video they were learning from.

5.6 System Design

Learn2Sign is designed as an interactive chatbot web application. The server-side is written in python programming language and uses Django as the server framework. The client-side interactions are handled using React which is a Javascript framework. The state of the system is tracked using session variables and a finite-state-machine architecture (FSM) on the server-side as shown in Figure 5.6. The FSM provides a method to gracefully manage state changes, handle exceptions and automatically run the assessments for every available curriculum. A new user begins interacting with the system by signing up for an account using her email address or signs in to her existing account. After this Learn2Sign gives a brief introduction and eventually lists all the curriculum available for learning. Since Learn2Sign is developed to be scalable - a new curriculum can be added by simply uploading new sign videos to the server. All interactions from here on happen either through confirmatory chat messages or via direct clicks on provided buttons as seen in Figure 5.5b.

The first step after introductions is for the learner to select a particular curriculum, from the list of the available curriculum. Any videos that have been pre-processed by extracting PoseNet like keypoints can be made available as a potential curriculum. After the learner selects a curriculum, the system transitions to the next state and waits for the learner to make a selection for the first video she wishes to learn. After that selection is made, the video is sent to the learner and is played on repeat until the learner is satisfied and proceeds either to practice the newly learned sign or to continue learning other signs. The video could be paused, repeated, and viewed in



(a) Correct. (b) Incorrect.

Figure 5.2: Visual Feedback for Correctness.

full-screen. The learner can also scroll up the chat window to view previous activity including rewatching previous videos. If the learner chooses to practice, after each practice session she receives feedback on her execution and is then given the options to either continue with other signs or to retry the same sign. If the learner chooses to retry, the tutor video is for that sign is shown again. This process repeats until the learner is ready to move to the optional assessments section to further cement her understanding as outlined in Figure 5.8.

5.6.1 User Interface

For initial data collection and for testing the UI, we developed an android application called L2S. We preloaded the application with 25 tutorial videos from Signing Savvy corresponding to 25 ASL signs Saavy (2018). The application has three main components: a) Learning Module b) Practice Module, and c) Extension.

Learning Module

The learning module of the L2S application is where all the tutorial videos are accessible. A learner selects an ASL word/phrase to learn and can then view the tutorial videos. The learner can pause, play, and repeat the tutorials as many times as needed. In this module, the learner can also record executions of their signs for self-assessment.

Practice Module

The practice module is designed to give automatic feedback to the learners. A learner selects a sign to practice and sets up their device to record their execution. After this, L2S determines if the learner performed the sign correctly. The result is correct if the sign meets the thresholds for movement, location, and hand-shape and a ‘correct’ feedback is given. If the system determines that the learner did not execute the sign correctly, appropriate feedback is provided as seen in Figure 1.3. Details about the recognition and feedback mechanisms are discussed in Section 5.7.

Extension Module

To extend the supported vocabulary of L2S, a learner can upload one or more tutorial videos from a source of their choosing. The application processes them for usability before they appear in the Learning Module as a new tutorial sign(s).

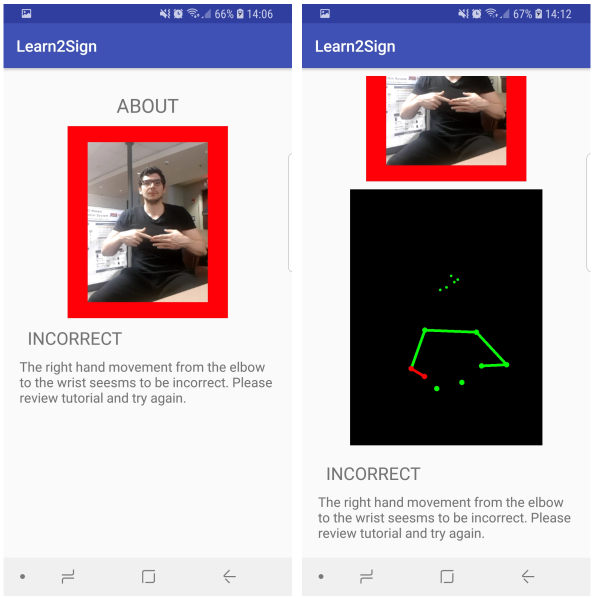
5.6.2 Data Collection

We collected signing videos from 100 learners, for 25 ASL signs with three repetitions each in real-world settings using L2S app. Learners used their own devices, with no restrictions on lighting conditions, distance to the camera or recording pose (sitting or standing up). After reviewing a tutorial video, a learner was given a 5 s setup time before recording a 3 s video using a front-facing camera. Both the tutorial and the newly recorded video were then displayed on the same screen for the user to accept or reject. This self-assessment served not only as a review but it also helped prune incorrect data due to device or timing errors as suggested by the new learner survey in Table 1.1.

5.7 Feedback Mechanisms

A variety of feedback could be constructed using the information available from the results of the location, movement, and handshape modules described above. In addition to separate feedback for each of the hands, feedback could also be presented in forms of annotated images or by using animations. For location feedback, the correct and the incorrect locations for each of the hands could be highlighted in different colors. For the handshape feedback, the image of the hand that resulted in the highest difference in similarity could be presented. Each of these types of possible feedback can be easily derived from the information available. However, they should be individually tested for usability, and care should be taken to not cognitively overload the learner with too much feedback at once. To keep the user interface simple and to better isolate testable components, the feedback given to the learner was kept simple. The feedback that Learn2Sign gave consisted of whether the location, the movement, and the handshape were correct for each of the hands. To draw the learners attention to the feedback she had to pay attention to improve, the negative feedback was sent in red and the positive feedback was sent in green. In contrast, all other chat messages from the system were in gray as seen in Figure 5.5.

L2S is designed to give incremental feedback to learners for the various modalities in sign language: a) Location b) Movement and c) Hand-shape. The various models are arranged in a waterfall architecture as seen in Figure 1.3. If the location of signing was not correct, then immediate feedback is provided and the learner is prompted to



INCORRECT

The right hand movement from the elbow to the wrist seesms to be incorrect. Please review tutorial and try again.

INCORRECT

The right hand movement from the elbow to the wrist seesms to be incorrect. Please review tutorial and try again.

(a) Sentence Feedback. (b) All Feedback

Figure 5.3: Feedback based on Movement (Best viewed in Color).

try again. Similarly, if the movement of the elbows or the wrists for either hand was incorrect, the learner is prompted to try again. Finally, if the shape and orientation of either of the hands does not appear to be correct, a hand-shape based feedback is provided. Consequently, the learner can move on to a practice a new sign, only if all these modalities were sufficiently correct. A waterfall architecture was chosen in the final application over a linear weighted combination to make learning progressive and to decrease the cognitive load on the learner due to the potential of mistakes in multiple modalities. This architecture also helps to reduce the time taken for recognition and feedback since the models are stacked in an increasing order of execution time. Each of the feedback screens shown to the user also has a link to the tutorial video. Users can also manually tune the amount of feedback by altering the value of 'feedback sensitivity' in the application settings. Increasing this value alters the thresholds for each of the sub-modules so that the overall rate of feedback is increased.

This involves a trade-off in performance which is summarized in Figure 4.9.

5.7.1 Location

To correctly and efficiently determine the location for signing, we first assume the shoulders stay fairly stationary throughout the execution of a sign. This is a fair assumption for ASL since there are no minimal pairs exclusively associated with a signer's shoulders. Then we divide the video canvas into 6 different sub-sections called *buckets* as seen in Figure 2.1. Then, as the learner executes any given sign, the location of both the wrist joints is tracked for each bucket resulting in a vector of length 6. With this mechanism, the feedback sensitivity is limited by the number of sub-sections we divide the original canvas in. The granularity was chosen to be both computationally accurate as well as linguistically meaningful. A finer level of location feedback would result in many false negatives during recognition and would require many more comparisons with allowed videos. This is because the phonetic definition of a sign does not always correspond fully to the phonemic realization of the sign-in practice. For instance, a sign that is described to be done in front of the stomach could be still correct if performed slightly higher or lower. Another consideration for choosing 6 divisions of the original video frame was that it is more feasible to track key locations of the human body such as the shoulder joints, but this tracking and partitioning becomes much more nuanced and noisy if body parts such as stomach or cheek are tracked.

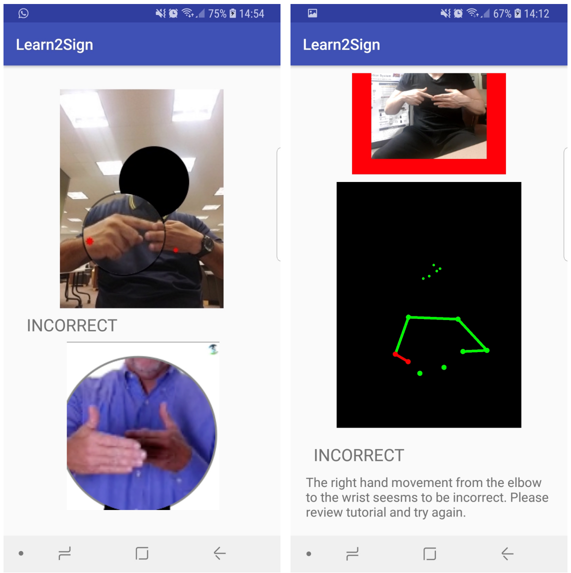
This same procedure is followed for the tutorials, and a cosine-based comparison between is done between the two vectors. A heuristic threshold that is determined during training is utilized as a cut-off point. If the resulting cosine similarity is lower than a threshold, some feedback is shown to the learner as seen in Figure 5.4. For each hand, the user's video is replayed in Graphics Interchange Format (GIF) with a red highlight on the location section that was incorrect and a green highlight on the section of the frame where the sign should have been executed. A text feedback with details and a link to the tutorial is also provided and the learner is prompted to try again.

5.7.2 Movement

Determination of correct movement is perhaps the single most important feedback we can provide to a learner. We compute a segmental DTW distance between a learner and the tutorial using keypoints for the wrists, elbows, and shoulders as suggested in Anguera *et al.* (2010). Normalization as discussed in Section 5.8.2 was found to be very important. Experimental results showed that segmental DTW outperformed DTW or Global Alignment Kernel (GAK). The granularity of movement feedback, or its sensitivity is determined by two factors: 1) decision threshold is chosen and 2) precision of the tracking mechanism. The decision threshold for movement was chosen by hyperparameter tuning to maximize the F-1 score of the model and the other hyper-parameters of the model are chosen to maximize the skill (as measured by area under the ROC curve) of the decision model. For comparing movement the wrist joint of the learner is tracked and is compared to the trajectory taken by the wrist bone of the tutor. The wrist-joint is a very common key-point that is tracked for human action recognition. However, there is also some lack of precision in this case due to the inability to track palm movement that happens independently of the arm movement. To obtain a more granular level of movement tracking and feedback, hand-glove based techniques can be used. However, this would result in a more invasive system with additional device requirements. The dataset had a wide variation in the number of frames per video. It was found that this affected the distance scores adversely. Thus, as an additional step of preprocessing, the video with the higher number of frames was down-sampled before comparison and the segmental DTW is utilized to find the best sub-sample matching. Thresholds for the signs were determined experimentally using 10 training videos for each sign. If segmental DTW distance between a learner's recording and a tutorial was higher than the threshold for each arm section, then a movement-based feedback is provided as seen in Figure 1.3. A GIF is replayed to the user with the section(s) of the arm for which the movement was incorrect in red as seen in Figure 5.7b. A textual feedback is also generated with an explanation after which the user is prompted to watch the tutorial and try again.

5.7.3 Hand Shape and Orientation

ASL signs which are otherwise similar, may differ only by the shape or orientation of the hands. Since, CNNs have state-of-the-art image recognition results, we utilized Inception v3 or Mobilenet CNN depending on the device being used. A model that



The right hand movement from the elbow to the wrist seesms to be incorrect. Please review tutorial and try again.

(a) Hand-shape feedback for AFTER.

(b) Movement Feedback for ABOUT.

Figure 5.4: Feedback Given by the App.

was pre-trained on ImageNet is retrained using hand-shape images from the training users. The wrist location obtained during pre-processing was used as a guide to auto-crop these hand-shape images. During recognition time, hand-shape images from each hand are extracted automatically in a similar way from a learner's recording. Then 6 images for each hand are passed separately through the CNN and the softmax layer is obtained and is concatenated together as seen in Figure 1.3. Similar processing is done on the tutorial video to obtain a vector of the same length. Then a cosine similarity is calculated on the resultant vector. If the similarity between a learner's sign and that of a tutorial is above a set threshold for a sign, then the execution is determined to be correct, otherwise the hand-shape based feedback as seen in Figure 5.7 is provided. The technique used here for recognition and feedback for handshape is based on a general comparison of the similarities of the shape of a learner's hand to that of a tutor at a time-aligned section of a sign video. This strikes a balance between trying to match the handshape exactly as some geometry-based models try to do vs. only comparing more coarse features like movement trajectories and the location of signing. This also follows from the fact that a phonemic realization of a handshape may not exactly reflect a phonetic definition of it for a particular sign. In addition, there is also a need to balance the sensitivity and specificity of the handshape model. In other words, a model that accepts only a very precise realization of a handshape will suffer from a high false-negative rate and will reject variations caused due to incomplete realizations of a handshape, movement artifacts, or slight differences in anatomy. Indeed for other applications where exact matches of handshapes are desired, geometry-based models could be employed Priyal and Bora (2013). Although the retrained CNN could theoretically be used as a classifier, we use it only as a feature extractor for cosine similarity to ensure that the system can extend to unseen classes. A new tutorial can then be effectively added to the system without the need for retraining. An analysis of the effectiveness of hand-shape and orientation recognizer is provided in Section 3.7.Similar to location and movement, feedback for handshape and orientation is also provided in the form of a replay GIF and text. A zoomed-in image of the incorrect hand shape is shown side by side with the correct image from a tutorial as seen in Figure 5.7(a).

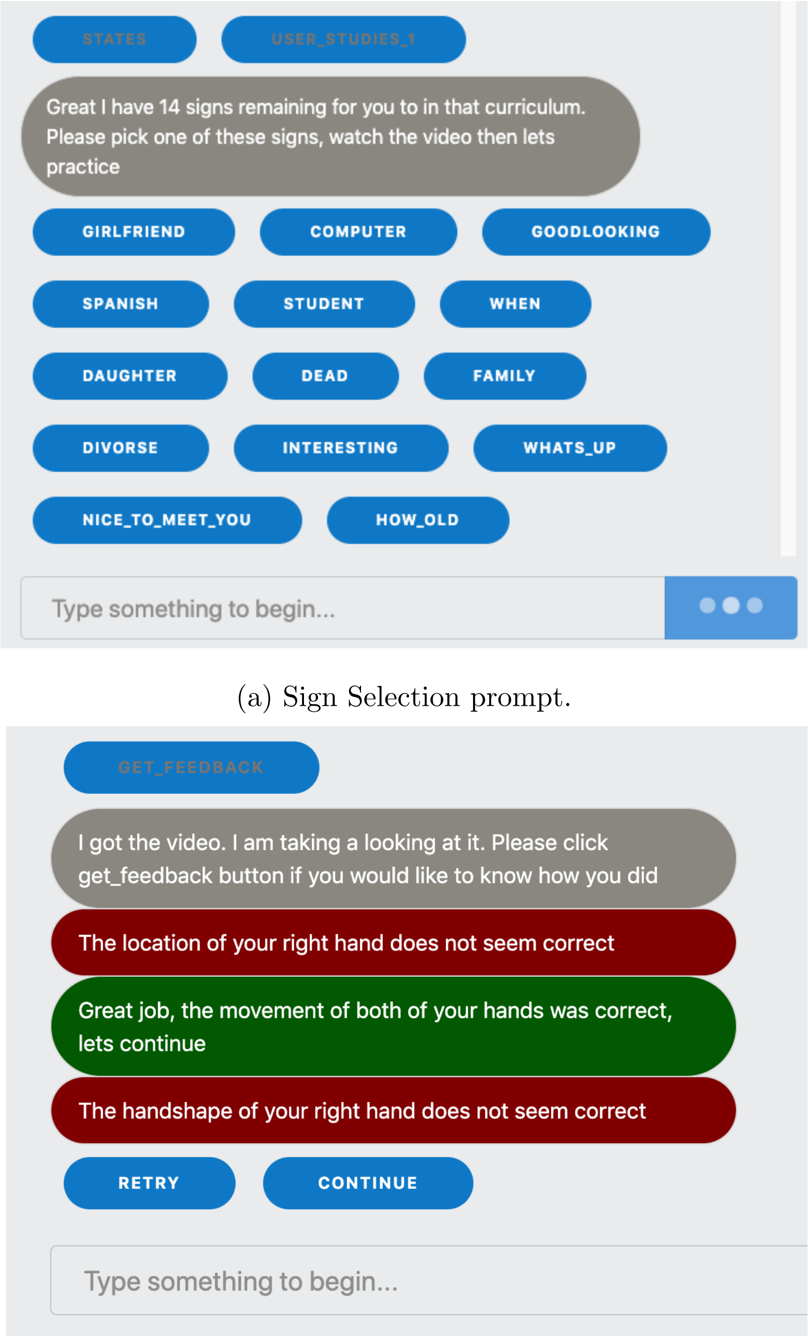
5.7.4 Combination and Thresholding

Different machine learning models for the same task can be compared to each other using the AUROC (Area Under the Receiver Operating Characteristics). AUROC is a measure of the performance of a classification model at various threshold settings Bradley (1997). While the AUROC metric can be used to choose a particular classifier over another one, the nature of the ROC curve itself helps us decide a threshold value. The ROC curve characterizes a classifier by plotting the True-Positive-Rate (y-axis) against the False-Positive-Rate (x-axis). The goal of a classifier would be to maximize the True-Positive-Rate while minimizing the False-Positive-Rate. In other words, if a learner performed a sign with the correct location, movement and handshape (orientation), the application should simply output correct (True Positive) and on the contrary, if either of the location, movement or hand-shape was incorrect, the learner should be alerted and given appropriate feedback (True Negative). The system can make two types of errors: 1) When the execution was correct, but the learner received feedback (False Negative) and 2) When the execution was incorrect, but the learner did not receive any feedback. The thresholds were chosen to maximize the F-1 Scores as shown in the Table 4.5. However, the value of the threshold may also be altered to trade-off one kind of possible error for another depending on the preference of the tutor or to better fit the needs of the learner. For instance, a higher threshold could be selected for applications or a curriculum where it is of critical importance for the learner's execution to be as close as possible to that of the tutor. If on the other hand if false-negative results hampers the user-experience too much, a lower threshold can be selected.

In the case of L2S, feedback is also given for each one of the modalities that the learner was incorrect in. However, as discussed in Section 5.6.1, the feedback should also not cause cognitive overload. Thus, in the process for determination of whether a sign is incorrect, we perform the detection incrementally. First, the location is verified to be correct, then the movement is analyzed, after which the hand-shapes are analyzed. If the location was incorrect, only the location feedback is provided until the learner can correct that, after which movement feedback if needed is provided. This is explained by Figure 1.3. This incremental process helps lower the cognitive load as well as provides the learner a progressive learning model. This is applicable specifically in educational systems since the cost of a false negative, which potentially leads to an additional round of practice, is not that very adverse while the cost of accepting a faulty execution may be higher since a learner might learn a sign incorrectly.

5.8 Design of Experiments

For this work, the user studies were designed to test a single hypothesis that interactive learning with feedback improves learning outcomes. A within-subject study was chosen and 26 University students (10 F, 16 M) with no prior formal experience with sign languages were recruited. 14 ASL signs were chosen from the available vocabulary to represent a good distribution for signs with varied locations, movements, and handshapes. The signs were chosen from the beginner vocabulary of the Signing Naturally textbook Smith *et al.* (2001). Almost all of the words chosen as listed in Table 4.5 were single words. Two words, 'go out' and 'good night' were



Great I have 14 signs remaining for you to in that curriculum. Please pick one of these signs, watch the video then lets practice

(a) Sign Selection prompt.

I got the video. I am taking a looking at it. Please click get_feedback button if you would like to know how you did

The location of your right hand does not seem correct

Great job, the movement of both of your hands was correct, lets continue

The handshape of your right hand does not seem correct

(b) Automatic feedback.

Figure 5.5: Learn2Sign interactive chat-bot interface. Right: In this case, the movement of both the hands were correct (green), but the location and handshape of the right hand were not correct. Feedback is supressed for the correct location and handshape for the left hand to prevent cognitive overload.

Age Bracket	**Percent**
18-21	26.9
21-25	42.3
26-30	15.4
31-40	15.4
>40	0

Table 5.1: Age Breakdown of Survey Participants

composite words. However, the instructional videos utilized for them did not have a perceivable pause between the words, and were thus treated as single words for learning and practice purposes. Testing with more composite words and accounting for the effects of co-articulation as well as pauses in movement is left for future work. All of the learners were asked to interact freely with the Learn2Sign application and a researcher was on standby to answer any questions. The participants were reminded that they would be assessed on the signs learned at the end of the study and given some instructions. After this, the user study followed the same state transitions as outlined in Section 5.6 and summarized in Figures 5.6 and 5.8 except for two crucial changes: 1) For every test subject, half of the 14 signs were randomly chosen to be practiced at least once and the other half were chosen to be continued without practice. The learners were free to practice and receive feedback as many times as they wished by clicking 'retry' after they received the feedback. 2) The assessments at the end of completing the curriculum were mandatory and consisted of 6 randomly chosen retention tests and the remaining 8 as execution tests. All user study participants were also required to fill a post-usage survey.

Some of the learners chose to replay to video multiple times and mimic the signs

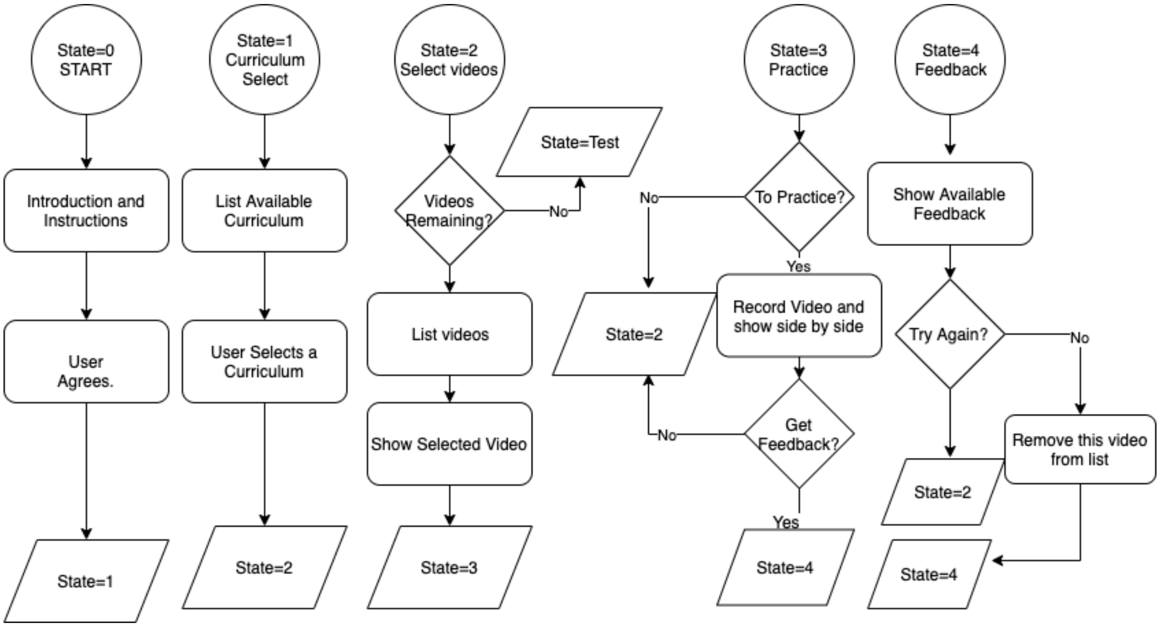


Figure 5.6: System Design as a Finite-State-Machine

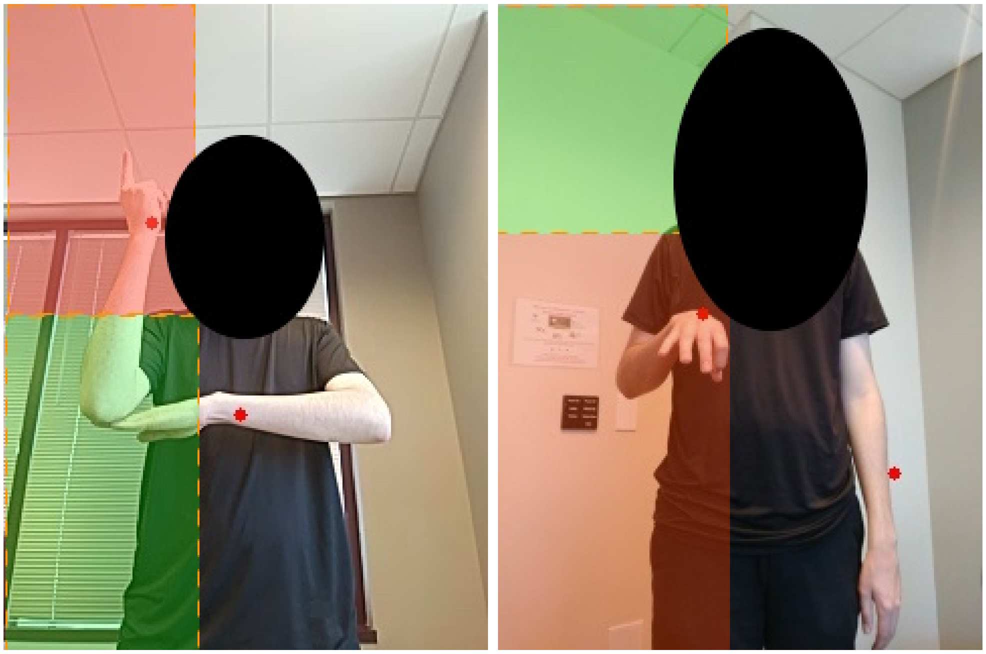
before proceeding. For half of the signs a 'practice now' button appeared and the learner had to send a recording of themselves performing the sign they just learned. Learners spent between 12 and 25 minutes interacting with the application as seen in Figure 5.9. After this step, the application provides feedback on the execution. Details on the feedback are provided in Section 5.7. For the other half of the signs, no such feedback was provided. The remaining signs are populated again as a list of buttons. After all of the signs are learned in this manner, the retention tests begin. Six out of the 14 signs were chosen randomly for retention tests. A video for the retention test was displayed with four options, one of which was correct. After the learner picked the answer the next retention test was shown. After all the retention tests were complete Learn2Sign proceeded to administer the execution tests. The remaining 8 signs from the curriculum where prompted on the screen one at a time and the learner recorder her execution before moving on to the next one until all execution tests were complete. After this, a link was sent for the postcompletion usage study. The submission of the survey marked the end of the user studies. During the assessment period, the participants were not given any feedback on their choices or their execution. The multiple-choice retention tests were auto-graded by Learn2Sign and results were stored as 0 for every incorrect choice and 1 for every correct choice. The videos for the execution tests were recorded and were graded offline. The execution tests resulted in a score of 0.5 for each sign if two of location, movement, and handshape were correct and 1 if all three were correct. Otherwise, the execution received a score of 0. The grading was done using a blind review process such that the grader did not know whether the sign being graded was or was not practiced. The results of the user studies are summarized in Section 5.9.

5.8.1 Data Collection

The methodology and evaluations are provided in Sections 5.5 and 3.7. As part of the work, we collected video data from 300 users executing 25 ASL signs three times each. The videos were recorded by L2S users in real-world settings without restrictions on device-type, lighting conditions, distance to the camera, or recording pose (sitting or standing up). This was to ensure generalization to real-world conditions, however, this makes the dataset more challenging. More details about the resulting dataset of about 15000 instances can be found in Lab (2018)

5.8.2 Preprocessing

Determining joint locations: Since, different devices record in different resolutions, all videos for learning, practice or extension are first converted to a 320*240 resolution. Then, PoseNet Javascript API for single pose estimation Papandreou *et al.* (2017) was used to compute the estimated locations and confidence levels for the various keypoints as seen in Table 2.1. Figure 2.1 shows the estimated eyes,



(a) HERE: Red box(upper): Detected Location, Green Box(lower): Correct Location.

(b) DEAF: Red box(upper): Detected Location, Green Box(lower): Correct Location.

Figure 5.7: Feedback for Incorrect Location for Right Hand.

shoulder and wrist locations for the signs TIGER and DECIDE for all the frames in one video.

Normalization: There is a difference in scale of the bodies relative to the frame-size corresponding to the distance between the learner and the camera. This scaling factor can negatively impact recognition since the relative location, movement, and handshape will vary with distance. We perform min-max normalization and zeroing based on the distance between the average estimated locations for the right and left shoulders throughout the video frame as suggested by Nadil *et al.* (2016). Normalization was found to be especially important for correct movement recognition.

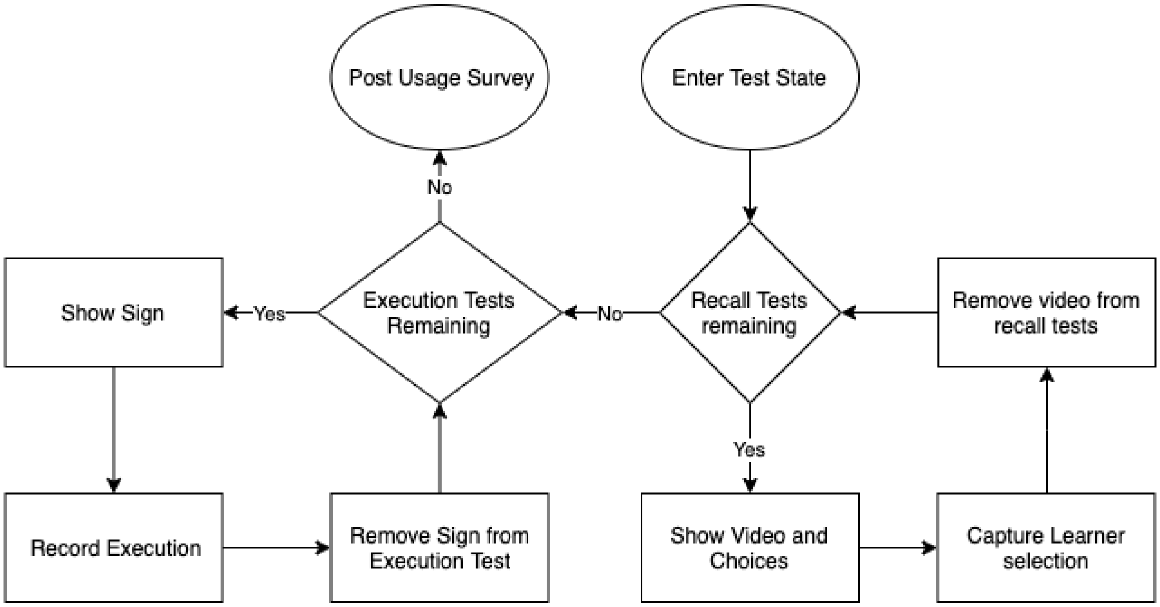


Figure 5.8: Flowchart for Retention and Execution Tests.

5.9 Results and Evaluation

The application built was evaluated first using a hold-out subset of the data collected for training. These evaluations help determine the proper algorithms to utilize and to determine appropriate thresholds for them. The final utility of the system will also depend upon design and usability factors which were also evaluated through user testing.

5.9.1 System Evaluation

An ideal system should give feedback to a learner only if their execution is incorrect. Giving unnecessary feedback for correct executions will hinder the learning process and decrease the usability. Conversely, providing sound and timely explanations for incorrect executions helps to improve utility and user trust. Smart systems such as L2S that use explainable machine learning tend to have a trade-off between explainability and performance which should be minimized.

The overall performance of the system was tested for 10 test users for a total of 750 signs. The training of the CNN for hand-shape feature extraction and optimal threshold determination was done using the remaining users. For each sign, 30 executions from the test dataset were taken as true class while 30 randomly selected executions from the pool of remaining signs were taken as incorrect class to avoid class imbalance. A pre-trained model from C3D Tran *et al.* (2015) was retrained with the data we collected and was used as the baseline for comparison. This model has an accuracy of 82.3 % on UCF101 Soomro *et al.* (2012) and 87.7 % in YUPENN-Scene Derpanis *et al.* (2012) datasets. The final recognition accuracy of C3D on L2S dataset using the same train-test split was 45.38 %. Our approach achieves a higher accuracy of 87.9 % while still offering explanations about its decisions in the form of learner feedback.

Determination of Appropriate Thresholds

To obtain the results, data collected from one learner was selected at random and served as the *tutorial* dataset. Then each sign for each user in the test dataset was compared against the corresponding tutorial sign. The location module had an overall recall of 96.4 % and a precision of 24.3 %. The lower precision is because many signs in the test dataset had similar locations. We performed a test comparing only the signs 'LARGE' to the sign 'FATHER' and both the precision and recall were 100 %. The movement module had an overall recall of 93.2 % and a precision of 52.4 %. The hand-shape module had a recall of 89 % and a precision of 74 %. The overall model is constructed as a waterfall combination of all three models such that the movement model is executed only when the location was found to be correct, and the hand-shape model is executed only when both the location and movement were correct. The overall precision, recall, f-1 score, and accuracies is summarized in Table 4.5.

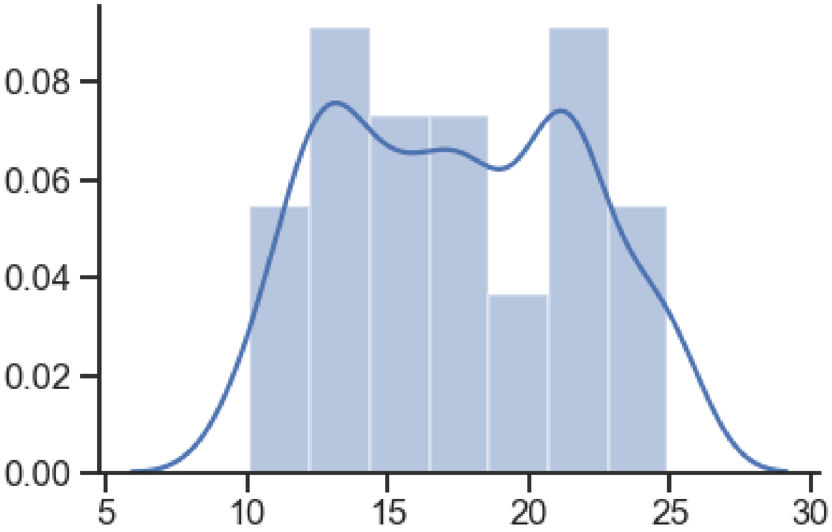


Figure 5.9: Learners Spent between 12 and 25 Minutes Interacting with the Application.

Since some signs were easier to learn than others, the distribution for the mistakes across the signs varied as seen in Figure 5.12. This type of information would be very helpful for instructors as they would be able to focus their attention on signs that students have more of a difficulty mastering.

5.9.2 Usage Evaluation

Retention Tests

A total of 156 retention tests were performed for 14 of the ASL signs by the 26 learners. Each learner was given 6 randomly selected sings for retention tests. For each test, the learner was given a video and had to choose among 4 options for the correct one. Thus, the baseline performance for random guessing would be 25%. The average performance across all learners on retention tests regardless of if feedback was received was 87%. The performance on retention tests improved to 90.79% with

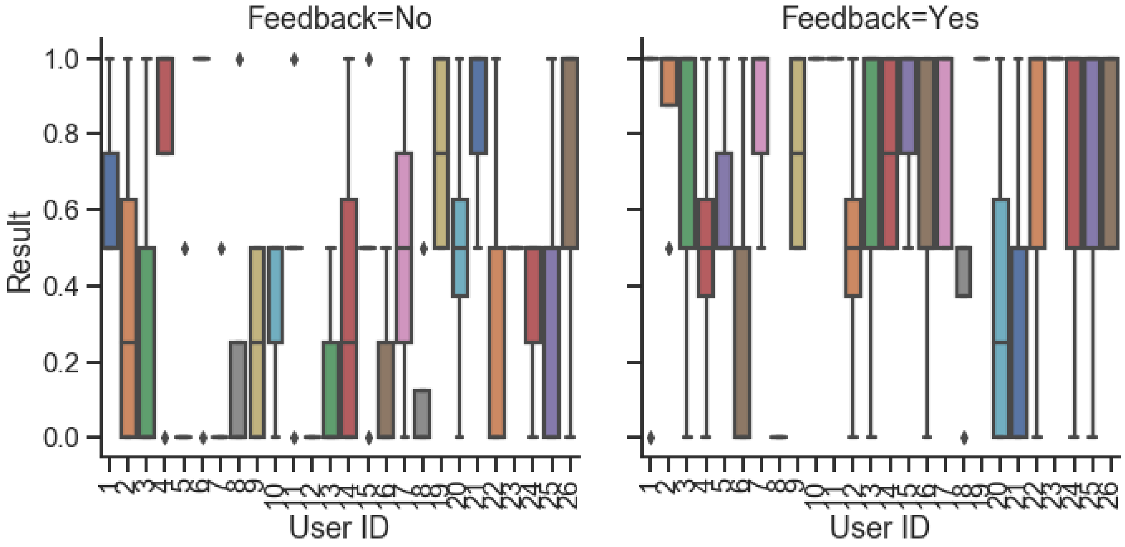


Figure 5.10: Execution Test Results for Users is Notably Better for Signs They Received Feedback on.

feedback while the performance for signs without practice and feedback was 83.67%. Overall we found that practice and feedback did not have significant improvement in performance for retention tests given in the form of multiple-choice questions. Figure 5.16 shows the relative improvement that practice and feedback had on the level of individual words. Figure 5.11 shows the improvement across the users. Although some of the words showed improvement with practice and feedback, a lot of the signs already had a 100 % retention performance. When performing a one-tail t-test for testing the difference between the retention results when feedback was given vs. that when feedback was not given, the p-value obtained was 0.095. Thus we can reject the null hypothesis that the distributions have the same mean only at a 90 % confidence level.

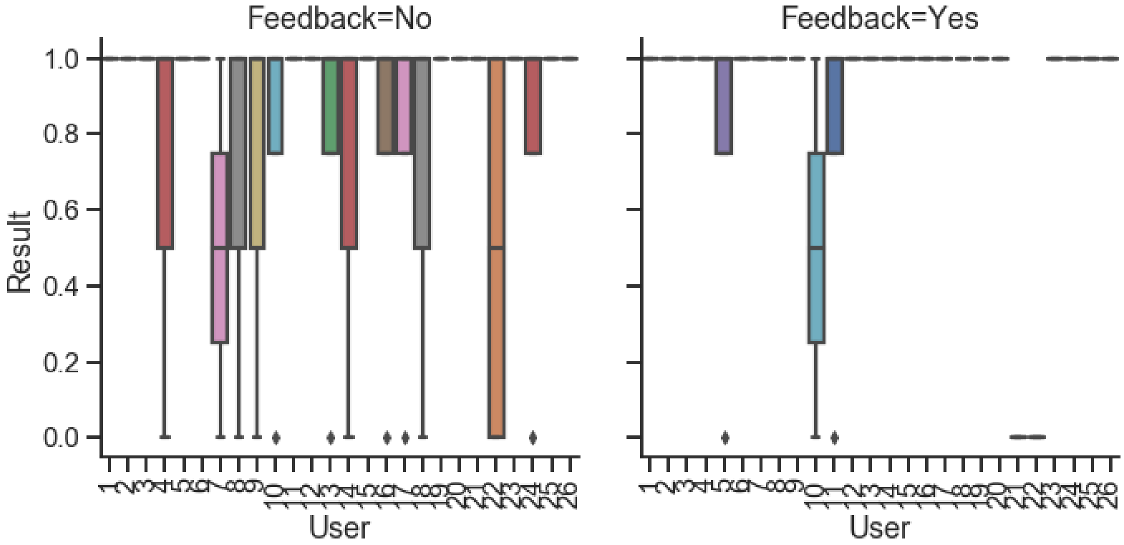


Figure 5.11: Retention Test Results for Users is Slightly Better for Signs They Received Feedback on.

Execution Tests

A total of 208 execution tests were performed for 8 of the 14 of the ASL signs. Each of the 26 learners was given 8 randomly selected sings to execute. After all retention tests were complete, the learner was given a sign and asked to begin recording its execution. Learners found execution tests significantly harder than retention tests and the average score for execution tests was 54.32 %. The execution tests were scored offline by the research team. If the learner had any two of location, movement, or handshape correct on both hands, then she received a score of 0.5 for that sign. If all three were correct, she received 1. Otherwise, she received 0. It was found that practicing and receiving feedback helped learners significantly to get better performance. Figure 5.14 shows a general improvement in results with increased practice. It also shows that there is higher uncertainty in the reported numbers for signs that were practiced 3 or more times because not many learners practiced signs more than twice as seen in Figure 5.15. Figure 5.13 shows the distribution shift for execution tests using a violin chart. The median value without practice was 0.5 while the median value with practice was 1. The effect of the number of practices was that the execution performance increased up to two practices, but then decreased slightly, but remained higher than the average without any practices. This can be seen in Figure 5.15. However, it should be noted that less than 10 % of the signs were practiced 3 or more times. This is reflected by the wide confidence intervals for 3 and 4 repetitions. Figure 5.17 shows the relative improvement that practice and feedback had on the level of individual words. A general improvement in performance is seen across the board for users as seen in Figure 5.10. When performing a one-tail t-test for testing the difference between the execution results when feedback was given vs. those when feedback was not given, the p-value obtained was 4.6e-7. Thus we can reject the null hypothesis that the mean for the two distributions is the same with at a very high confidence level.

5.9.3 Post Usage Survey

As part of the study, all of the participants were required to complete a post-usage survey to evaluate the efficacy of the application. The results of this survey are summarized in Table 5.2. Apart from the questions covered in the table, we received a lot of positive feedback on how the interactivity helps them to stay motivated and engaged while using the application. We also received a lot of suggestions for improvements, some of which were User Interface related. Some of the users also expressed concerns about the fact that the front-facing camera showed a flipped version of the signs and that was confusing when they saw the tutor‘s execution side-by-side with their own. However, most users were able to properly understand the effect of mirroring and using the correct hand. For this study only right-hand dominant tutors

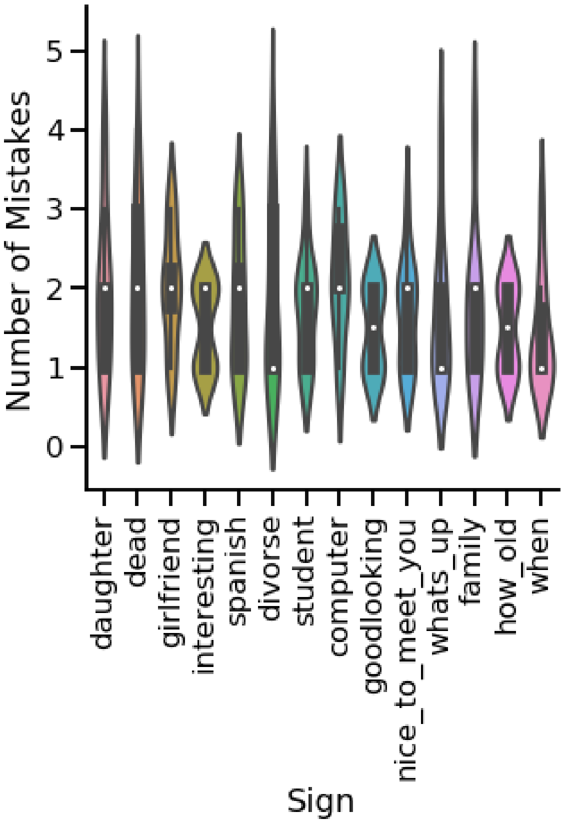


Figure 5.12: Distribution of mistakes across the different signs.

and users were present.

5.10 Discussion and Future Work

Due to the nature of random selections, some signs were easier to recall and execute than others. Some of this can be explained due to the pantomime nature of signs. This means that in American Sign Language some signs tend to mimic or somehow represent their semantic meanings. This makes some of the signs easy to retain and guess. This is especially true in the context of a multiple-choice type question when a sign such as ’daughter’ which makes a visual reference to the act of carrying a child. While for other signs such as ‘when’ has a reference to a clock to indicate time,

Question Category	**Very Low**	**Low**	**Medium**	**High**	**Very High**
Importance of Feedback	0%	3.8%	0%	30.8%	65.4%
Helpfulness of ASL tutor application	0%	0%	15.4%	26.9%	57.4%
Was the feedback useful for learning	0%	3.8%	3.8%	46.2%	46.2%
Helpfulness of location feedback	0%	0%	11.5%	38.5%	50%
Helpfulness of movement feedback	0%	7.7%	11.5%	34.6%	46.2%
Helpfulness of handshape feedback	0%	8%	20%	56%	16%
Preference to use this application over learning with videos	0%	0%	11.5%	26.9%	61.5%

Table 5.2: Results of Post-usage Survey from all 26 Participants. Participants Found Location Feedback Adequate, but Wanted More Information for Handshape Feedbacks.

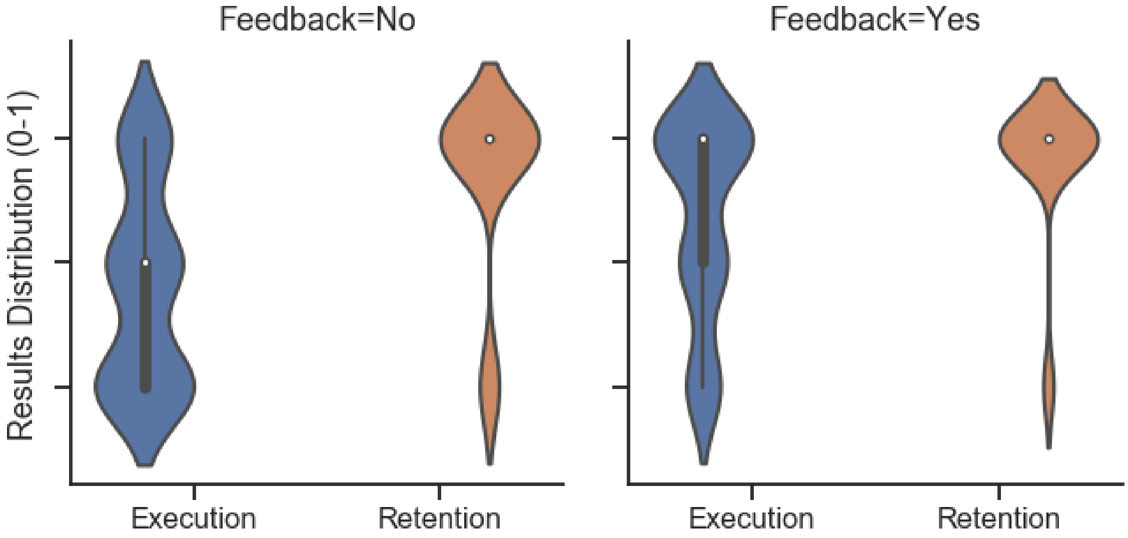


Figure 5.13: Distribution Changes for Retention and Execution Performances with Practice.

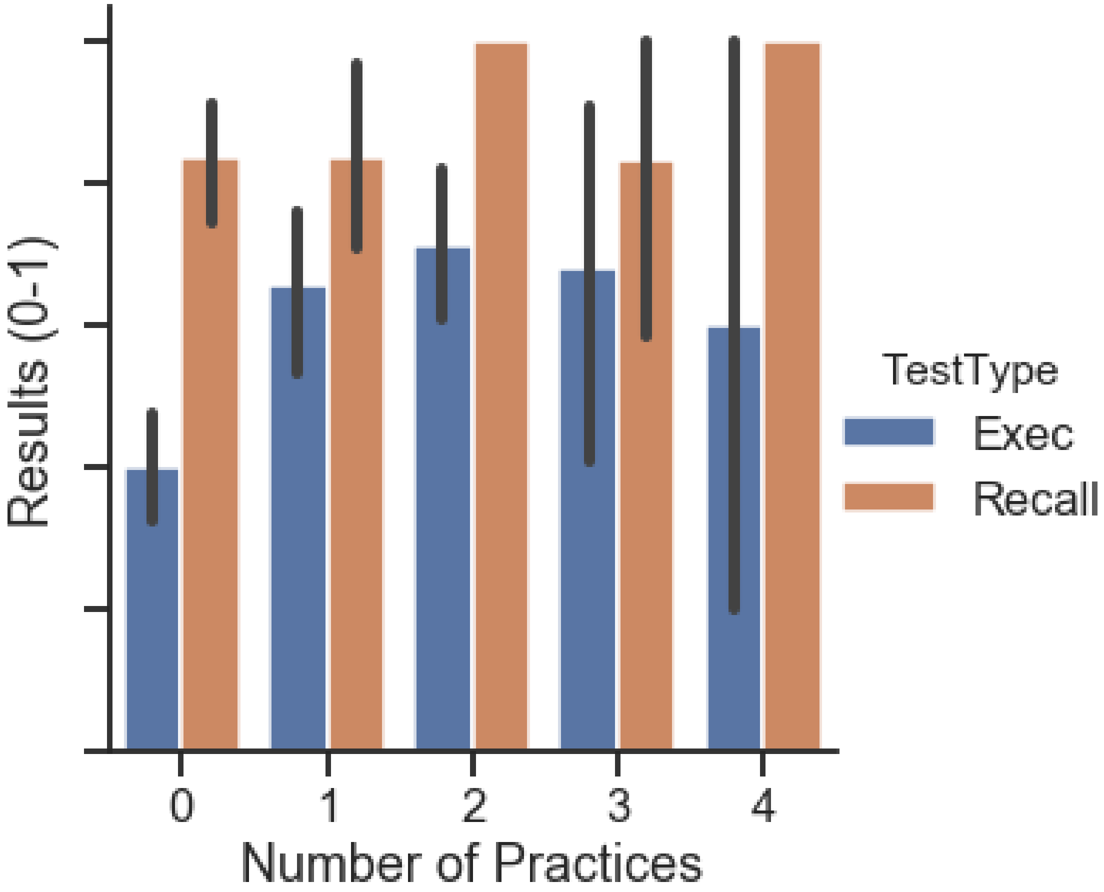


Figure 5.14: Effect of Number of Attempts. Uncertainty is Higher for Higher practice numbers

but the reference is much less obvious. Execution tests, on the other hand, were in general more difficult as they require a learner to memorize the movements, location, and handshape to be able to perform them. While learners found the execution tests harder in general, there was also a much clearer benefit to practicing and receiving feedback to execution performance when compared to retention performance. This could also be explained in part since the kinds of feedback provided by Learn2Sign were directly related to the execution of the signs. We hypothesize that if references to the pantomime of the signs or some other mnemonic tips are shared, the retention

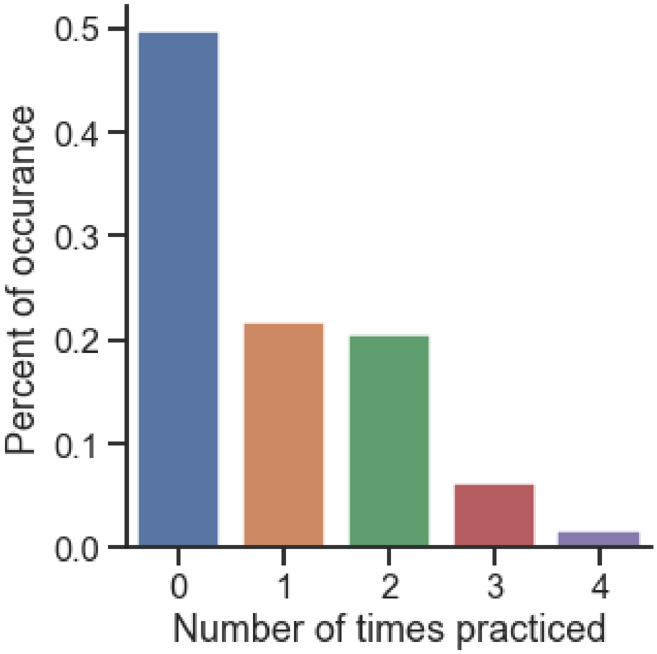


Figure 5.15: Only Half of the Test Signs were Practiced. Most learners Chose to Practice Once or Twice.

and perhaps the execution performance of the new learners will be further enhanced. This is left for future work. After the user studies, learners mentioned that although the movement and handshape feedback were very useful, some visual forms of feedback should be added perhaps in the form of the cropped hand that was the most incorrect or a gif showing the movement trajectories. We think that this will be an excellent addition to an application like Learn2Sign. Signs in ASL are not specified for the right and left hand but according to hand dominance. In addition to this, due to the mirroring effect of recorded videos, some students were confused about the correct hand to utilize. Although we did not directly tackle this issue in the experimental setup, learners were instructed to think of signs in terms of dominant and non-dominant hands. However, if a left-hand dominant student tried to replicate a right-hand dominant instructor and did take into account the effect of mirroring in

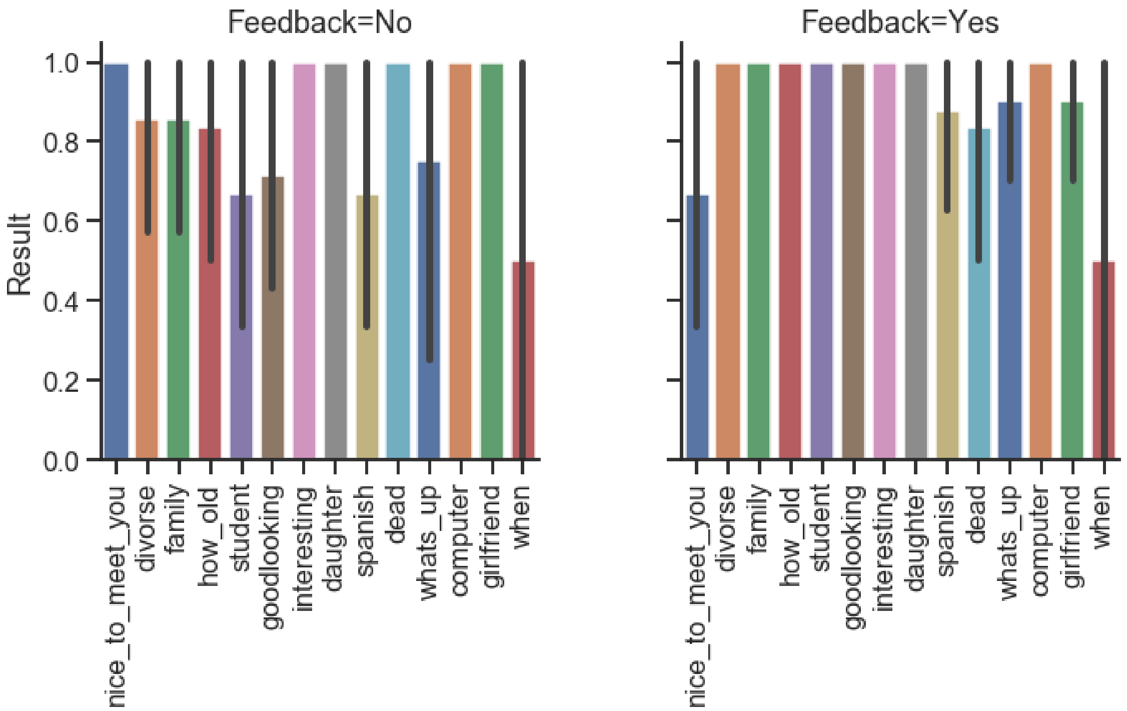


Figure 5.16: Practice and Feedback show Slight Improvement for Retention Test Results across All Signs.

the video, her execution would be deemed incorrect by L2S. In future studies, this should be handled by allowing the learned to specify their dominant hand before beginning the tests.

5.11 Conclusions

The increase in demand for learning sign languages will be accompanied by a corresponding demand for technology-assisted learning techniques. In addition, even instructor-led classroom sessions will benefit from automated practice and feedback solutions that increase engagement enhance learning outcomes. There is currently a vacuum in this space due to the complexities of sign language recognition and the lack of sufficient research for feedback mechanisms. In this work, I presented

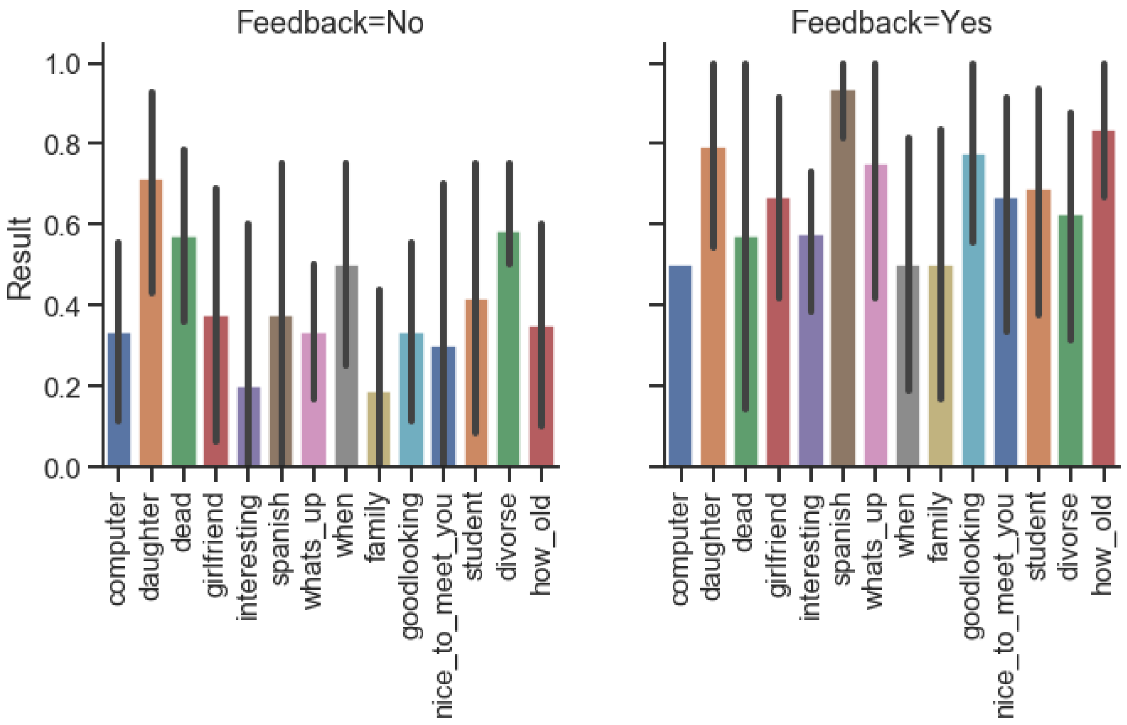


Figure 5.17: Practice and Feedback show Notable Improvement for Execution Test Results for Across All Signs.

Learn2Sign, which is a chatbot interface for teaching arbitrary sign language signs to new learners with a special focus on providing conceptual feedback. Leveraging the latest techniques of signal processing, computer vision, and explainable AI, I showed that Learn2Sign can produce improvements towards both the execution and retention of sign language signs. We showed that the execution performance of 26 new learners for 14 different ASL increased from an average of 0.4 to 0.68. I also described the technical details of the various modules involved and how the design of the system is easily scalable. The code for the Learn2Sign web application will be open-sourced along with this document.

Chapter 6

CONCLUSIONS

6.1 Conclusion

There is an increasing need and demand for learning sign language. Feedback is very important for language learning and intelligent language learning applications must provide effective and meaningful feedback. There have also been significant advances in research for recognizing sign languages, however technological solutions that leverage them to provide intelligent learning environments do not exist. In this work, we identify different types of potential feedback we can provide to learners of sign language and address some challenges in doing so. We propose a pipeline of three non-parametric recognition modules and an incremental feedback mechanism to facilitate learning. We tested our system on real-world data from a variety of devices and settings to achieve a final recognition accuracy of 87.9 %. This demonstrates that using explainable machine learning for gesture learning is desirable and effective. We also provided different types of feedback mechanisms based on the results of a user survey and best practices in implementing them. Finally, we collected data from 100 users of L2S with 3 repetitions for each of the 25 signs for a total of 7500 instances Lab (2018).

The increase in demand for learning sign languages will be accompanied by a corresponding demand for technology-assisted learning techniques. In addition, even instructor-led classroom sessions will benefit from automated practice and feedback solutions that increase engagement enhance learning outcomes. There is currently a vacuum in this space due to the complexities of sign language recognition and the lack of sufficient research for feedback mechanisms. In this work, we presented L2S, which is a chatbot interface for teaching arbitrary sign language signs to new learners with a special focus on providing conceptual feedback. Leveraging the latest techniques of signal processing, computer vision, and explainable AI, we showed that L2S can produce improvements towards both the execution and retention of sign language signs. We showed that the execution performance of 26 new learners for 14 different ASL increased from an average of 0.4 to 0.68. We also described the technical details of the various modules involved and how the design of the system is easily scalable. The code for the L2S web application will be open-sourced along with this paper. The web application will be preloaded with the curriculum we used and be publicly hosted.

6.2 Discussion and Future Work

Due to the nature of random selections, some signs were easier to recall and execute than others. Some of this can be explained due to the pantomime nature of signs. This means that in American Sign Language some signs tend to mimic or somehow represent their semantic meanings. This makes some of the signs easy to retain and guess. This is especially true in the context of a multiple-choice type question when a sign such as 'daughter' which makes a visual reference to the act of carrying a child. While for other signs such as ‘when’ has a reference to a clock to indicate time, but the reference is much less obvious. Execution tests, on the other hand, were in general more difficult as they require a learner to memorize the movements, location, and handshape to be able to perform them. While learners found the execution tests harder in general, there was also a much clearer benefit to practicing and receiving feedback to execution performance when compared to retention performance. This could also be explained in part since the kinds of feedback provided by L2S were directly related to the execution of the signs. We hypothesize that if references to the pantomime of the signs or some other mnemonic tips are shared, the retention and perhaps the execution performance of the new learners will be further enhanced. This is left for future work. After the user studies, learners mentioned that although the movement and handshape feedback were very useful, some visual forms of feedback should be added perhaps in the form of the cropped hand that was the most incorrect or a gif showing the movement trajectories. We think that this will be an excellent addition to an application like L2S.

We demonstrated the need for a feedback-based technological solution for sign language learning and provided an implementation with a modular feedback mechanism. The user preference for the desired amount of feedback can be changed by altering the value for 'Feedback Sensitivity'. The trade-off between 'Feedback Sensitivity' and the amount of feedback received as well as other performance metrics is summarized in Figure 4.9. Although we designed our feedback mechanism based on principles from linguistics and user survey, only a large scale usage of such an application will provide definitive best practices for the most effective feedback. In such future studies, issues such as the extent of user control for determining types of feedback and the possibility of peer-to-peer feedback for on-line learning has to be evaluated as suggested by works such as Fiebrink *et al.* (2011). This work provides the foundations and feasibility for interactive and intelligent sign language learning to pave the path for such future work.

We collect usage and interaction data from 100 new learners as part of this work, which will be foundational to assist future researchers. Although the focus of this work was on the manual portion of sign languages, the preprocessing includes location estimates for the eyes, ears, and the nose. This can be utilized for including facial expression recognition and feedback in future works. We evaluated only 25 isolated words for ASL, but in the future, this work can be extended to more words and phrases and to include other sign languages since the general principles will remain the same. In this work, we used sign language as a test application, however, the insights from this work can be easily applied to other gesture domains such as combat sign training for military or industrial operator signs.

REFERENCES

Aaron J. Newman, D. B., "A critical period for right hemisphere recruitment in american sign language processing", in "Multimodal Signals: Cognitive and Algorithmic Issues", pp. 76–80 (2002).

Abreu, J. G., J. M. Teixeira, L. S. Figueiredo and V. Teichrieb, "Evaluating sign language recognition using the myo armband", in "2016 XVIII Symposium on Virtual and Augmented Reality (SVR)", pp. 64–70 (IEEE, 2016).

Adadi, A. and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (xai)", IEEE Access **6**, 52138–52160 (2018).

Alpaydin, E., *Introduction to machine learning* (MIT press, 2014).

Anguera, X., R. Macrae and N. Oliver, "Partial sequence matching using an unbounded dynamic time warping algorithm", in "Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on", pp. 3582–3585 (IEEE, 2010).

Association, M. L., "Language Enrollment Database", https://apps.mla.org/flsurvey_search, [Online; accessed 24-September-2018] (2016).

Bahan, B., *Non-manual Realization of Agreement in American Sign Language*, UMI Dissertation Services (UMI, 1996), URL https://books.google.com/books?id=FN8BNAEACAAJ.

Baker, S., "The importance of fingerspelling for reading", Visual Language and Visual Learning Science of Learning Center.(Research Brief No. 1). Washington, DC (2010).

Banerjee, A. and S. K. Gupta, "Analysis of smart mobile applications for healthcare under dynamic context changes", Mobile Computing, IEEE Transactions on **14**, 5, 904–919 (2015).

Banerjee, A., I. Lamrani, P. Paudyal and S. Gupta, "Generation of movement explanations for testing gesture based co-operative learning applications", in "2019 IEEE International Conference On Artificial Intelligence Testing (AITest)", pp. 9–16 (IEEE, 2019).

Battison, R., "Lexical borrowing in american sign language.", (1978).

Berndt, D. J. and J. Clifford, "Using dynamic time warping to find patterns in time series.", in "KDD workshop", vol. 10, pp. 359–370 (Seattle, WA, 1994).

Bilal, S., R. Akmeliawati, A. A. Shafie and M. J. E. Salami, "Hidden markov model for human to computer interaction: a study on human hand gesture recognition", Artificial Intelligence Review **40**, 4, 495–516 (2013).

Blum, A. L. and P. Langley, “Selection of relevant features and examples in machine learning”, Artificial intelligence **97**, 1, 245–271 (1997).

Bogo, F., A. Kanazawa, C. Lassner, P. Gehler, J. Romero and M. J. Black, “Keep it smpl: Automatic estimation of 3d human pose and shape from a single image”, in “European Conference on Computer Vision”, pp. 561–578 (Springer, 2016).

Bossard, B., A. Braffort and M. Jardino, “Some issues in sign language processing”, in “International Gesture Workshop”, pp. 90–100 (Springer, 2003).

Bradley, A. P., “The use of the area under the roc curve in the evaluation of machine learning algorithms”, Pattern recognition **30**, 7, 1145–1159 (1997).

Brennan, M., “Making borrowing work in british sign language”, Foreign vocabulary in sign languages: A cross-linguistic investigation of word formation pp. 49–85 (2001).

Brentari, D. and C. Padden, “Native and foreign vocabulary in american sign language: A lexicon with multiple origins”, Foreign vocabulary in sign languages pp. 87–120 (2001).

Chen, C.-H. and D. Ramanan, “3d human pose estimation= 2d pose estimation+ matching”, in “CVPR”, vol. 2, p. 6 (2017).

Chen, F.-S., C.-M. Fu and C.-L. Huang, “Hand gesture recognition using a real-time tracking method and hidden markov models”, Image and vision computing **21**, 8, 745–758 (2003).

Chen, Q., G. Hu, F. Gu and P. Xiang, “Learning optimal warping window size of dtw for time series classification”, in “2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)”, pp. 1272–1277 (IEEE, 2012).

Chen, X. and A. L. Yuille, “Articulated pose estimation by a graphical model with image dependent pairwise relations”, in “Advances in neural information processing systems”, pp. 1736–1744 (2014).

Chen, X., X. Zhang, Z.-Y. Zhao, J.-H. Yang, V. Lantz and K.-Q. Wang, “Hand gesture recognition research based on surface emg sensors and 2d-accelerometers”, in “Wearable Computers, 2007 11th IEEE International Symposium on”, pp. 11–14 (IEEE, 2007).

Chen, Y., H. Jiang, C. Li, X. Jia and P. Ghamisi, “Deep feature extraction and classification of hyperspectral images based on convolutional neural networks”, IEEE Transactions on Geoscience and Remote Sensing **54**, 10, 6232–6251 (2016).

Cihan Camgoz, N., S. Hadfield, O. Koller, H. Ney and R. Bowden, “Neural sign language translation”, in “Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition”, pp. 7784–7793 (2018).

Compton, A. E., “American sign language as a foreign language equivalent at james madison university”, (2016).

Corballis, M. C. and M. C. Corballis, *From hand to mouth: The origins of language* (Princeton University Press, 2002).

Corradini, A., “Dynamic time warping for off-line recognition of a small gesture vocabulary”, in “Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 2001. Proceedings. IEEE ICCV Workshop on”, pp. 82–89 (IEEE, 2001).

Costello, E., *American sign language dictionary* (Random House Reference &, 2008a).

Costello, E., *Random House Webster’s American Sign Language Dictionary* (Random House Reference, 2008b), URL `https://books.google.com/books?id=nqehzaWHTZIC`.

Darken, “Github MyoLib by darken”, `https://github.com/d4rken/myolib`, accessed: 2016-05-04 (2015).

Derpanis, K. G., M. Lecce, K. Daniilidis and R. P. Wildes, “Dynamic scene understanding: The role of orientation features in space and time in scene classification”, in “Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on”, pp. 1306–1313 (IEEE, 2012).

Dipietro, L., A. M. Sabatini and P. Dario, “A survey of glove-based systems and their applications”, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **38**, 4, 461–482 (2008).

Doran, D., S. Schulz and T. R. Besold, “What does explainable ai really mean? a new conceptualization of perspectives”, arXiv preprint arXiv:1710.00794 (2017).

Ehsani, F. and E. Knodt, “Speech technology in computer-aided language learning: Strengths and limitations of a new call paradigm”, (1998).

El Hayek, H., J. Nacouzi, A. Kassem, M. Hamad and S. El-Murr, “Sign to letter translator system using a hand glove”, in “e-Technologies and Networks for Development (ICeND), 2014 Third International Conference on”, pp. 146–150 (IEEE, 2014).

Emmorey, K., *Language, cognition, and the brain: Insights from sign language research* (Psychology Press, 2001).

Emmorey, K., R. Bosworth and T. Kraljic, “Visual feedback and self-monitoring of sign language”, Journal of Memory and Language **61**, 3, 398–411 (2009).

Fang, G., W. Gao and D. Zhao, “Large vocabulary sign language recognition based on fuzzy decision trees”, Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on **34**, 3, 305–314 (2004).

Feichtenhofer, C., A. Pinz and A. Zisserman, “Convolutional two-stream network fusion for video action recognition”, in “Proceedings of the IEEE conference on computer vision and pattern recognition”, pp. 1933–1941 (2016).

Fels, S. S. and G. E. Hinton, "Glove-talk: A neural network interface between a dataglove and a speech synthesizer", Neural Networks, IEEE Transactions on **4**, 1, 2–8 (1993).

Fels, S. S. and G. E. Hinton, "Glove-talkii-a neural-network interface which maps gestures to parallel formant speech synthesizer controls", IEEE transactions on neural networks **9**, 1, 205–212 (1998).

Fiebrink, R., P. R. Cook and D. Trueman, "Human model evaluation in interactive supervised learning", in "Proceedings of the SIGCHI Conference on Human Factors in Computing Systems", pp. 147–156 (ACM, 2011).

Frati, V. and D. Prattichizzo, "Using kinect for hand tracking and rendering in wearable haptics", in "2011 IEEE World Haptics Conference", pp. 317–321 (IEEE, 2011).

Freund, Y., R. Schapire and N. Abe, "A short introduction to boosting", Journal-Japanese Society For Artificial Intelligence **14**, 771-780, 1612 (1999).

Fukkink, R. G., N. Trienekens and L. J. Kramer, "Video feedback in education and training: Putting learning in the picture", Educational Psychology Review **23**, 1, 45–63 (2011).

Grobel, K. and M. Assan, "Isolated sign language recognition using hidden markov models", in "Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on", vol. 1, pp. 162–167 (IEEE, 1997).

Gupta, S. K., T. Mukherjee and K. K. Venkatasubramanian, *Body area networks: Safety, security, and sustainability* (Cambridge University Press, 2013).

Haug, T., "Review of sign language assessment instruments", Sign Language & Linguistics **8**, 1, 61–98 (2005).

Hermans, D., H. Knoors and L. Verhoeven, "Assessment of sign language development: The case of deaf children in the netherlands", Journal of Deaf Studies and Deaf Education **15**, 2, 107–119 (2009).

Ho, D., E. Liang, I. Stoica, P. Abbeel and X. Chen, "Population based augmentation: Efficient learning of augmentation policy schedules", arXiv preprint arXiv:1905.05393 (2019).

Huang, D.-Y., W.-C. Hu and S.-H. Chang, "Vision-based hand gesture recognition using pca+ gabor filters and svm", in "Intelligent Information Hiding and Multimedia Signal Processing, 2009. IIH-MSP'09. Fifth International Conference on", pp. 1–4 (IEEE, 2009).

Hudson, J. N., "Computer-aided learning in the real world of medical education: does the quality of interaction with the computer affect student learning?", Medical education **38**, 8, 887–895 (2004).

Huenerfauth, M., E. Gale, B. Penly, M. Willard and D. Hariharan, “Comparing methods of displaying language feedback for student videos of american sign language”, in “Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility”, pp. 139–146 (2015).

Joachims, T., “Text categorization with support vector machines: Learning with many relevant features”, Machine learning: ECML-98 pp. 137–142 (1998).

Johnston, T. and A. Schembri, *Australian Sign Language (Auslan): An introduction to sign language linguistics* (Cambridge University Press, 2007), URL `https://books.google.com/books?id=dnxDgCvEnJoC`.

Kamzin, A., P. Paudyal, A. Banerjee and S. K. Gupta, “Evaluating the gap between hype and performance of ai systems”, (????).

Karthikeyan, D. and M. G. Muthulakshmi, “English letters finger spelling sign language recognition system”, International Journal of Engineering Trends and Technology **107**, 334–339 (2014).

Kelly, D., J. McDonald and C. Markham, “A system for teaching sign language using live gesture feedback”, in “2008 8th IEEE International Conference on Automatic Face & Gesture Recognition”, pp. 1–2 (IEEE, 2008).

Kelly, D., J. Reilly Delannoy, J. Mc Donald and C. Markham, “A framework for continuous multimodal sign language recognition”, in “Proceedings of the 2009 international conference on Multimodal interfaces”, pp. 351–358 (ACM, 2009).

Khan, R. Z. and N. A. Ibraheem, “Survey on gesture recognition for hand image postures”, Computer and Information Science **5**, 3, 110 (2012).

Kim, D., O. Hilliges, S. Izadi, A. D. Butler, J. Chen, I. Oikonomidis and P. Olivier, “Digits: freehand 3d interactions anywhere using a wrist-worn gloveless sensor”, in “Proceedings of the 25th annual ACM symposium on User interface software and technology”, pp. 167–176 (ACM, 2012).

Kim, J., J. Wagner, M. Rehm and E. André, “Bi-channel sensor fusion for automatic sign language recognition”, in “Automatic Face Gesture Recognition, 2008. FG’08. 8th IEEE International Conference on”, pp. 1–6 (IEEE, 2008).

Kim, K.-W., M.-S. Lee, B.-R. Soon, M.-H. Ryu and J.-N. Kim, “Recognition of sign language with an inertial sensor-based data glove”, Technology and Health Care **24**, s1, S223–S230 (2015).

Koller, O., J. Forster and H. Ney, “Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers”, Computer Vision and Image Understanding **141**, 108–125 (2015).

Kornblith, S., J. Shlens and Q. V. Le, “Do better imagenet models transfer better?”, in “Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition”, pp. 2661–2671 (2019).

Kortelainen, J. M., M. Van Gils and J. Parkka, “Multichannel bed pressure sensor for sleep monitoring”, in “Computing in Cardiology (CinC), 2012”, pp. 313–316 (IEEE, 2012).

Kumar, P., H. Gauba, P. P. Roy and D. P. Dogra, “Coupled hmm-based multi-sensor data fusion for sign language recognition”, Pattern Recognition Letters **86**, 1–8 (2017).

Kuroda, T., Y. Tabata, A. Goto, H. Ikuta, M. Murakami *et al.*, “Consumer price data-glove for sign language recognition”, in “Proc. of 5th Intl Conf. Disability, Virtual Reality Assoc. Tech., Oxford, UK”, pp. 253–258 (2004).

Kuznetsova, A., L. Leal-Taixé and B. Rosenhahn, “Real-time sign language recognition using a consumer depth camera”, in “Proceedings of the IEEE International Conference on Computer Vision Workshops”, pp. 83–90 (2013).

Lab, I., “Learn2sign details page”, URL `https://impact.asu.edu` (2018).

Lee, J., A. Banerjee and S. K. Gupta, “Mt-diet: Automated smartphone based diet assessment with infrared images”, in “2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)”, pp. 1–6 (IEEE, 2016).

Lee, J., A. Banerjee, P. Paudyal and S. K. Gupta, “Mt-diet: Automated diet assessment using myo and thermal”, in “Late-Breaking Research Abstract at the conference on Wireless Health”, p. 20 (????).

Lee, J., P. Paudyal, A. Banerjee and S. K. Gupta, “Fit-eve&adam: Estimation of velocity & energy for automated diet activity monitoring”, in “2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)”, pp. 1071–1074 (IEEE, 2017).

Lee, J., P. Paudyal, A. Banerjee and S. K. Gupta, “Idea: Instant detection of eating action using wrist-worn sensors in absence of user-specific model”, in “Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization”, pp. 371–372 (2018).

Lee, J., P. Paudyal, A. Banerjee and S. K. Gupta, “A user-adaptive modeling for eating action identification from wristband time series”, ACM Transactions on Interactive Intelligent Systems (TiiS) **9**, 4, 1–35 (2019).

Lewis, D. D., “Naive (bayes) at forty: The independence assumption in information retrieval”, in “European conference on machine learning”, pp. 4–15 (Springer, 1998).

Li, Y., X. Chen, J. Tian, X. Zhang, K. Wang and J. Yang, “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”, ICMI-MLMI ’10, pp. 17:1–17:7 (ACM, New York, NY, USA, 2010), URL `http://doi.acm.org/10.1145/1891903.1891926`.

Liang, R.-H. and M. Ouhyoung, “A real-time continuous gesture recognition system for sign language”, in “Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on”, pp. 558–567 (IEEE, 1998).

Liaw, A. and M. Wiener, “Classification and regression by randomforest”, R news **2**, 3, 18–22 (2002).

Lieberman, P., “Motor control, speech, and the evolution of human language”, Studies in the Evolution of Language **3**, 255–271 (2003).

Lifeprint, “Finger Spelling for asl”, `http://www.lifeprint.com/asl101`, accessed: 2016-04-15 (2016).

Lightbown, P. M. and N. Spada, “Focus-on-form and corrective feedback in communicative language teaching: Effects on second language learning”, Studies in second language acquisition **12**, 4, 429–448 (1990).

Lim, B., A. Sarkar, A. Smith-Renner and S. Stumpf, “Exss: explainable smart systems 2019”, in “Proceedings of the 24th International Conference on Intelligent User Interfaces: Companion”, pp. 125–126 (2019).

Lim, K. M., A. W. Tan and S. C. Tan, “A feature covariance matrix with serial particle filter for isolated sign language recognition”, Expert Systems with Applications **54**, 208–218 (2016).

Lu, Z., X. Chen, Q. Li, X. Zhang and P. Zhou, “A hand gesture recognition framework and wearable gesture-based interaction prototype for mobile devices”, Human-Machine Systems, IEEE Transactions on **44**, 2, 293–299 (2014).

Mackey, A., “Feedback, noticing and instructed second language learning”, Applied linguistics **27**, 3, 405–430 (2006).

Magnan, S. S. and M. Back, “Social interaction and linguistic gain during study abroad”, Foreign Language Annals **40**, 1, 43–61 (2007).

Maller, S., J. Singleton, S. Supalla and T. Wix, “The development and psychometric properties of the american sign language proficiency assessment (asl-pa).”, Journal of Deaf Studies and Deaf Education **4**, 4, 249–269 (1999).

Marschark, M. and P. E. Spencer, *The Oxford handbook of deaf studies, language, and education*, vol. 2 (Oxford University Press, 2010).

Mayberry, R. I., “When timing is everything: Age of first-language acquisition effects on second-language learning”, Applied Psycholinguistics **28**, 3, 537–549 (2007).

Miller, R. B., “Response time in man-computer conversational transactions”, in “Proceedings of the December 9-11, 1968, fall joint computer conference, part I”, pp. 267–277 (ACM, 1968).

Mitra, S. and T. Acharya, “Gesture recognition: A survey”, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on **37**, 3, 311–324 (2007).

Morency, L.-P., A. Quattoni and T. Darrell, “Latent-dynamic discriminative models for continuous gesture recognition”, in “Computer Vision and Pattern Recognition, 2007. CVPR’07. IEEE Conference on”, pp. 1–8 (IEEE, 2007).

Müller, M., “Dynamic time warping”, Information retrieval for music and motion pp. 69–84 (2007).

Nadil, M., F. Souami, A. Labed and H. Sahbi, “Kcca-based technique for profile face identification”, EURASIP Journal on Image and Video Processing **2017**, 1, 2 (2016).

Niederberger, N., “Does the knowledge of a natural sign language facilitate deaf childrens learning to read and write”, Sign bilingualism: Language development, interaction, and maintenance in sign language contact situations pp. 29–50 (2008).

Oikonomidis, I., N. Kyriazis and A. A. Argyros, “Efficient model-based 3d tracking of hand articulations using kinect.”, in “BmVC”, vol. 1, p. 3 (2011).

Organization, W. H., “Deafness and hearing loss”, `http://www.who.int/news-room/fact-sheets/`, [Online; accessed 24-September-2018] (2018).

Oskooyee, K. S., A. Banerjee and S. K. Gupta, “Neuro movie theatre: A real-time internet-of-people based mobile application”, (2015).

Oz, C. and M. C. Leu, “American sign language word recognition with a sensory glove using artificial neural networks”, Engineering Applications of Artificial Intelligence **24**, 7, 1204–1213 (2011).

Padden, C., “Learning to fingerspell twice: Young signing children?s acquisition of fingerspelling”, Advances in the sign language development of deaf children pp. 189–201 (2006).

Padden, C. A. and D. M. Perlmutter, “American sign language and the architecture of phonological theory”, Natural Language & Linguistic Theory **5**, 3, 335–375 (1987).

Papandreou, G., T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler and K. Murphy, “Towards accurate multi-person pose estimation in the wild”, in “CVPR”, vol. 3, p. 6 (2017).

Paudyal, P., A. Banerjee and S. Gupta, “Gesture recognition and communication”, US Patent App. 15/952,784 (2018).

Paudyal, P., A. Banerjee and S. Gupta, “On evaluating the effects of feedback for sign language learning using explainable ai”, in “Proceedings of the 25th International Conference on Intelligent User Interfaces Companion”, pp. 83–84 (2020).

Paudyal, P., A. Banerjee and S. K. Gupta, “Sceptre: a pervasive, non-invasive, and programmable gesture recognition technology”, in “Proceedings of the 21st International Conference on Intelligent User Interfaces”, pp. 282–293 (ACM, 2016).

Paudyal, P., A. Banerjee, Y. Hu and S. Gupta, “Davee: A deaf accessible virtual environment for education”, in “Proceedings of the 2019 on Creativity and Cognition”, pp. 522–526 (2019a).

Paudyal, P., J. Lee, A. Banerjee and S. K. Gupta, “Dyfav: Dynamic feature selection and voting for real-time recognition of fingerspelled alphabet using wearables”, in “Proceedings of the 22nd International Conference on Intelligent User Interfaces”, pp. 457–467 (ACM, 2017).

Paudyal, P., J. Lee, A. Banerjee and S. K. Gupta, “A comparison of techniques for sign language alphabet recognition using armband wearables”, ACM Transactions on Interactive Intelligent Systems (TiiS) **9**, 2-3, 1–26 (2019b).

Paudyal, P., J. Lee, A. Kamzin, M. Soudki, A. Banerjee and S. K. Gupta, “Learn2sign: Explainable ai for sign language learning.”, in “IUI Workshops”, (2019c).

Paulino da Silva, J., M. V. Lamar and J. L. Bordim, “Accuracy and efficiency performance of the icp procedure applied to sign language recognition”, CLEI Electronic Journal **17**, 2, 11–11 (2014).

Pedersoli, F., S. Benini, N. Adami and R. Leonardi, “Xkin: an open source framework for hand pose and gesture recognition using kinect”, The Visual Computer **30**, 10, 1107–1122 (2014).

Pennington, M. C. and P. Rogerson-Revell, “Using technology for pronunciation teaching, learning, and assessment”, in “English Pronunciation Teaching and Research”, pp. 235–286 (Springer, 2019).

Pieters, W., “Explanation and trust: what to tell the user in security and ai?”, Ethics and information technology **13**, 1, 53–64 (2011).

Pore, M., K. Sadeghi, V. Chakati, A. Banerjee and S. K. Gupta, “Enabling real-time collaborative brain-mobile interactive applications on volunteer mobile devices”, in “Proceedings of the 2nd International Workshop on Hot Topics in Wireless”, pp. 46–50 (ACM, 2015).

Praveen, N., N. Karanth and M. Megha, “Sign language interpreter using a smart glove”, in “Advances in Electronics, Computers and Communications (ICAECC), 2014 International Conference on”, pp. 1–5 (IEEE, 2014).

Priyal, S. P. and P. K. Bora, “A robust static hand gesture recognition system using geometry based normalizations and krawtchouk moments”, Pattern Recognition **46**, 8, 2202–2219 (2013).

Pu, P. and L. Chen, “Trust-inspiring explanation interfaces for recommender systems”, Knowledge-Based Systems **20**, 6, 542–556 (2007).

Pugeault, N. and R. Bowden, “Spelling it out: Real-time asl fingerspelling recognition”, in “Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on”, pp. 1114–1119 (IEEE, 2011).

Quinto-Pozos, D., “Rates of fingerspelling in american sign language”, in “Poster presented at 10th Theoretical Issues in Sign Language Research conference, West Lafayette, Indiana”, vol. 30 (2010).

Rabiner, L. R., “A tutorial on hidden markov models and selected applications in speech recognition”, Proceedings of the IEEE **77**, 2, 257–286 (1989).

Ratanamahatana, C. A. and E. Keogh, “Everything you know about dynamic time warping is wrong”, in “Third workshop on mining temporal and sequential data”, vol. 32 (Citeseer, 2004).

Redmon, J., S. Divvala, R. Girshick and A. Farhadi, “You only look once: Unified, real-time object detection”, in “Proceedings of the IEEE conference on computer vision and pattern recognition”, pp. 779–788 (2016).

Robertson, S., C. Munteanu and G. Penn, “Designing pronunciation learning tools: The case for interactivity against over-engineering”, in “Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems”, p. 356 (ACM, 2018).

Rosen, R. S., “Beginning l2 production errors in asl lexical phonology: A cognitive phonology model”, Sign Language & Linguistics **7**, 1, 31–61 (2004).

Rosen, R. S., “American sign language curricula: A review”, Sign Language Studies **10**, 3, 348–381 (2010).

Saavy, S., “Signing Saavy: Your Sign Language Resouce”, `https://www.signingsavvy.com/`, [Online; accessed 28-September-2018] (2018).

Salvador, S. and P. Chan, “Toward accurate dynamic time warping in linear time and space”, Intelligent Data Analysis **11**, 5, 561–580 (2007).

Sandler, W., “The phonological organization of sign languages”, Language and linguistics compass **6**, 3, 162–182 (2012).

Sarafianos, N., B. Boteanu, B. Ionescu and I. A. Kakadiaris, “3d human pose estimation: A review of the literature and analysis of covariates”, Computer Vision and Image Understanding **152**, 1–20 (2016).

Savur, C., “American sign language recognition system by using surface emg signal”, (2015).

Savur, C. and F. Sahin, “Real-time american sign language recognition system using surface emg signal”, in “2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)”, pp. 497–502 (IEEE, 2015).

Schembri, A., G. Wigglesworth, T. Johnston, G. Leigh, R. Adam and R. Barker, “Issues in development of the test battery for australian sign language morphology and syntax”, Journal of Deaf Studies and Deaf Education **7**, 1, 18–40 (2002).

Sheen, Y., “Corrective feedback and learner uptake in communicative classrooms across instructional settings”, Language teaching research **8**, 3, 263–300 (2004).

Shield, A. and R. P. Meier, "Learning an embodied visual language: Four imitation strategies available to sign learners", Frontiers in psychology **9**, 811 (2018).

SignAll, "SignAll education system", https://www.signall.us/education/, accessed: 2019-12-30 (2020).

Singha, J. and K. Das, "Indian sign language recognition using eigen value weighted euclidean distance based classification technique", arXiv preprint arXiv:1303.0634 (2013).

Singleton, J. L. and E. L. Newport, "When learners surpass their models: The acquisition of american sign language from inconsistent input", Cognitive psychology **49**, 4, 370–407 (2004).

Skehan, P., *A cognitive approach to language learning* (Oxford University Press, 1998).

Smilkov, D., N. Thorat, Y. Assogba, A. Yuan, N. Kreeger, P. Yu, K. Zhang, S. Cai, E. Nielsen, D. Soergel *et al.*, "Tensorflow. js: Machine learning for the web and beyond", arXiv preprint arXiv:1901.05350 (2019).

Smith, C., E. M. Lentz and K. Mikos, *Signing naturally* (DawnSignPress, 2001).

Sohankar, J., K. Sadeghi, A. Banerjee and S. K. Gupta, "E-bias: A pervasive eeg-based identification and authentication system", in "Proceedings of the 11th ACM Symposium on QoS and Security for Wireless and Mobile Networks", pp. 165–172 (ACM, 2015).

Sole, M. M. and M. Tsoeu, "Sign language recognition using the extreme learning machine", in "AFRICON, 2011", pp. 1–6 (IEEE, 2011).

Soomro, K., A. R. Zamir and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild", arXiv preprint arXiv:1212.0402 (2012).

Starner, T., J. Weaver and A. Pentland, "Real-time american sign language recognition using desk and wearable computer based video", IEEE Transactions on Pattern Analysis and Machine Intelligence **20**, 12, 1371–1375 (1998).

Stokoe, W., D. Casterline and C. Croneberg, *A Dictionary of American Sign Language on Linguistic Principles* (Linstok Press, 1976), URL https://books.google.com/books?id=WjAFAQAAIAAJ.

Stokoe, W., E. C. for Linguistics and C. for Applied Linguistics, *The Study of Sign Language* (ERIC Clearinghouse for Linguistics, Center for Applied Linguistics, 1970), URL https://books.google.com/books?id=l4RCAAAAIAAJ.

Stokoe, W. C., "Sign language structure: An outline of the visual communication systems of the american deaf", Journal of deaf studies and deaf education **10**, 1, 3–37 (2005).

Stone, R., "Talking back required", https://www.rosettastone.com/, [Online; accessed 28-September-2018] (2016).

Strong, M. and S. F. Rudser, “An assessment instrument for sign language interpreters”, Sign Language Studies pp. 343–362 (1985).

Tai, Y., J. Yang and X. Liu, “Image super-resolution via deep recursive residual network”, in “Proceedings of the IEEE conference on computer vision and pattern recognition”, pp. 3147–3155 (2017).

Tennant, R. and M. Brown, *The American Sign Language Handshape Dictionary* (Clerc Books, 1998), URL `https://books.google.com/books?id=27WtFCWcEucC`.

Tome, D., C. Russell and L. Agapito, “Lifting from the deep: Convolutional 3d pose estimation from a single image”, CVPR 2017 Proceedings pp. 2500–2509 (2017).

Tompson, J. J., A. Jain, Y. LeCun and C. Bregler, “Joint training of a convolutional network and a graphical model for human pose estimation”, in “Advances in neural information processing systems”, pp. 1799–1807 (2014).

Tran, D., L. Bourdev, R. Fergus, L. Torresani and M. Paluri, “Learning spatiotemporal features with 3d convolutional networks”, in “Proceedings of the IEEE international conference on computer vision”, pp. 4489–4497 (2015).

Tubaiz, N., T. Shanableh and K. Assaleh, “Glove-based continuous arabic sign language recognition in user-dependent mode”, Human-Machine Systems, IEEE Transactions on **45**, 4, 526–533 (2015).

Tzeng, E., J. Hoffman, K. Saenko and T. Darrell, “Adversarial discriminative domain adaptation”, in “Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition”, pp. 7167–7176 (2017).

Valli, C. and C. Lucas, *Linguistics of American sign language: an introduction* (Gallaudet University Press, 2000).

Van der Kleij, F. M., R. C. Feskens and T. J. Eggen, “Effects of feedback in a computer-based learning environment on students learning outcomes: A meta-analysis”, Review of educational research **85**, 4, 475–511 (2015).

Venkateswara, H., V. N. Balasubramanian, P. Lade and S. Panchanathan, “Multiresolution match kernels for gesture video classification”, in “Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on”, pp. 1–4 (IEEE, 2013).

Vesselinov, R. and J. Grego, “Duolingo effectiveness study”, City University of New York, USA **28** (2012).

Vogler, C. and D. Metaxas, “Asl recognition based on a coupling between hmms and 3d motion analysis”, in “Computer Vision, 1998. Sixth International Conference on”, pp. 363–369 (IEEE, 1998).

Wilcox, S., *The phonetics of fingerspelling*, vol. 4 (John Benjamins Publishing, 1992).

Wilson, A. D. and A. F. Bobick, "Parametric hidden markov models for gesture recognition", Pattern Analysis and Machine Intelligence, IEEE Transactions on **21**, 9, 884–900 (1999).

Wu, R., S. Yan, Y. Shan, Q. Dang and G. Sun, "Deep image: Scaling up image recognition", arXiv preprint arXiv:1501.02876 (2015).

Yu, P., Y. Pan, C. Li, Z. Zhang, Q. Shi, W. Chu, M. Liu and Z. Zhu, "User-centred design for chinese-oriented spoken english learning system", Computer Assisted Language Learning **29**, 5, 984–1000 (2016).

Yun, L. K., T. T. Swee, R. Anuar, Z. Yahya, A. Yahya and M. R. A. Kadir, "Sign language recognition system using semg and hidden markov model", (2012).

Zhang, X., X. Chen, Y. Li, V. Lantz, K. Wang and J. Yang, "A framework for hand gesture recognition based on accelerometer and emg sensors", Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on **41**, 6, 1064–1076 (2011).

Zhao, W., "On automatic assessment of rehabilitation exercises with realtime feedback", in "2016 IEEE International Conference on Electro Information Technology (EIT)", pp. 0376–0381 (IEEE, 2016).