



A MULTI SENSOR APPROACH TO BOTSWANA SIGN LANGUAGE
DATASET WITH A VIEW OF ADDRESSING OCCLUSION

By

KABELO GERMANUS MADISE

Reg. No: 17100075

BSc (Hons) (Computer Systems Engineering) Sunderland

Department of Computer Science and Information Systems,

Faculty of Science,

Botswana International University of Science and Technology

Student E-mail Address, Phone Number including the code (+267)

Kabelo.madise@studentmail.biust.ac.bw, +267 74637449

A Dissertation Submitted to the Faculty of Science for the Study Leading to
Masters Dissertation in Partial Fulfilment of the Requirements for the Award
of the Degree of Master of Science in Computer Science of BIUST

Supervisor(s): Hlomani Hlomani, Ph.D.

Department of Computer Science and Information Systems,

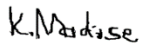
Faculty of Science, BIUST

hlomanihb@biust.ac.bw

Signature H. B. Hlomani Date: 26/10/20

Declaration and Copyright

I, **Kabelo Germanus Madise**, declare that this dissertation/thesis is my own original work and that it has not been presented and will not be presented to any other university for a similar or any other degree award.

Signature 

This dissertation/thesis is copyright material protected under the Berne Convention, the Copyright Act of 1999 and other international and national enactments, in that behalf, on intellectual property. It must not be reproduced by any means, in full or in part, except for short extracts in fair dealing; for researcher private study, critical scholarly review or discourse with an acknowledgement, without the written permission of the office of the Provost, on behalf of both the author and the BIUST.

Certification

The undersigned certifies that he/she has read and hereby recommends for acceptance by the College of ... a dissertation/thesis titled: ***A MULTI SENSOR APPROACH TO BOTSWANA SIGN LANGUAGE DATASET WITH A VIEW OF ADDRESSING OCCLUSION***, in fulfilment of the requirements for the degree of Master of Science/Doctor of Philosophy in (*Programme of Study*) of the BIUST.


.....

Dr. Hlomani Hlomani
(Supervisor)

Date: 26/10/20.....

Acknowledgement

I would first like to thank my thesis advisor Dr Hlomani Hlomani of the Faculty of Computer Science at Botswana International University of Science and Technology. The door to Dr Hlomani Hlomani's office was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this paper to be my own work but steered me in the right the direction whenever he thought I needed it.

I would also like to thank the experts who were involved in the validation survey for this research project: Botswana Deaf Society. Without their passionate participation and input, the validation survey could not have been successfully conducted. I would also like to acknowledge my best friend Mosimanegape T. Maake for reading of this thesis, A Special thank you to Kabelo, Edward, Knowell, Mike and Vincent, I am gratefully indebted for their valuable comments on this thesis.

I must express my very profound gratitude to my parents and to my partner for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you. Finally, I must thank God for making everything possible.

Abstract

Automatic Sign Language Recognition (ASLR) helps with converting hand gestures to spoken language, therefore, enabling communication between those able to hear and those unable to hear. There is abundant research work on ASLR of British Sign Language and American Sign Language. However, Botswana Sign Language has received less attention at least in terms of computational representation leading to automatic sign language recognition which can be attributed to lack of a Botswana Sign Language dataset Sign Language Dataset. Work done on other languages is not always directly applicable to Botswana Sign Language because sign languages differ significantly from country to country. A dataset plays a pivotal role in sign language recognition pipeline. However, one of the major challenges researcher's encounter is accurately extracting hands and fingers of a signer when the hands or fingers are not in the field of view of the camera (Occlusion). Researchers have argued that using multiple sensors addresses occlusion better than using a single sensor. This study proposes an approach to developing a Botswana Sign Language dataset based on tracking data from the Microsoft's Kinect sensor and the leap motion controller. The feature sets from both devices are combined in order to improve recognition performance (especially when occlusion). Recognition is performed by Support Vector Machines (SVM) and K Nearest Neighbor (KNN). The resulting dataset consisted of five thousand four hundred and thirty-three (5433) Botswana Sign Language gestures comprised of five (5) different sign words. The experimental results obtained show that recognition performance improves when compared to using one device to capture sign gestures. An overall recognition accuracy of 99.90% and 99.40% have been recorded using SVM and KNN respectively.

Table of Contents

Declaration and Copyright	2
Certification.....	3
Acknowledgement	4
Abstract.....	5
1. Introduction	8
1.1 Introduction to the problem	8
1.2 Background to the problem	10
1.3 Problem Statement	13
1.4 General Objective	13
Specific Objectives	13
1.5 Research Questions	13
1.6 Expected Outcomes	14
1.7 Justification of Study	14
1.8 Thesis Structure	15
2 Literature Review	16
Overview	16
2.1 Data Acquisition	17
2.2 Image Based Approach	17
2.3 Data Glove Approach	20
2.4 Sensor Based Approach	22
2.4.1 Kinect Based recognition	23
2.4.2 Leap Motion Based recognition	27
2.5 Multi Modal Approach	30
2.6 Summary	35
3 Methodology.....	41
3.1 Data Acquisition	41
3.1.1 The Development Environment and Accessing the Data	43
3.2 Pre-Processing.....	45
3.3 Feature Extraction.....	46
3.3.1 Finger direction	46
3.3.2 Fingertip position	47
3.3.3 Palm position	47

3.3.4	Elbow position.....	47
3.3.5	Wrist position.....	47
3.4	Synchronizing Data.....	47
3.5	Development of Algorithm	49
3.6	Experimental Setup.....	52
3.7	Evaluation	54
3.7.1	Gesture Classification.....	54
3.7.2	Performance Measure	57
3.7.2.4	False negative	58
4.	Experiment Results and Discussion.....	59
4.1	Data Collection	59
4.2	Gesture Classification.....	64
4.2.1	Support Vector Machines	64
4.2.2	K Nearest Neighbor	67
4.3	Comparative Analysis.....	70
4.3.1	Self-Comparative Analysis of the SVM and the KNN within this study.....	70
5.	Conclusion and future works	74
5.1	Introduction	74
5.2	Limitation of our work	74
5.3	Future work.....	76
	References	77

1. Introduction

1.1 Introduction to the problem

Sign language is a visual language that is used by the hearing-impaired community to communicate. To establish communication they use hand gestures, finger spelling, finger poses and movements of the arm or body, lip movements and facial expressions. Sign language is neither a simplified language nor a derived form of the spoken language. The grammar and semantics of sign language are unlike those of the spoken languages. The structure of a sentence in a spoken language is linear, one word followed by another, whereas in sign language, a simultaneous structure exists with a parallel temporal and spatial configuration. The configuration on a sign language sentence carries rich information about time, location, person or predicate [1], [2]. According to Rao et al. [3] sign language is not a universal language and it is different from country to country. In the United Kingdom there is British Sign Language (BSL), in the United States there is American Sign Language (ASL), and in Botswana there is Botswana Sign Language. In sign language there are no standards for signs and different groups of people can use different sign gestures to convey the same meaning.

According to Botswana Deaf Society, Botswana is said to have about sixty thousand (60 000) people living with disabilities some of them are hearing impaired. The primary sign language used in Botswana is Botswana sign language. The main difference between sign languages is that they use different sign gestures (hand positions, finger position and facial expression) to convey the same meaning [2]. Sign languages are developed over time among communities in order to establish communication. However, the main challenge in Botswana is that the majority of the hearing community are unable to understand sign language. Therefore, the hearing-impaired community are unable to seek help from public service providers because they lack the sign language skills. There are some professional sign language interpreters that can be able to serve as real time sign language interpreters, however, Sign language interpreters have proved to be expensive and not always available [2].

The alternative to sign language would be written communication; however, it is impersonal and even impractical when an emergency occurs. Fortunately, the progress of technology in human computer interaction has enabled humans to interact with computers. As such, researchers have

proposed to develop systems that capture sign gestures, then translate the sign gestures to speech or text in order to bridge the communication gap between the hearing and hearing impaired.

According to Kumar et al. [4], research in automatic sign language recognition started in the 1990s. Success in recent years of speech recognition systems has opened research in converting sign gestures to speech and text to enable communication between hearing and hearing impaired. Researchers have used different devices such as a digital camera, depth camera, data glove, accelerometer, Leap motion controller and Microsoft Kinect in order to acquire sign gestures, then use machine learning algorithms to recognize the sign gestures [2], [5].

However, there are a few challenges encountered by researchers when developing automatic sign language recognition system. There are two main approaches, which are for the most part based on image/vision, and the data glove. The vision/image-based approaches use a camera or depth sensors to acquire images or video inputs and then train machine learning algorithms such as Support Vector Machine (SVM), K Nearest Neighbor (KNN) and Artificial Neural Networks (ANN) to recognize the sign inputs. Ryumin et al [6], believes that the image-based approaches are affected by computer vision problems such as variation in lighting condition, cluttered background, camera resolution and occlusion. As such, in automatic sign language recognition, occlusion occurs when some parts of the hand or fingers are hidden from the field of view of the camera, making it hard to recognize input sign gestures. Furthermore, misrepresentation of sign gestures leads to misclassification and incorrect recognition accuracy [7]. With the data glove approach, the user has to wear a glove that has sensors and motion trackers. However, with the data glove approach, data is directly obtained from each sensor making this approach more accurate than the image-based approach. Although, the data glove is more accurate, it is costly to implement. It is also too uncomfortable to use and this limits its popularity [5].

In summary, automatic sign language recognition research on other sign language exists. Examples include those on the American Sign Language [8], [9], [10] where researchers have explored ways to automatically recognize sign language gesture and convert to text. However, the Botswana Sign Language has received little to no attention. Work done on other sign languages is not always directly applicable to Botswana sign language since every sign language is different from one another, for this reason each sign language requires its own dataset [2], [11]–[13]. There is a need for research on the automatic recognition of the Botswana sign Language to facilitate

communication between the hearing and hearing impaired by translating sign gestures to text and voice.

1.2 Background to the problem

Efforts to automatically convert sign language gestures to spoken language dates back to the early 90s. However, recently advancements in Human Computer Interaction has opened up new ways in which the hearing and hearing impaired can communicate with one another [14]. Many researchers (e.g. [8], [9], [10]) are trying to develop systems that convert sign language gestures to spoken language in various sign languages. This is achieved by capturing sign gestures and then converting the gestures into text or speech. The process of automatic sign language recognition entails hand tracking and segmenting hand movements and finger poses of the signer in every acquired frame of the image or video. The next step is recognizing hands, finger poses, facial expressions, head tilting, lip pattern and mouthing [15]. The recognition is commonly performed by machine learning algorithms such as K Nearest Neighbor (KNN), Artificial Neural Networks (ANN), Hidden Markov Model (HMM) and Support Vector Machines (SVM) [16]–[19]. From there, input sign gestures are matched to a word and then translated to spoken language.

Research in automatic sign language recognition can be achieved using two approaches which are the data glove and the image/vision based [4]. The image-based approach entails the use of a camera to capture either static or dynamic images. The images are then recognized using algorithms and sentences are produced from the interpretation of the images. The algorithms are also used to extract features from the video or image and remove unwanted background noise [11]. One of the challenges of the image based approach is that the image acquisition process is affected by various environmental conditions such as varying lighting condition and complex background [6]. Other environmental concerns are camera position, camera resolution and occlusion (e.g. when some of the fingers or the hand is hidden from the cameras view). The camera resolution also affects the images captured, a camera with a higher resolution produce high quality images. However, high quality images require more computational time and memory space [17]. Another automatic sign language recognition approach is the data glove approach. The user is required to wear a glove that consist of flex sensors and motion trackers. In the data glove approach tracking data (hand and finger positions) is directly obtained from each sensor [2]. The data glove is deemed to be more accurate. Rajaganapathy et al. [13] added that the glove approach requires less

computational time and that the response is fast. However, the glove approach is deemed to be expensive and intrusive as it can affect the signer's ability to sign naturally [5]. This has inspired the use non-intrusive image-based techniques for automatic sign language recognition [13].

According to Kumar et al. [2] automatic sign language recognition systems that are built from a single camera or sensor, achieve low accuracies when it comes to recognizing input signs [5]. Furthermore, single sensors fail to capture sign gestures accurately because all articulations of sign gesture cannot be captured by a single camera or sensor. Although environmental concerns such as background complexity, place of the sensor or camera make it hard to recognize sign gestures, however, the major challenge in automatic sign language recognition is occlusion. Occlusion occurs when aspects of a sign gesture are hidden from the field of view of the camera. Chandel et al. [20], suggested adding multiple views to capture hand shapes and finger poses to address occlusion. Ronchetti et al. [1], suggested the use of a constant background that matches the signer's shirt in order to address the issue of background noise. Some researchers choose to use colored gloves which makes it easier to track hands of the signer.

The advent of low price depth sensors such as the Leap motion controller and Microsoft Kinect has made it more convenient to capture 3-Dimensional data [7]. The Microsoft Kinect sensor has been used in different applications among which include robotics, gaming and gesture recognition. However, one the weakness of the device is that it is unable to track smaller body parts such as fingers, even though, in sign language finger spelling forms an integral part as it helps differentiate between similar sign gestures [2].

There exists research work that are based on the Microsoft Kinect for sign language recognition. However, they are deemed insufficient to acquire different hand and finger movements. Additionally, the systems fail to recognize the input sign gestures when occlusion occurs. In such a situation it becomes hard to correctly classify sign gesture [2]. It has been observed that Microsoft's Kinect sensor has gained popularity in recognizing gesture but the sensor is unable to detect finger movements [5]. Conversely, the Leap motion controller is able to capture three-dimensional data and it has been built for tracking hand and finger movements. However, automatic sign language recognition systems that use leap motion sensors fail mainly because leap motion sensors have a small field of view and they are only limited to acquire finger and hand movements [2]. The combination of the Kinect sensor and leap motion sensor would offer

complementary results. However, researchers have found it to be challenging to synchronize data from the two devices because of hardware configurations [2], [3], [4], [5].

There are research works on automatic sign language recognition of various sign languages that are based on multiple sensors for capturing sign gestures. However, we have not found any multi sensor-based work on Botswana Sign Language. It has been observed that automatic recognition systems that use multiple sensors are able to capture input signs accurately. An accurate input sign means there is less work for algorithms that perform recognition [21]. It has been pointed out that Sign Languages are different from country to country, each have their own lexicon and group of signs [13]. Sign Language recognition is, therefore, a problem that needs to be approached differently in each region because new gestures require new training data and likely entail new challenges that have not been considered before [1].

To the best of our knowledge, Botswana Sign Language has not received sufficient attention as there are no sign language datasets available. The issue that limits the development of robust automatic sign language recognition systems for different sign languages is the lack of available datasets for model training [16]. In order to develop models for automatic recognition of the Botswana Sign language there is a need for a dataset which researchers can be able to use. Success in other languages does not mean it will be the same with Botswana Sign Language. There is a need for a dataset which is developed based on the Kinect and leap motion sensor, as it has been pointed out from other research work that automatic sign language recognition based on multiple sensors achieves better recognition performance [2],[4],[5],[8]. The reason behind using two sensors is to address with the problem of occlusion. If one sensor is unable to capture a sign input, it will be captured by another sensor. Because of different hardware configurations in the Kinect and leap motion sensor there is a need for an algorithm that merges sign data from the two sensors. There is a need for a research level dataset that represents the group of signs used in the Botswana Sign Language. While the automatic sign language techniques are similar for different sign languages, training a recognition systems for a new sign language requires having an dataset for that language [1]. A dataset plays an important role in the automatic sign language recognition pipeline [12]. The reliability and accuracy of automatic sign language recognition systems rely on the quality of the training data [6]. Creating a dataset for training sign language recognition models involves being able to acquire and model all the signs gestures, the sign acquisition process is

subject to background noises, which can be handled by using a constant background different from the signer.

1.3 Problem Statement

Research on automated sign language recognition of Botswana Sign Language is lacking, which can be attributed to the lack of a Botswana Sign Language dataset. The problem that this research address, is the lack of Botswana Sign Language dataset or an approach to create such a dataset. In automatic sign language recognition, a dataset plays a pivotal role of training recognition models [1]. Research has revealed that robust sign language recognition systems are mostly built with multiple sensors [2], [4], [5]. Robust automatic recognition of Botswana sign language could assist in the translation process and the integration of the hearing impaired [1], [19]. Having a Sign Language Dataset is one of the phases in the automated sign language recognition pipeline. One of the major challenges faced by researchers in sign language recognition is extracting the signers hand shapes accurately [2], [5] Having the dataset based on a multi sensors approach is a research trend which many researchers are adopting in many different sign languages in an attempt to avoid occlusion during data capture, which attributes to increased accuracy of sign inputs.

1.4 General Objective

The main objective of this research is to develop a multi sensor-based dataset for Botswana Sign Language.

Specific Objectives

1. To identify approaches of synchronizing data from the Microsoft Kinect and the Leap motion controller sensors
2. To collect data from the Microsoft Kinect and the Leap motion controller sensors
3. To pre-process the acquired data
4. To develop an algorithm that merges data from the Microsoft Kinect and the Leap motion controller sensors.
5. Test and evaluate the dataset by using an existing sign language recognition algorithm.

1.5 Research Questions

1. How can data from the Microsoft Kinect and the Leap motion controller sensors be captured?

2. What are the approaches for synchronizing data from the Microsoft Kinect and Leap motion controller?
3. How does merging the data from the Microsoft Kinect and the Leap motion controller sensors impact the accuracy of the recognition of the captured sign input?
4. How do existing sign language recognition algorithms perform on the developed dataset?
5. What is the accuracy of the captured sign inputs?

1.6 Expected Outcomes

The expected outcomes of this study is ultimately a Botswana Sign Language Dataset. Leading to the production of the dataset, other contributions have been a demonstration of the multi-sensor approach to the development of the dataset with the view to address occlusion mostly based on the Microsoft Kinect sensor and the Leap motion controller. An interesting contribution has also been an algorithm that combines data from the Kinect and leap motion sensor. The benefit of using two sensors is to capture sign gestures accurately and to avoid occlusion if one sensor fails to capture input sign accurately the other sensor could assist. As mentioned by Ronchetti et al. [1] in order to have automated sign language recognition systems it is vital to have a dataset of the sign language. Automatic sign language recognition systems are important in bridging the gap between the hearing impaired and the hearing by converting sign gesture to speech or text.

1.7 Justification of Study

Research in automatic sign language recognition entails converting sign gestures to text or voice. There exist several researches works done in different sign languages that attempt to bridge the communication gap between the hearing and the hearing impaired. However, there are several challenges with the research area. There are a lot of environmental conditions that affect sign language recognition such as occlusion, background noise, low visibility, lighting, camera resolution and position of the camera. Occlusion seems to be one of the greatest problems in the world of computer vision. Occlusion occurs when the capturing device is not able to get a full view of the sign gesture. This happens when some parts of the hand or fingers are hidden from the cameras view. Occlusion leads to misrepresentation of sign gestures and incorrect classification of sign gestures. In order to address the problem of occlusion and build robust automatic sign language recognition systems, researchers (e.g. [2]–[5], [7]) have suggested the use of multiple

sensors for capturing sign gestures. The most commonly used sensors are the Microsoft Kinect and the leap motion controller. The Microsoft Kinect is mostly used for tracking body movements while the Leap motion is mostly used for tracking finger and hands movements.

Research on automated sign language recognition of Botswana Sign Language has received little to no attention. To the best of our knowledge Botswana Sign Language Dataset based on multiple sensors is lacking. Work done in other languages is not always applicable for Botswana sign language. Automatic sign language recognition is a problem that needs to be addressed differently in each country. New hand and finger movements require training data. Having a dataset is part of an important phase in the automatic sign language recognition pipeline. Having a Botswana Sign Language dataset could kick start research on systems that convert Botswana Sign Language to text or voice. Without a dataset, sign language recognition becomes impossible because classification algorithms rely on a dataset. Botswana sign language lacks a dataset, and this hinders research in automatic sign language recognition of Botswana sign language. This research will contribute towards bridging the communication gap between the hearing and the hearing-impaired Batswana and help them to enjoy the benefits of improved human computer interaction by building a Botswana Sign Language Dataset.

1.8 Thesis Structure

The dissertation is organized as follows:

- **Chapter two** reviews current methods in sign language and current studies that have been carried out in this research area.
- **Chapter Three** is the Methodology, which explains the experimental setup and how the research will be carried out to achieve the stated objectives.
- **Chapter Four** discuss the results obtained from this research.
- **Chapter Five** summarizes the work carried out and future plans to improve the research.

2 Literature Review

Overview

The hearing-impaired face challenges when attempting to establish communication with the hearing. The principal concern is that most of the hearing are unable to comprehend sign language communication [34]. Sign language interpreters can be engaged to facilitate the communication; however, their services are expensive and having an interpreter to facilitate communication leads to loss of privacy. According to Oz et al. [22] many hearing impaired individuals communicate with the hearing in the form of writing, however, it is deemed to be inconvenient when two people are far from each other but need to communicate. In an attempt to recognize sign languages, several research works have been carried, however, the complex body and hand movements during gesture communication prove to be a challenge.

Automatic sign language recognition is a research area that aims to translate sign gestures to speech or text. This chapter will discuss some of the existing approaches to automatically translating sign language to spoken language. Most of the Automatic Sign Language Recognition systems follow the same steps, Figure 1 depicts how most of them work. First is data acquisition, where sign gestures are captured and movements of body parts such as hands, fingers, arms, and lips are captured using cameras or sensors. The next step entails extracting features from the captured input data. The captured input data provides information about manual or non-manual information. Even though non-manual features like lip movement, facial expressions and head movements can change a sign gesture altogether most sign gestures are expressed by using manual features. Then the captured details about the features are classified using machine learning algorithms. The classification phase requires a sign language dataset to correctly classify new sign gestures [7]. The last phase entails converting the captured gesture to text or voice.

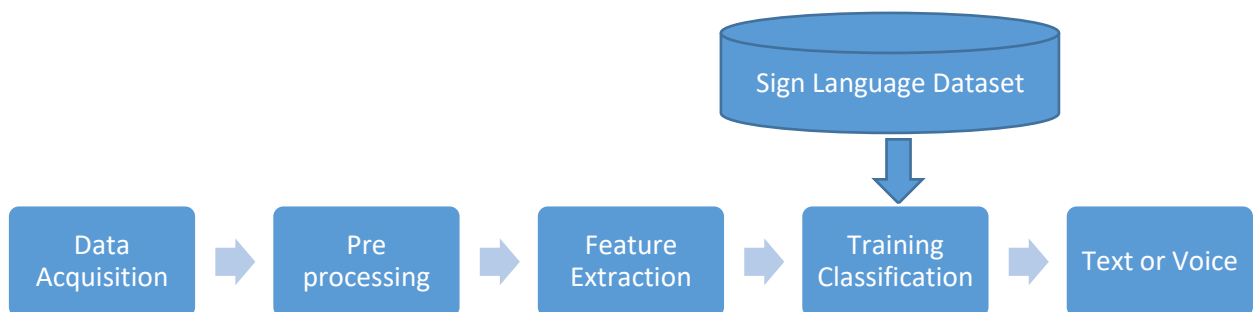


Figure 1 How Automatic Sign Language Recognition works

2.1 Data Acquisition

Research in automatic sign language recognition can be achieved by using one of the two approaches such as data glove or image. In the image-based approach input sign gestures are acquired using a camera. The acquired images contain details about finger poses, finger direction, palm position, arm position, arm movements, facial expressions and hand positions. The captured image is further processed to isolate the signers face, arms, hands and fingers from other objects contained in the image. Although, the image-based approach is practical and natural for real time applications, however, they are affected by different environmental conditions. In the glove-based approach, however, it is difficult to detect other features such as facial expression and lip movements during gesture communication. However, hand and finger attributes are derived with ease. The downside of the glove-based approach is that the gloves are uncomfortable and expensive. Furthermore, the data glove approach is deemed to be intrusive to the signer and costly to implement, researchers prefer the image-based approach [2]. The image-based approach has evolved from 2D to 3D using depth sensors analysis. Using the 2-Dimensional approach gesture classification heavily relies on extracted features using image processing algorithms such as contour modelling, segmentation and shape detection. However, 3-Dimensional hand gesture recognition is a novel way which has various applications.

2.2 Image Based Approach

Several research works [4],[5],[8], [18] have been done in an attempt to overcome the barrier of communication between those able to hear and unable to hear by developing systems that translate sign gestures automatically to spoken language using cameras or sensors. Cameras are used to capture the sign gestures and machine learning algorithms are used recognize hand gestures. Kumar et al. [2] argued that recognition of sign gestures using the cameras or sensors is preferred by most researchers over the data glove approach, mainly because users are not required to wear devices which can hinder finger and hand movements.

Ronchetti et al. [1] revealed that sign languages are different from country to country and that their syntax and semantics are different. The authors further revealed that translating sign language automatically to spoken language is a problem that needs to be tackled differently for different sign languages, several researchers agree with this viewpoint [2], [4], [5], [8], and [7].

Ronchetti et al. [1] proposed to develop an Argentinian Sign Language dataset. Ten (10) subjects recorded sixty-four (64) different sign gestures of the sign language. The collected dataset was called LSA64. The dataset was recorded using a standard RGB camera. Several research works have suggested that using a 2-Dimensional camera to capture sign gestures is affected by different environmental conditions which make it difficult to accurately capture sign gestures [2], [4], [5]. Therefore, to overcome the problem of environmental background complexity Ronchetti et al. [1] used colored gloves during sign gesture capture to make it easy to extract hand regions from the image.

Ravikiran et al. [17] proposed to develop an algorithm that would determine the number of open fingers on a captured image that represents a sign gesture of the American Sign Language alphabets. Knowing how many fingers are open it is then possible to then determine if a gesture belongs to a class of gestures that have fingers open or not. Finger detection is achieved by boundary tracing and fingertip detection. Even though the authors did not use colored gloves, the camera used to capture the sign gestures was placed very close to the signer, a constant white background was used, and the lighting was controlled. The camera resolution used was 640x480, the choice of camera used would make it a challenge to extract hand gesture accurately if two hands of the signers captured or when occlusion occurs. Shanableh et al. [23] presented a user independent system for recognition of hand movements of the Arabic sign language. Furthermore, for classification of hand movements they used K Nearest Neighbor (KNN) and Polynomial classifiers and the overall classification accuracy achieved is 87%. There was no background restriction. They used a single camera for sign input acquisition. The developed system is more suitable for mobile platforms as the algorithms require less storage and less computation power. However, the authors collected the dataset from three (3) different signers who were required to wear colored gloves. Furthermore, each word was performed fifty (50) times by each signer resulting in dataset with three thousand four hundred and fifty (3450) video samples. Unfortunately, their approach had a small number of different signers which resulted in the classifiers giving inaccurate results when tested with people of different body sizes from those that recorded the dataset. Additionally, Chansri et al. [11] added that the use of restrictions such as colored gloves does not help the system to generalize.

Work done by Das et al. [12] presented a system that recognize alphabets (A ,B ,C ,D) sign gestures based on movements and shape of the lip in real time. The system uses a USB webcam to access the live video and an object detector tool from MATLAB was used to derive facial expressions. The overall recognition accuracy achieved was 90.67%, the total number of tested samples was forty-three (43). Although, their system achieved a high recognition rate the camera used is of low resolution, if the sample size was increased and more subjects were used to record sign gestures their recognition rate could decrease. Kumar et al. [4] suggested that machine learning algorithms used to recognize sign gestures requires the data to be repeated many times for them to be able to classify new data and avoid overfitting. Additionally, the system is only limited to alphabets spelling and Kumar et al. [5] revealed that alphabets alone cannot be used to achieve communication with those who are unable to hear.

The majority of systems which recognize sign language gestures fail to include both manual features (hand gesture, finger poses and hand motions) and non-manual features (facial expression, lip movements and head movements). Shohieb et al. [24] proposed to develop a dataset containing manual and non-manual features of the Arabic Sign Language. They captured the sign gesture under controlled lighting and a constant background. According to Shohieb et al. [24] automatic sign language recognition requires having a comprehensive dataset and that the available sign language dataset contain a few gestures. However, their approach only included alphabets and numbers which are not enough to establish communication with the hearing impaired. Sign gestures of alphabets mostly use one hand, words however use two hands and words can establish communication when compared to alphabets. Most existing automatic sign language datasets are of alphabets of various sign languages; however, datasets of words are few. The lack of publicly available sign language dataset hinders progress in the research area and datasets having fewer gestures.

Most automatic sign language recognition systems are created under controlled environment. This is because sign language recognition using cameras is affected by environmental factors such as the background environment. According to Rao et al. [3] tracking hands in a disorderly background environment is a major challenge in this research area. Another challenge in automatic sign language recognition is the background lighting of signer relating to the contrast of the light where the camera is placed. Rao et al. [3] believes that systems developed under controlled environments

can never work in the real world. Rao et al. [3] developed a real time system that recognize the Indian Sign Language (ISL) hand movements to spoken language under complex background. The collected dataset contained eighteen (18) sign gestures recorded by ten (10) different individuals. The application is developed for mobile smart phones and uses the front selfie camera to capture video input, the video input is then processed, and then gestures are converted to the corresponding meaning. The fact that the system is developed for mobile platform makes the system convenient and accessible since the majority of people have smart phones, so they can download the application and start communicating. The overall recognition accuracy achieved using Artificial Neural Network (ANN) is 90%. However, most smart phones front selfie camera have low resolutions which may decrease the recognition accuracy [1]. The proposed system is not tested under lab conditions of controlled lighting, constant background and colored gloves. Notably, complex background environment is one of the gaps in recognition of hand gestures as complex backgrounds make it difficult to retrieve the hands and fingers of the signer. Most researchers address this problem by putting a constant background and the use of colored gloves.

Divers carryout duties underwater which prevents them from bringing electronic devices that would facilitate communication. In gesture recognition cameras are mostly used to acquire hand gestures, however, their use underwater is difficult due to its cloudiness and lack of light. Gustin et al [25] proposed a concept of using a high resolution multi beam sonar for diver hand gesture recognition. Hand gesture are recognized using Support Vector Machines and Convex Hull method. The results obtained on more than a thousand (1000) samples using the convex hull method achieved an accuracy of 92% and 94% using Support Vector Machines. The fusion of the two-classification approaches achieved 99%.

2.3 Data Glove Approach

The data gloves-based approaches use gyroscope or accelerometer to monitor hand gestures. The sensors are used to track and detect fingers and hands movements. The glove consists of electric cables that connect to each sensor which then connect to a computer.

Oz et al. [22] proposed to develop a system to recognize American Sign Language(ASL) words and translating the ASL gestures to spoken Language. Sign gestures were captured using a sensory data glove. The acquired data was about hand shapes and finger joint angles. Oz et al. [22] used Artificial Neural Networks for classification of signs in real time, and the overall accuracy

achieved is 90%. The data glove approach is deemed to be more accurate than vision-based techniques however for sign gestures that require the use of both hands requires two gloves which could lead to a costly setup and complex configuration [5].

Most studies are aimed at developing systems that convert sign languages to text, however not so many are aimed at teaching learners finger movements during sign language communication. Using a data glove Iwasako et al. [26] developed a finger motion skill learning system. The system assists learners to identify motion mistakes instinctively by overlapping a skilled person's 3Dimensional models. The glove approach is deemed more suitable for accurately capturing finger movements and poses, as Microsoft Kinect sensors fail to get detailed motion data of the hands as it's unable to recognize small body parts like fingers. Similarly, Shukor et al. [27] developed a system that uses a data glove to acquire hand and finger movements in order to translate the Malaysian Sign language alphabets to spoken language. The overall accuracy recorded from four individuals who tested the device was 95% using Hidden Markov Model.

Pezzuoli et al. [28] Proposed an application that converts American Sign Language words and phrases to speech and text using a data glove based device. The data glove was used to capture arm and finger movements. The data glove is equipped with multiple sensors to track movements of the fingers and an accelerometer to track arm movements. The application is able to recognize sign gestures (sentences and words) and translate to spoken language, however, Pezzuoli et al. [28] pointed out that there are some challenges associated with using the device which includes the cost of the device, the comfort of the device and also restriction in finger and hand movements due to the number of cables connected to the device.

Similarly, López-Noriega et al. [29] developed software that helps those unable to hear communicate with those able to hear by identifying the positions of fingers and hands, using 5DT gloves. The aim is to enable hand signs to be translated into spoken language using audio signals. Recognition was performed using Artificial Neural Networks. Also, an unexperienced user of the system is able to communicate using basic sentences, this shows that the system is intuitive and realistic.

The glove approach is deemed to be more accurate than image-based approaches especially with capturing sign gestures accurately. One of the major challenges in this research area is accurately capturing sign gestures which could be hindered by various environmental conditions (e.g. lighting

conditions, camera resolution) and occlusion. However, the data glove approach addresses occlusion better as each sensor is connected directly to a finger, which means each finger's movement is captured separately.

According to Oz et al. [22] for sign gestures to be captured accurately through the image based approach requires the use of multiple cameras, which would require complicated image processing methods that would require more processing power and slow recognition rates. With the glove approach there is no need to perform preprocessing techniques as the sign gestures are captured accurately. However, the limitation of the glove approach is that for gestures that require the use of both hands, the use of two data gloves is required, which is costly [36]. Moreover, a user is required to put on a glove which is connected to cables which can restrict the user's natural movements of hands and fingers. Furthermore, the glove approach fails to include other features such as lip movements or facial expressions, however, Kumar et al. [5] revealed that sign language is not only reliant on the movements of the hands but include non-manual features such as facial expressions. Kumar et al. [5] further revealed that non- manual features such as facial expressions could be used to distinguish between sign gestures with similar hand and finger movements.

2.4 Sensor Based Approach

Research work in the area of recognizing sign gestures and translating them to spoken language has progressed from 2-Dimensional image to 3-Demisional using depth sensors (for example Microsoft Kinect and Leap motion controller). The 2-Dimensional approach use a standard camera to capture sign gesture and retrieve hands attributes from the captured image using machine learning algorithms. However, 3-Dimensional hand gesture recognition is a new approach and various researchers have employed it in various applications such as automatic sign language recognition systems, robotics, and gaming. The introduction of low-cost depth sensors such as the Leap motion controller and Microsoft Kinect has increased their use in tackling the problem of automatic sign recognition systems. In this section of the document we discuss some of the existing sensor-based recognition.

2.4.1 Kinect Based recognition

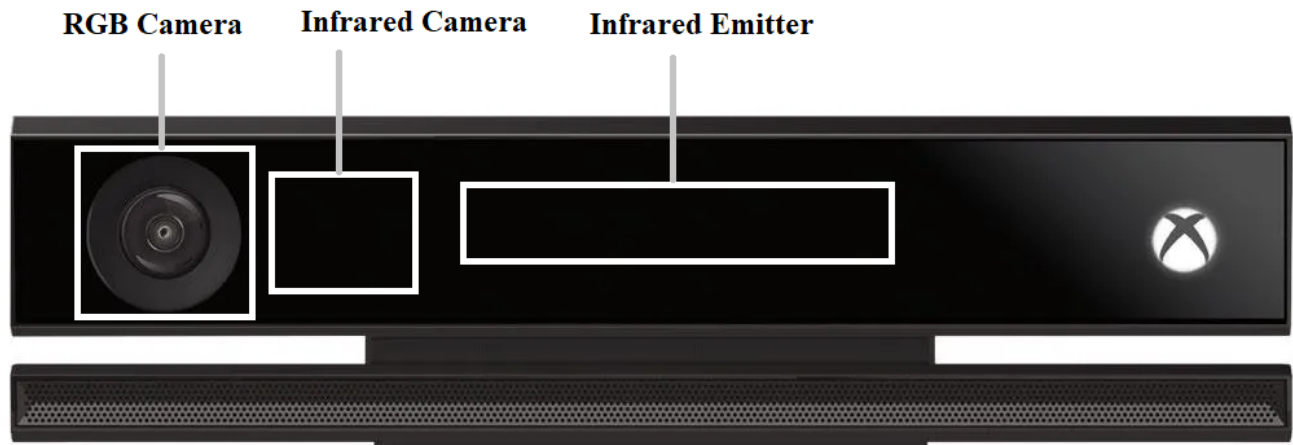


Figure 2. Microsoft Kinect 2.0 for Microsoft Windows Personal computer and Xbox One.

Microsoft Kinect is a commonly used devices in gesture recognitions. The Microsoft Kinect is a motion tracking device which Microsoft built for both the Windows Personal computers and Xbox gaming console. The device is able to track twenty-five (25) joints within its field of view, Figure 3 provides a list of joints which can be tracked by the device. The devices can be able to track people within a range of 0.5 to 4.5 meters. The device connects to the Windows Personal computer via USB. The second generation of the sensor, the Microsoft Kinect Version 2 for windows depicted in Figure 2 was introduced in 2014. The device has an Red Green Blue (RGB) camera and infrared depth sensor. The infrared depth sensor enables the device to capture and detect surrounding objects in varied lighting conditions and cluttered background [30]. The device is also equipped with a freely accessible Software Development Kit (SDK) which allows developers to access gesture tracking information. The Microsoft Kinect Version 2 has improved features from those of the first release (Microsoft Kinect 360) which includes more skeleton joints [31]. The tracked positions of the Kinect V2 are more accurate when compared to those of the Kinect 360. Even though the Kinect provides tracking information about different body parts and joints it lacks the ability track fingers as confirmed by Elons et al. [30].

For this research, the Microsoft Kinect v2 was used to capture sign gesture however most existing research work uses the old version, this could be attributed to the fact that Microsoft stopped manufacturing the Microsoft Kinect v2. In this section of the document we discuss some of the existing works done using the Microsoft Kinect.

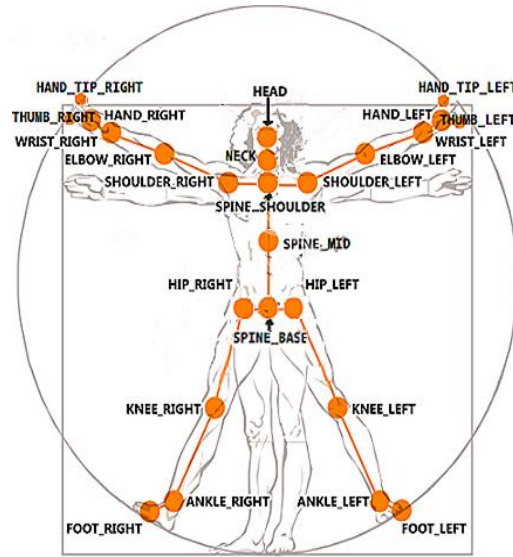


Figure 3 Human Joints tracked by the Microsoft Kinect

Ahmed et al.[32] Developed an application that would facilitate communication between those able to hear and those unable to hear by translating gestures to speech and speech to sign gestures. Gestures are performed in front of a Microsoft Kinect; the system then converts the gestures to speech. The Microsoft Kinect has a built-in microphone which is used to capture speech and then the application displays an animated sign gesture that matched the recorded speech. The resulting dataset contained four hundred and fifty (450) video samples. However, their dataset was too small to be used with recognition techniques such as Hidden Markov Model (HMM), Support Vector Machines (SVM) or Artificial Neural Networks(ANN) [32]. Therefore, they used Euclidean Distance for classification of feature vectors. Furthermore, the overall accuracy achieved was 83% under different states of occlusion. Their approach focused on using Microsoft Kinect Version 2 gesture builder which omits finger poses and movements. Although, fingerspelling forms an integral part of sign language communication [2].

Aliyu et al. [33] developed a system for recognizing gestures of the Arabic Sign Language using Microsoft Kinect to acquire the gestures. The system was tested on twenty (20) sign gestures of which each was performed ten times resulting in a sample size of two hundred (200). In their approach, depth images are acquired from the device. Then, they used hand segmentation algorithm to extract hand regions because the Microsoft Kinect is unable to track small body parts such as fingers as confirmed by Kumar et al. [2]. Furthermore, they achieved an overall recognition

accuracy of 99.8% using Linear Discriminant Analysis. However, their dataset had fewer samples which could result in decrease in the recognition results when more samples are added [7].

Work done by Lai et al. [34] offered an approach to recognise hand gestures in real-time by using features acquired from Microsoft Kinect skeletal tracking data. The Microsoft skeletal tracking data provides coordinate information about the different human joints. However, in their research the authors found out that not all joints data provided by the device significantly contributes to hand gesture recognition. Then, identified left elbow, right elbow, left wrist, right wrist, left hand and right hand as human joints that are mostly relevant for hand gesture recognition. Similarly, this research adopted the approach of accessing skeletal tracking data with the focus being on left elbow, right elbow, left wrist and right wrist. Furthermore, in their approach twenty (20) individuals were used to perform eight (8) gestures five (5) times. Then, K-Nearest Neighbor was used to perform classification and achieved an overall recognition of 97.25%.

Work done by Sun et al. [35] proposed an approach to recognize American Sign Language (ASL) with Microsoft Kinect. In their approach color images and skeletal tracking data from the device were utilized. Their dataset comprised of seventy-three (73) ASL sign gestures which were recorded by nine (9) individuals and performed each sign gesture (3) times resulting in a dataset with one thousand nine hundred and seventy-one (1971). For extracting, they used Histogram of Oriented Gradients (HOG) to extract hand regions from the depth image. Then, skeletal tracking data provided by the Kinect included elbows, shoulders, wrist and hands. Finally, they used Support Vector Machines (SVM) to perform recognition and achieved a recognition rate of 86.0%.

Similarly, work done by Dominio et al. [36] proposed an approach to recognize hand gestures by exploiting color images and depth information provided by the Microsoft Kinect. In their approach the first dataset collected contained ten (10) different gestures performed by ten (10) different individuals who repeated each gesture ten (10) times resulting in a dataset of one thousand (1000) samples. The second dataset used contained twelve (12) American Sign Language (ASL) gestures performed by fourteen (14) individuals who repeated each sign ten (10) times. Then, Support Vector Machines were used to perform recognition and achieved an overall accuracy of 100% on dataset with ten (10) gestures and 97.60% on ASL dataset. However, their approach was not robust to sign gestures when the fingers were folded or when occlusion occurred. Although, their

approach achieved high recognition accuracy their results show that increase in dataset affects their accuracy.

In an attempt to make Automatic Sign Language Recognition system more realistic Sulfayanti et al. [37] offered an approach to recognise alphabets of the American Sign Language using the Microsoft Kinect in real time. The overall recognition accuracy achieved using K-Nearest Neighbor was 69.79%. Furthermore, their approach was not robust to accurately recognize gestures that had fingers open. Similarly, in an attempt to make Sign Language Recognition systems more suited for the real world Carneiro et al. [38] offered an approach to recognize the Brazilian Sign Language static sign gestures using Microsoft Kinect. Differently from most studies they captured images under complex and varying lighting conditions. The dataset was collected from four (4) individuals who performed the same gesture in varying environments (well-lit indoor, indoor low light, outdoor natural light). The overall recognition accuracy achieved using eigenfaces and eigenhands is 89%.

One of the major challenges in this research area, is that the image acquisition is affected by environmental lighting and complex background. Chansri et al. [11] addressed the challenge by developing a system that converts Thai Sign Language to text under complex environment. Their results showed that their application is robust to track hand gestures under complex environments and they achieved an overall recognition rate of 84.05%. However, their hand segmentation algorithm was not tested using two hands to determine if the algorithm would be able to extract two hands from the image.

Most research works have been able to develop automatic sign language recognition systems based on the Microsoft Kinect. The device is perfectly suited for hand and body tracking as per the sentiments echoed by Elons et al. [30]. Unfortunately, the Microsoft Kinect is unable to recognize hand shapes and finger poses but, in sign language they are critical to differentiate between similar sign gestures. However, researchers have addressed this problem by accessing the depth image from the device which requires machine learning algorithms to extract hand and finger features thus increasing the computational load. Differently from their approach our research uses gesture tracking data which is calculated by the device. Marin et al. [7] confirms through his work that tracking data calculated by the device is more accurate than accessing the raw image from the device. Aliyu et al. [33] revealed that the Microsoft Kinect sign language recognition systems fail

because the Kinect is unable to track fingers which is very important and Kumar et al. [5] agrees by suggesting the use of Leap Motion Controller together with the Microsoft Kinect. Several [2],[4],[7] researchers have argued that using a single sensor is not enough to capture all the articulations of sign gestures especially when complex gestures are performed they further revealed using more than one sensor reduces occlusion.

2.4.2 Leap Motion Based recognition



Figure 4 Leap motion controller

The leap motion controller depicted in Figure 4 is a sensor that is built for tracking hand and finger movements [39] without the user touching it. The device connects to the computer via USB. A major issue for the device, is that for it to accurately detect hand gesture, gestures need to be performed between the ranges of twenty-five (25) to six hundred (600) millimetres above the device. The device has an accuracy of about two hundred (200) μm which makes it appropriate for gaming and gesture recognition applications [39]. The device has two built in cameras each capturing frames at the rate of two hundred (200) frames per second. Cohen et al. [40] revealed that the devices consist of three Light Emitting Diodes (LED) incorporated into the two cameras and added that the infrared enables capturing sign gestures under varying lighting conditions and cluttered background. The device comes with a freely accessible Software Development Kit (SDK). Researchers can use the device's Application Programming Interface (API) to access the device's tracking data. However, the device does not provide access to images of gestures that are performed. In this section we discuss some of the existing automatic sign language recognition systems based on the leap motion controller.

Work done by Pramunanto et al. [39] offered an approach to capture the Indonesian Sign Language using the leap motion controller and used Naïve Bayes for classification. The authors selected nineteen (19) tracking features automatically calculated by the device. One person performed each of the ten (10) sign gestures twenty-five (25) times. Using one person for gesture acquisition makes their work biased since it does not represent a variation of individuals. Differently from their approach our dataset was collected from ten (10) individuals who comprised of both males and females. All the gestures performed using a single hand, differently from their approach all our gestures are performed using both hands. The problem experienced by the authors was that during sign gesture capture some sign gestures were inaccurately captured because the device could not detect some fingers. Additionally, they recommended that the Leap motion controller should be used away from direct sunlight as it affects the infrared sensor of the device. The overall accuracy achieved is 80.53%. Although, their recognition accuracy is high their sample size is small and could prove to be a problem when new sign gestures are to be classified, as their approach does not avoid overfitting.

Potter et al. [41] developed a system for recognizing gestures of the Australian Sign Language using the Leap motion controller. Their results showed that the device has a problem of misrepresenting sign gestures and that it is nearly impossible to acquire sign gestures when the hands are hidden from the view of the device or when occlusion occurs. Furthermore, they decided that one (1) leap motion is not enough to accurately capture complex hand and finger movements. For this reason, they suggested the use of two (2) leap motion controllers to capture sign gestures simultaneously in order to reduce occlusion. Similarly, Cohen et al. [40] developed a system to recognise alphabets of the Israeli sign language using the leap motion. Their main focus was on fingerspelling. They performed recognition of the gestures using Support Vector Machines and achieved recognition accuracy rate between 85% - 92%. However, their study was only limited to alphabets and failed to include gestures performed using both hands. Differently from their work this study focused on words sign gestures which according to Potter et al. [41] require complex hand movements when compared to alphabets. Kumar et al.[4] Revealed that recognition systems based on fingerspelling alone are not enough to facilitate communication between those that can hear and those unable to hear.

Similarly, work done by Chuan et al. [42] offered an approach to recognize twenty-six (26) alphabets of the American Sign Language using the Leap Motion Controller. They obtained finger features accessed through the Application Programming Interface (API) of the device. However, they revealed that not all hand and finger features provided by the device is relevant to sign language recognition, for this reason researchers choose only meaningful features for their work. In their approach they used two (2) individuals to capture the two (2) sets of datasets. Then, used K-Nearest Neighbor (KNN) and Support Vector Machines (SVM) to classify sign gestures, they achieved recognition rates of 72.78% and 79.83% respectively. Furthermore, they believed that the reason for incorrectly classified gestures was because of the Leap motion. They observed that during sign gesture acquisition the visual feedback of the performed sign gesture was different from the gesture performed.

Work done by Kumar et al. [43] proposed an approach to recognize fifty-six (56) Indian Sign Language gestures that include single and double hands using the leap motion controller. The resulting dataset had two thousand two hundred and forty (2240) gestures which were recorded from ten (10) people who repeated each gesture four (4) times. In their approach they focused on capturing manual sign and finger spelling. They used Support Vector Machines and Bidirectional Long Short-Term Memory Neural Networks and achieved an overall recognition accuracy of 63.57%.

Majority of existing approaches focus on static gestures, however, work done by Elons et al. [30] offered an approach to recognise fifty (50) dynamic sign gestures of the Arabic Sign language using the leap motion controller. The sign gestures are recorded from four (4) different people. They used Multi-layer perceptron Neural Network and achieved an overall recognition of 88%. Unfortunately, their approach failed to include non-manual features which can be important in differentiating between similar sign gestures because the Leap motion controller is unable to track non-manual features. Additionally, they recommended the addition of another sensor to track the non-manual features to enhance their system. Moreover, the other problem encountered by the authors is inaccurately capturing sign gestures. The authors could improve their work and reduce the effect of occlusion by including the Microsoft Kinect to track body and facial expressions and combine data from the two devices, these sentiments are echoed by Kumar et al. [5]. Although, the leap motion controller is highly accurate at tracking hand and finger movements,

however, it has a problem of misrepresenting sign gestures when regions of the hand are self-occluded or out of the field of view of the device. Shao et al. [44] reveals that misrepresentation of sign gestures affects both the Microsoft Kinect and Leap motion controller based applications. Shao et al. [44] further points out that the use of more than one device could reduce the effect of occlusion. Several authors [2], [5],[7] confirm that using more than one sensor for sign gesture capturing reduces the effect of occlusion and increases the recognition rate.

2.5 Multi Modal Approach

Several [2], [4], [7],[45] research work suggest that using a single device is not enough to capture all the hand and finger movements. Some sign gestures during data acquisition are incorrectly captured especially when some or parts of the sign gesture are not in the field of view of the device (occlusion). Misrepresentation of sign gestures leads to incorrect classification leading to low recognition rates [2]. In an attempt to capture sign gestures more accurately researchers have proposed hybrid frameworks and models by combining more than one device or sensors.

Work done by Marin et al. [7] offers an approach to combine different feature sets from the leap motion controller and the Microsoft Kinect to improve the recognition of hands and fingers. In their approach they accessed 3Dimensional tracking data computed by the Leap motion. Furthermore, the raw image captured by the Kinect was accessed, then relevant features are extracted from the image. In an attempt to improve recognition accuracy features from the leap motion controller were concatenated with those of the Microsoft Kinect. The captured sign gestures were of a subset of the American Sign Language alphabets. All gestures were performed with a single hand. The resulting dataset contained ten (10) sign gestures which were performed by fourteen (14) individuals who repeated each gesture ten (10) times. Their overall recognition accuracy achieved was 91.28% using Support Vector Machines showing better recognition accuracy than those of single sensor-based recognition. However, their focus was on accurate gesture recognition rather than developing a system with a larger gesture vocabulary. Differently from their approach this research was not focussed on finger spelling and gestures that are performed using a single hand, this research offers an approach that requires both hands and the focus is on recognizing sign language words instead of alphabets. As such, words require complex hand and arm configuration so the inclusion of features such as wrist and elbow positions are important.

Similarly work done by Kumar et al. [4] offers an approach to merge data from the leap motion and Microsoft Kinect. In their approach ten (10) different people performed twenty-five (25) sign words from the Indian Sign Language and the two devices were positioned at different places while data was captured simultaneously. Using Coupled Hidden Markov Model (CHMM) for recognition they achieved accuracy rate of 90.80%. However, their approach was tested on a small-scale dataset which had a total of two hundred (200) sign gestures which were performed using a single hand. A major challenge in this research area is the lack of publicly available dataset and existing datasets having fewer sign gestures.

Additionally, another major challenge in this research area is correctly acquiring sign gestures when gestures are hidden from the view of the sensor. In an attempt to address the problem of occlusion Kumar et al. [2] developed a model that combines data from the Microsoft Kinect and leap motion controller. The motivation being that if one device fails to capture sign gestures accurately the data from the other device would be used. They extracted raw depth images from the Kinect and applied image processing techniques to extract relevant hand and finger features. However, using the Kinect depth image requires extensive computations for extracting finger and hand features, hence increasing the computational load. Differently from their approach this study offers an approach to use skeletal tracking data computed by the Microsoft Kinect. They extracted relevant tracking data computed by the Leap motion controller and concatenated with the features extracted from the Kinect depth image. Their dataset contains fifty (50) Indian Sign Language dynamic sign gestures performed by ten (10) different people and each sign was repeated fifteen (15) times resulting in sample size of seven thousand five hundred (7500). Their approach achieved an overall accuracy of 97.85% and 94.55% using Hidden Markov Model (HMM) and Bidirectional Long Short-Term Memory Neural Network (BLSTM-NN) based sequential classifier. They discovered that using data from the combined features achieves better results than using the devices separately. However, their approach was restricted to hand and finger movements and omitted arm tracking data which are essential in differentiating between similar sign gestures, while the approach presented in this study includes tracking data for the wrists and elbows.

Work done by Mohandes et al. [46] offers an approach to recognize gestures of the Arabic Sign Language using two Leap motion controllers to address the problem of occlusion. The two leap motion devices were placed perpendicular from one another to acquire input signs to prevent cases

where one finger is occluded by another finger or hand. Leap motion controller is able to provide tracking information about the hands and fingers. In their approach the two devices were used to capture twenty-eight (28) Arabic Sign Language alphabets. Feature fusion from the two devices gives 97.7% classification accuracy with Linear Discriminant Analysis classifier. From the experiment carried out, better recognition was observed when the data from the two devices was fused together as opposed to when classification was performed on data from an individual Leap motion controller. Their research was limited to static sign gestures that only require the use of one hand to perform. The leap motion controller is only limited to hand tracking their research failed to include other features which are relevant to sign languages such arm movement's facial expressions and lip movements. Their work was only focused on recognition of alphabets, however, research focused on recognition of alphabets alone is rendered incomplete, mainly because hearing impaired people communicate using words [47].

Similarly work done by Deriche et al. [45] offered a different approach to address the problem of occlusion and missing tracking data (finger and hand tracking data) by combining tracking data from two (2) leap Motion Controllers. In their approach one (1) leap motion is used to capture the frontal view and the other is used to capture the side view of the signer's hand gestures. If one device does not have a good view of the sign gesture, tracking data from the other device is used and if both devices have a good view of the sign gesture the features are combined together. Their dataset consisted of one hundred (100) dynamic words of the Arabic Sign Language which were performed by two (2) people. Each sign gesture was repeated ten (10) times resulting in a dataset with two thousand (2000) samples. Their approach achieved an overall recognition of 92% on Arabic Sign Language dynamic sign gestures. Even though, their approach achieved a high recognition rate and robustness to occlusion they omitted non-manual features like movements of body and facial expressions.

However, Kumar et al. [5] pointed out that a sign language recognition system that omit the signer's facial expressions is incomplete. Facial expressions are vital in differentiating between similar gestures [2]. Work done by Kumar et al. [5] offered an approach to incorporate the signer's facial expressions along with hand gestures. In their approach they used Microsoft Kinect 360 sensor to capture the facial expressions of the signer while the leap motion controller is used to capture hand and finger movements. They collected a dataset of fifty-one (51) dynamic sign words

of Indian Sign Language and this was recorded from ten (10) people and each sign was repeated eight (8) times resulting in a dataset of four thousand and eighty (4080) samples. The recognition was performed using Hidden Markov Model (HMM) then applied Independent Bayesian Classification Combination achieved a recognition accuracy of 97.06%. Their results show better results when compared to recognition using a single sensor. However, facial expressions are mostly ignored by researchers because of their complexity.

Work done by Athitsos et al. [19] offers an approach to develop a dataset of the American Sign Language. The dataset is developed as part of system that allows users to be able to lookup meanings of gestures. The sign inputs are captured simultaneously using four standard cameras. Two cameras are used to capture the frontal views, one camera capturing the signer's side view and the other one capturing the zoomed in face of the signer. The resulting dataset requires more computational power and large storage space. Their work used standard 2-Dimensional cameras which are affected by various environmental conditions. However, during capturing of sign gestures they failed to address varying environmental conditions and resorted to collecting the dataset under laboratory conditions (constant background and controlled lighting).

Sreejith et al. [47], developed a system that allows a user to browse through images using gestures. Their approach used skeletal tracking data computed by the Microsoft Kinect. To compensate for the Kinect's inability to track fingers they combined it with the leap motion controller's tracking data of fingers and hands. They used five (5) individuals to perform each gesture ten (10) times resulting in a dataset with fifty (50) gestures. They reported a recognition rate of 94%. Unfortunately, their approach was focused on an application with a smaller vocabulary. Similarly, this research adopted the approach of accessing 3-Dimensional tracking data computed by the Kinect instead of accessing the raw image from the device.

Leap motion controller and Microsoft Kinect have influenced their use in image-based sign language recognition due to their high recognition performance and low cost. Image based sign language recognition approach has a few challenges, one of them is self-occlusion. Self-occlusion occurs when part of the finger or hand is hidden from the camera view. Kiselev et al. [48] proposed to address the challenge of self-occlusion with three (3) leap motion controllers to capture numeric sign gestures, their research work has shown that increasing the number of cameras for sign gestures capture can lead to better recognition accuracy. Kiselev et al. [48], argued, that the leap

motion controller has a small field of view, but a higher resolution as compared to the Microsoft Kinect, however the use of the two devices complement each other when the Microsoft Kinect is used for body tracking and the leap motion controller is used for hand tracking.

Previous work has been limited to developing systems that acquire gestures using a single device (For example using a Leap motion controller or a Microsoft Kinect). However, it is nearly not possible to acquire all gestures with a single sensor [24]. When occlusion happens or if the hand is hidden from the camera. Occlusion leads to misclassification of sign gestures because the camera fails to capture the sign gesture accurately. The recognition results obtained from using multiple sensors are significantly higher than those based on single sensors.

Majority of research that captures sign gestures using the leap motion and Microsoft Kinect have shown that it is possible to combine data from the two devices even though the devices are configured differently. They achieved this by extracting relevant features from the two devices and combining the features into one feature set. Works done by [7],[2], [4], [5] have combined data from the two devices, however, their approaches involved extracting data from the raw depth image from the Kinect which then required image processing techniques to be employed to extract relevant features. This study adopted the approach from the research done by Kumar et al. [2] of concatenating gesture tracking data from the two devices however, their approach was limited to hand and finger tracking, they also extracted raw images from the Kinect. Differently from their approach this study offers an approach to use the data computed by the Microsoft Kinect and to include wrist and elbow tracking features which were omitted in their approach.

2.6 Summary

Table 1 Summary of related Literature

Method	Authors	Gestures	Restrictions	Approach (features and classifier)	Dataset and accuracy
Data Glove	Oz et al. [22]	Finger joint angles and hand shapes of the American Sign Language (ASL) 50 ASL words recorded		Artificial Neural Networks	An overall accuracy of 90% is achieved
	Shukor et al. [27]	Malaysian Sign Language alphabets recorded by 4 individuals			Average recognition accuracy was 89%
	Pezzuoli et al. [28]	Device is able to detect arm and five finger movements of American sign Language phrases			Average recognition rate of 74%
	López-Noriega et al. [29]	Fingers and hands movements		Artificial Neural Networks	
2D images	Ronchetti et al. [1]	64 sign gestures of Argentinian Sign Language, recorded by 10 non experts executed 5 times	Colored gloves were used during acquisition, limited to finger spelling only	Hidden Markov Model and Gaussian Mixture Model	3200 sign gesture samples recorded, 95% overall accuracy achieved
	Ravikiran et al. [17]	American Sign Language alphabets, dataset was recorded from 5 different individuals	Limited to finger detection, controlled lighting and constant background	Boundary tracing combined with finger tip detection	95% overall accuracy
	Shanableh et al. [23]	23 words of the Arabic Sign Language recorded	Colored gloves	K-Nearest Neighbor and polynomial classifiers	3450 video samples recorded, Reported a

	Das et al. [12]	from 3 people 50 times Lip movements of alphabets A, B, C, D		Histogram of Oriented Gradients of extracted lip image is used as features, Artificial Neural Network for classification	recognition rate of 87% A total of 43 video samples recorded, 90.67% recognition accuracy was achieved
	Shohieb et al. [24]	Facial and static hand gestures of Arabic sign Language	Lighting and background laboratory conditions	Signs World Atlas	97% and 95% recognition
	Rao et al [3]	18 sign gestures recorded by 10 individuals Indian Sign Language		Artificial Neural Networks	Overall Accuracy of 90%
Kinect based	Ahmed et al.[32]	30 words of Arabic Sign Language		Euclidean Distance	Dataset consist of 450 videos samples, overall accuracy of 83%
	Aliyu et al.[33]	20 words Of the Arabic sign Language		Linear Discriminant Analysis	Dataset Consist of 200 samples, Average accuracy 99.8%
	Lai et al. [34]	8 simple gestures performed by 20 individuals performed 5 times		K-Nearest Neighbor	Overall accuracy 97.25%
	Sun et al. [35]	73 signs from American Sign Language, performed by 9 individuals repeated 3 times		Support Vector Machines	Dataset of 1971 samples, Overall accuracy

	Dominio et al. [36]	10 gestures performed by 10 individuals repeated 10 times, 12 American Sign Language gestures performed by 10 individuals repeated 10 times	Support Vector Machines	achieved is 86.0% 10 gestures recognition accuracy of 100 %, 12 gestures recognition accuracy of 97.27 %
	Sulfayanti et al. [37]	American sign language alphabets	K-Nearest Neighbor	And overall accuracy achieved is 69.79%
Leap Motion Based	Pramunanto et al. [39]	10 sign gestures of Indonesian Sign Language recorded by 1 person, repeated 25 times	Naïve Bayes	Dataset of 250 samples, Overall accuracy of 80.53%
	Potter et al. [41]	26 Australian Sign Language Alphabets	Artificial Neural Networks	
	Cohen et al [40]	Israeli Sign Language Alphabets	Support Vector Machines	Accuracy between 85% and 92%
	Elons et al. [30]	50 dynamic gestures of the Arabic Sign Language recorded from 4 individuals	Multi-layer perceptron Neural Network	Overall accuracy of 88%
	Chuan et al. [42]	26 American Sign Language alphabets recorded from 2 individuals	Support Vector Machines and K-Nearest Neighbor	Achieved recognition of 79.83% and 72.78 respectively
	Kumar et al. [43]	56 gestures of Indian Sign Language recorded from 10 people who repeated each gesture 4 times	Support Vector Machines and Bidirectional Long Short-Term Memory	Dataset of 2240 samples with overall recognition of 63.57%

Multimodal Based Approaches			Neural Networks	
Leap motion + Kinect	Marin et al. [7]	10 sign gestures of American Sign Language alphabets recorded by 14 people repeated 10 times	Support Vector Machines	Dataset of 1400 samples With overall accuracy of 91.28%
	Kumar et al. [4]	25 sign words Indian Sign Language repeated 8 times by 4 people	Coupled Hidden Markov Model	Dataset of 200 samples Overall recognition of 90.80%
	Kumar et al. [2]	50 sign words Indian Sign Language repeated 15 times by 8 people	Hidden Markov Model and Bidirectional Long Short- Term Memory Neural Network	Dataset of 7500 recognition of 97.85% and 94.55%
	Mohandes et al. [46]	28 Arabic Sign Language alphabets	Linear Discriminant Analysis	Recognition of 97.7%
	Deriche et al. [45]	100 words Arabic Sign Language repeated 10 times by 2 people	Linear Discriminant Analysis and Gaussian Mixture Model	Dataset of 2000 samples Recognition of 92%
	Kumar et al. [5]	51 words Indian Sign Language repeated 8 times by 10 people	Hidden Markov Model	Dataset of 4080 samples recognition of 96.05% and 94.27%
	Sreejith et al.[47]	10 gestures recorded by 5 people		Overall recognition between 94% and 100%

Research in automatic sign language has two well-known approaches glove based and image-based approaches. The glove-based approach requires a signer to wear a glove that has motion sensors that track hand and finger movements. It is deemed more accurate when it comes to capturing sign gestures accurately, however, it affects the users from signing naturally which is why the image-based approach is preferred. With the image-based approach the signer's sign gestures are captured using a camera or sensor and machine learning algorithms are used to perform feature extraction and recognition. The recognition phase converts the sign gestures to text or speech. However, in order to train machine-learning algorithms a dataset is required.

The lack of publicly available datasets for different sign languages is major issue in this research area. Furthermore, existing sign language datasets are recorded under laboratory conditions where the signer is required to wear colored gloves for easy extraction of hand features, or a constant background is used to avoid background noise. Existing datasets also have only a few captured sign gestures however machine-learning algorithms require more repetition to give recognition results that are not biased. Work done on other sign languages is not directly applicable because sign language is not a universal language and different from country to country [2], [3], [5]. Existing automatic sign language systems are focused on incomplete recognition, with recognition of alphabets only not adding other relevant features which are relevant to sign languages such as arm movements, lips or facial expressions. The other challenge faced in this field is the problem of occlusion, which happens when some fingers or hands are hidden from the view of the camera. Occlusion leads to the misrepresentation and misclassification of sign gestures [3]. Many researchers [2], [3], [4], [5], [28], [40] have attempted to address the problem of occlusion by using multiple sensors for capturing sign gestures and have achieved promising results.

As per our understanding the majority of existing approaches for sign language recognition fail to accurately acquire sign gestures. Although, the glove approach is the most accurate in terms of capturing input signs, the set up involved is rather complex, the device is also expensive and using the device is disruptive to the signer. Researchers prefer the image-based approach; however, it is affected by many environmental apprehensions such as background, lighting, camera resolution, and position of the camera. As per our understanding single sensors-based systems fail to deliver a good view of the gestures when a part of the gesture is hidden from the camera or when occlusion

occurs. It has been observed that using multiple sensors for data capture achieves better results and can handle occlusion.

3 Methodology

This chapter of the document outlines details on the approach to the creation of Botswana sign language dataset. Researchers have different approaches of carrying out sign language recognition in different sign languages [1], [2], [4], [5], [7]. However, they follow the same pattern, which usually includes dataset collection (or acquisition), data preprocessing, feature extraction, training and classification. Without a dataset, sign language recognition becomes impossible because classification algorithms (which are used for recognition) rely on a dataset. Botswana sign language lacks a dataset, and this hinders research in automatic translation of sign gestures to spoken language. One of the major challenges in automatic translation of sign gestures to spoken language is accurately capturing sign gestures. Occlusion affects the ability of sign gestures to be captured accurately. Varying lighting and complex background also lead to misrepresentation of sign gestures and poor recognition accuracy.

This study offers an approach to capture Botswana Sign Language sign gestures accurately using the Microsoft Kinect and the Leap motion controller. The sensors were placed at different viewing positions from each other. Since the Leap motion controller has a restrictive range of view, it is therefore, limited to acquiring fingers and hand positions, it is used to capture data about finger direction, fingertip position and palm position. The Microsoft Kinect was used to track the movements of the elbows and wrist since it is unable to track fingers. In sign language fingertip position, direction and orientation, and hand direction play a key role in differentiating between gestures, these sentiments are supported by Marin et al. [7]. According to Marin et al. [7] Using multiple sensors captures sign gestures more accurately and addresses occlusion better than single sensor. This study offers an approach to address occlusion by combining data from the two sensors as depicted in Figure 5. The problem of complex background and varying lighting is addressed by using depth sensors as they are not affected by varying lighting and complex background as confirmed by Kumar et al. [5] .

3.1 Data Acquisition

The two depth sensors (Leap motion controller and Microsoft Kinect) provides access of 3-Dimensional data about the scene through the Application Programming Interface (API). In this study input sign gestures are acquired at the same time from the two sensors. In contrast to this study's approach, sign gestures recorded using a standard camera, hand segmentation and

recognition becomes difficult due to occlusion, background complexity and varied lighting. However, depth sensors are not affected by issues of background noise so there is no need to use a constant background when sign gestures are being recorded, contrary to the norm.



Figure 5 Microsoft Kinect and Leap Motion Controller Simultaneously Capturing Sign gestures
[2]

The data acquisition set up is depicted in by Figure 5. The leap motion controller was positioned below the signer's hands and fingers. The Microsoft Kinect sensor was positioned in front of the signer to capture vertical and horizontal positions of the elbows and wrist during sign gestures. Sign gesture data was captured simultaneously from both sensors.

3.1.1 The Development Environment and Accessing the Data

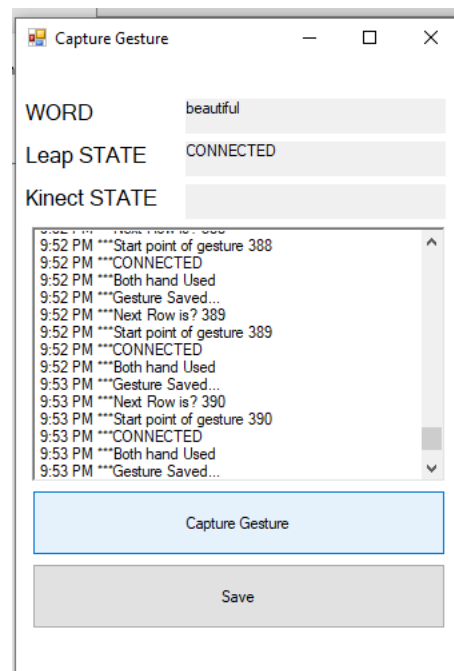


Figure 6 Graphical User Interface (GUI) of the C# Developed application for capturing sign gestures

To capture the sign gesture data from the Leap motion controller and Microsoft Kinect there was a need to develop an application that can access the data from the two devices and combine the extracted features. The laptop that was used to develop the C# application for capturing data from the two sensors had the following specification: Intel(R) Core (TM) i3-4005u CPU @1.70GHz. To access data from the two devices, Microsoft Visual studio 2017 and the C# programming language was used to develop the application. The Software Development Kit (SDK) used for the leap motion controller version used was the c# version 3.2.1 and the Microsoft Kinect version used is version: 2.0.1410.19000.

Weka was used for classification. Weka version 3.8.3 is a tool that contains a collection of machine learning algorithms. Because not all classification algorithms required are implemented in Weka v 3.2.1, python programming language version 3.6.6 was therefore used to implement other classification algorithms using Jupyter Notebook version 5.7.4. Figure 6 depicts the Graphical User Interface for the developed application. Each sign gesture has an associated meaning, so each time the sign gesture is captured the user must input any of the selected Botswana Sign Language words in the word text field. The user continues to capture the sign gestures, if there are any errors

during the capture the application will feedback. Once all the gestures are captured, they are saved to an excel file.

```
Hand firstHand = hands[0];
_list.Add(wordTxt.Text);

String handType1 = firstHand.GetType().ToString();

Hand Direction

Palm Position
_list.Add(palmPositionX.ToString());
_list.Add(palmPositionY.ToString());
_list.Add(palmPositionZ.ToString());

#region Finger Position
//get details about the fingers
List<Finger> fingers = firstHand.Fingers;
foreach (Finger finger in fingers)
{
```

Figure 7 Snapshot of C# code of leap motion controller's API used to access tracking data

Figure 7 depicts how the leap motion's API was used to access the device's tracking data of performed sign gestures. The Frame class provides access to the frame from the leap motion. The position details of the detected hands were accessed through the Leap motion controller API's Hands class. Through the Hand class we accessed the details about the fingers. The fingers class provides different attributes about the fingers such as the direction and the tip position of the fingers.

```

private void Reader_MultiSourceFrameArrived(object sender, MultiSourceFrameArrivedEventArgs e)
{
    var reference = e.FrameReference.AcquireFrame();
    // Body
    using (var frame = reference.BodyFrameReference.AcquireFrame())
    {
        if (frame != null)
        {
            _bodies = new Body[frame.BodyFrameSource.BodyCount];
            frame.GetAndRefreshBodyData(_bodies);
            foreach (var body in _bodies)
            {
                if (body != null)
                {
                    if (body.IsTracked)
                    {
                        // Get the details about the Elbow And Wrist
                        elbowLeft= body.Joints[JointType.ElbowLeft];
                        elbowRight = body.Joints[JointType.ElbowRight];

                        wristLeft = body.Joints[JointType.WristLeft];
                        wristRight = body.Joints[JointType.WristRight];
                    }
                }
            }
        }
    }
}

```

Figure 8 Snapshot of C# code of Microsoft Kinect's API used to access tracking data

Figure 8 depicts a snapshot of the Microsoft Kinect API used to access tracking data. Frame data is accessed from the Microsoft Kinect, each frame contains different details about the signer's body tracking details. The body class provides access to the different body joints such as feet, hip, shoulders, elbow and wrist. However, for this research the focus is on the elbow and wrist as they are considered more relevant for sign gestures, as confirmed by Marin et al [7].

3.2 Pre-Processing

Most research in automatic sign language recognition access the raw images from the Microsoft Kinect and perform feature extraction using machine learning algorithms to extract hand regions, However, complex computations are required in order to acquire data about the sign gesture. In contrast, this research uses 3-Dimensional tracking data that is computed by the device instead of accessing the raw depth images. Therefore, pre-processing techniques for extracting hand and finger attributes were not required. Additionally, accessing the gesture tracking data computed by the devices is deemed more accurate and reduces the computational load [2], [4], [5], [7].

3.3 Feature Extraction

Some of the features provided by the API of the devices are not relevant for sign language, therefore, there is a need to derive more meaningful features. In this section, we present details of the 3-Dimensional features extracted from the sign gestures. The selected features have been tried and tested by other research works in various sign languages [2], [4], [5], [7]. Through the API of the Microsoft Kinect, information about elbow positions and wrist positions were captured using the developed application depicted in Figure 6. As said earlier, the Leap motion controller was used to capture finger direction, fingertip position and palm position.

For purposes of this research, the selected words of Botswana sign language require the user to perform sign gestures using both hands. The extracted features were of the two hands and arms. Sub section 3.3.1 to 3.3.5 defines the features of interest to this research which are also depicted in Figure 9.

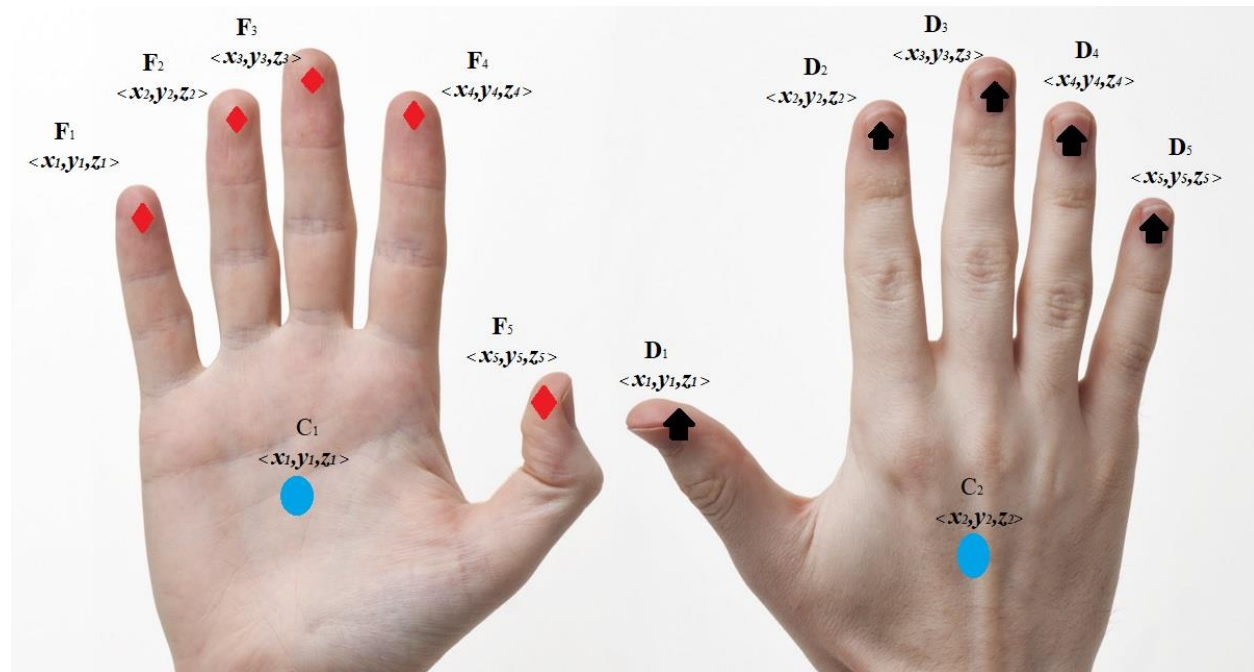


Figure 9 Set of features extracted by the leap Motion controller [2]

3.3.1 Finger direction

The finger direction gives information about the pointing direction of the finger and the details are returned in a vector of X, Y and Z. The pointing direction differentiates between similar sign gestures that have different finger position.

$D_i, i = 1, \dots, 10$ represents the 3D positions of the detected finger from the two hands where each position consists of $\langle X_i, Y_i, Z_i \rangle$ vectors in 3-Dimensional space.

3.3.2 Fingertip position:

Likewise, the fingertip position gives information about the 3-Dimensional point vector of the end point of each finger.

$F_i, i = 1, \dots, 10$ represents the 3-Dimensional fingertip positions of the detected fingers from the two hands where each position consists of $\langle X_i, Y_i, Z_i \rangle$ vectors in 3-Dimensional space.

3.3.3 Palm position

Similarly, the palm position gives information about the center position of the palm from the leap motion controller in millimeters. The data returned by the leap motion controller is a vector representing coordinates of the palm position.

$C_i, i = 1, \dots, 2$ represents the 3-Dimensional positions of the center of each hand where each position consists of $\langle X_i, Y_i, Z_i \rangle$ vectors in 3-Dimensional space.

Sub section 3.3.4 and 3.3.5 define the features are extracted from the Microsoft Kinect:

3.3.4 Elbow position

The elbow position gives 3-Dimensional information about the elbow of each hand when sign gestures are captured.

$E_i, i = 1, \dots, 2$ represents the 3-Dimensional positions of the elbow of each hand where each position consists of $\langle X_i, Y_i, Z_i \rangle$ vectors in 3-Dimensional space.

3.3.5 Wrist position

Likewise, the wrist position gives 3-Dimensional information about the wrist of each hand when sign gestures are captured.

$W_i, i = 1, \dots, 2$ represents the 3-Dimensional positions of the wrist of each hand where each position consists of $\langle X_i, Y_i, Z_i \rangle$ vectors in 3-Dimensional space.

3.4 Synchronizing Data

The use of a single device fails to accurately capture sign gestures. To overcome the problem of occlusion and misrepresentation of sign gestures during gesture capture, this research proposes the use of the Leap motion controller and Microsoft Kinect device and keeping the devices in different views. The limitation of the Microsoft Kinect being unable to capture fingers is addressed by using the leap motion to capture hand and finger tracking data. However, because the Leap motion is unable to capture elbows and wrist tracking data, the Microsoft Kinect is used to capture elbows and wrist tracking data. The two sensors complement each other while data is being collected. Several research works [2], [4], [5], [7], [12] have shown that it is possible to combine data from the two devices even though they have different hardware configurations. This has been achieved by extracting features from the two devices and combining the extracted features. This research adopted the approach from [2], [5], [7] to combine features from the devices as depicted in Figure 10.

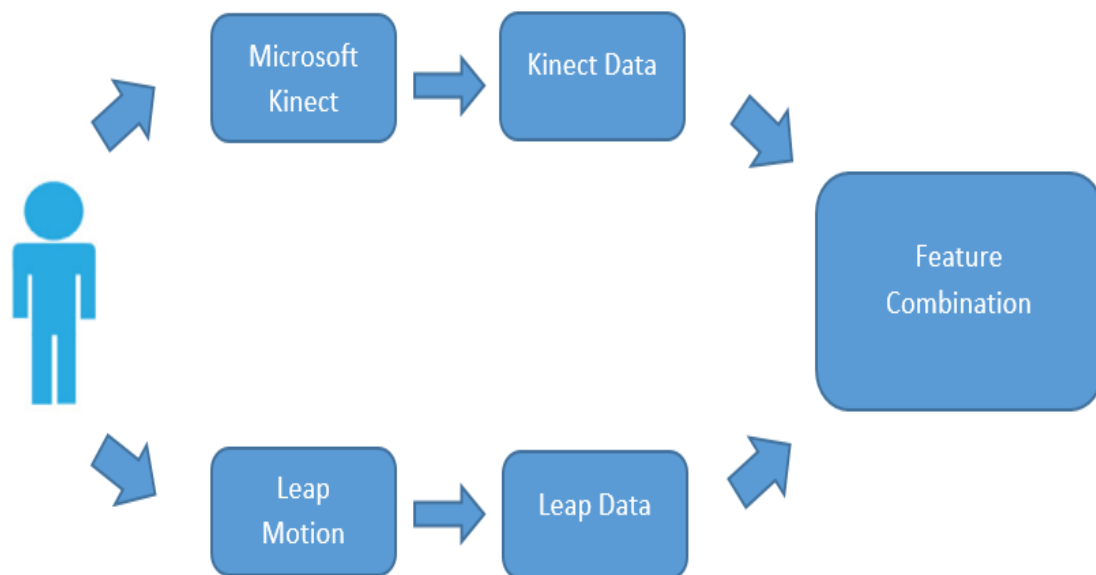


Figure 10 Diagram of the Approach Proposed

In this section we describe the details of the selected approach. The features derived from the Microsoft Kinect form a feature vector, and those derived from the Leap motion controller form another feature vector. The two feature vectors are then concatenated together resulting in a new feature vector.

Each captured sign gesture is described by a set of two feature Sets:

- The set $f_{leap} = [D, C, F]$ consist of all the features extracted from the leap motion controller. The features contain 3D information about the finger direction, the hand palm position and the finger position.
- The set $f_{kinect} = [E, W]$ consist of all the features extracted from the Microsoft Kinect. The features contain 3D information about the elbow and wrist positions of the two hands.

The complete features set is obtained by concatenating the two-feature sets.

$$f_{leap+kinect} = [f_{leap} + f_{kinect}]$$

3.5 Development of Algorithm

In this section we discuss the algorithm that enables sign gesture data to be captured simultaneously from the two sensors. The developed application grants access to hand gesture tracking data through each device's Application Programming Interface. The two devices capture sign gesture simultaneously from different viewing angles. The two devices provide different data; therefore, a set of different features are derived from the depth sensors. In our approach, our algorithm concatenates the extracted features from the two sensors. Depicted by Figure 11 is a flow chart of how our proposed algorithm works. Figure 12 shows C# code snippet of how features from the two devices are concatenated.

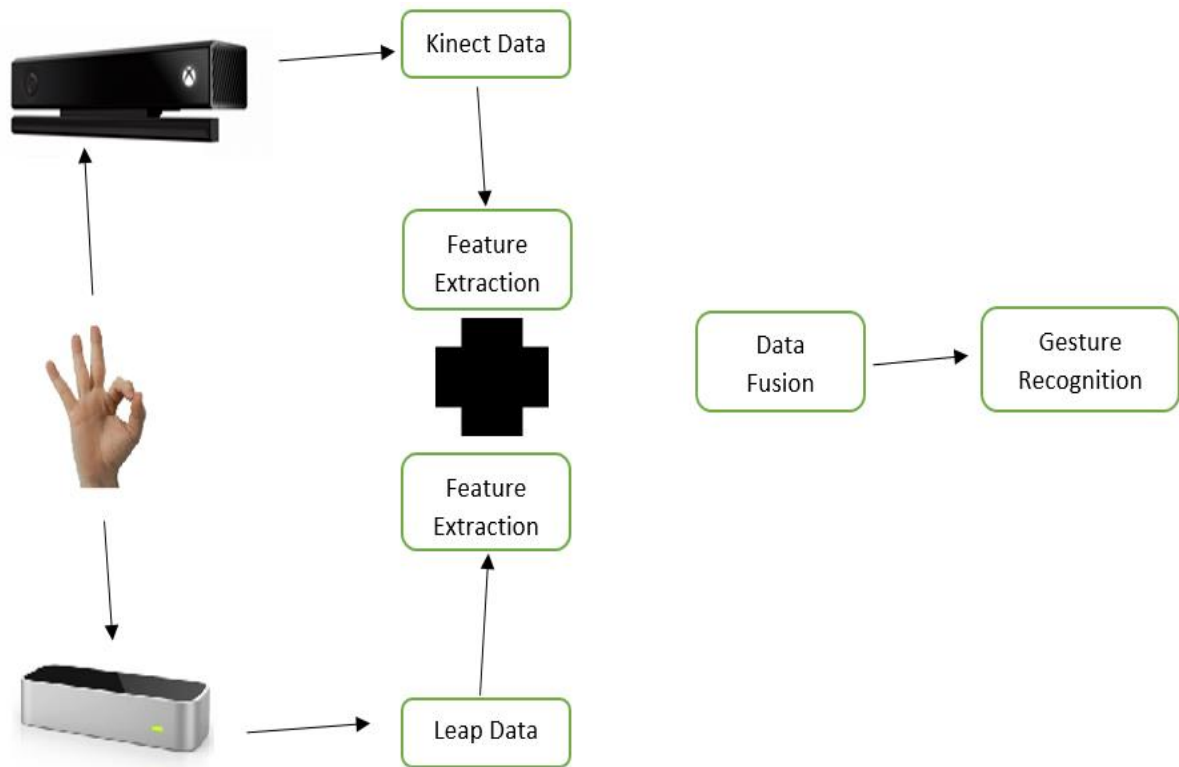


Figure 11 Diagram showing how the proposed feature combination algorithm works

```

#region Kinect Coordinates

//left Elbow Position
_KinectData.Add(elbowLeft.Position.X.ToString());
_KinectData.Add(elbowLeft.Position.Y.ToString());
_KinectData.Add(elbowLeft.Position.Z.ToString());

//LEFT Wrist Position
_KinectData.Add(wristLeft.Position.X.ToString());
_KinectData.Add(wristLeft.Position.Y.ToString());
_KinectData.Add(wristLeft.Position.Z.ToString());

//RIGHT Elbow Position
_KinectData.Add(elbowRight.Position.X.ToString());
_KinectData.Add(elbowRight.Position.Y.ToString());
_KinectData.Add(elbowRight.Position.Z.ToString());

//Rght Wrist Position
_KinectData.Add(wristRight.Position.X.ToString());
_KinectData.Add(wristRight.Position.Y.ToString());
_KinectData.Add(wristRight.Position.Z.ToString());

#endregion

//Merge data From the Two sensors
var _mergedDataFromTheSensors=_leapMotionData.Concat(_KinectData);

```

Figure 12 C# code Snippet showing combining tracking data from the Microsoft Kinect and Leap motion controller

In order to establish communication with the hearing, the hearing-impaired use hand gestures, finger spelling, finger poses, movement of the arm and body. To build a robust sign language recognition system a combination of hand movements and body movements have to be incorporated [49]. For this research, a combination of the finger positions and directions was combined with arm positions. Figure 13 shows the Pseudo-code for concatenating data from the sensors. Before sign gestures are saved the word identifying the sign gestures is entered. Since this study is restricted to capturing static Botswana Sign Language gestures that require the use of both hands, the number of hands detected has to be two. The computed tracking data from the two devices is accessed and relevant features are selected. Then the features from the two devices are

concatenated. Finally, the concatenated features are saved to a Microsoft Excel Comma Separated Values (CSV) file.

Algorithm 1: How to obtain data from Kinect and Leap Motion

```

initialization;
while While number of      detected hands is greater than 0 do
    Input word corresponding to sign gesture      performed ;
    if detected hands count is equal to two then
        Extract hand and finger features from leap Motion ;
        Extract elbow and wrist features from the Kinect ;
        Combine Features from leap motion the Kinect ;
        Save combined Features to csv file ;
    else
        print 'Please use both hands';
    end
end

```

Figure 13 How to obtain data from Microsoft Kinect and Leap Motion Controller

3.6 Experimental Setup

In this section we describe how the Botswana sign language dataset was created. A dataset is required to perform sign language recognition. For this research, sign gesture data computed by the Microsoft Kinect and Leap motion controller were accessed through the Application Programming Interface. Furthermore, selected features were then concatenated by the developed algorithm. Every sign gesture performed was saved to one Microsoft Excel file. For a sign language dataset to work with machine learning algorithms, there is a need for the dataset to be properly labelled as argued by Ryumin et al. [6]. For every sign gesture captured, the set of attributes from the two sensors were used to distinctly identify the different columns in the excel file, each row identifies a single record for a sign gesture.

Every sign language dataset contains a list of either alphabets, numbers, words or sentences. Most sign language dataset are mostly for alphabets or the datasets has fewer instances. Contrary to this, this research was focused on creating a dataset for words. Unlike alphabets, words can be used to

communicate with the hearing impaired. There are two types of sign language gestures used for sign language communication - , static and dynamic gesture. Static gestures are gestures that remain unchanged over time (hands and finger poses remain unchanged). Dynamic gestures are gestures that form a number of hand and finger movements over time. This study focused on capturing five (5) commonly used static words from the Botswana sign language that use both hands of the signer. However, there are other words in the Botswana Sign Language that are dynamic, that require the use of facial expression and that require the use of only one hand. The area of Automatic sign language recognition is broad. To build robust systems it is essential to start with one problem and building from it. Therefore, researchers begin with static gestures then moving to adding facial and dynamic gestures.

Each sign gesture was performed by different signers of different demographics to have variations. A total of ten (10) signers participated in the sign gesture capturing of which, half were males and the other half females. The signers were recruited from different parts of Palapye, Botswana to participate in the data collection. The group of individuals did not include the hearing impaired, but an interpreter was present to ensure that sign gestures are performed correctly. Before the capturing of data began, the individuals were given some time to familiarize themselves with the Botswana sign language. The individuals were given some time to practice capturing sign gestures with Microsoft Kinect and Leap motion controller. Once the individuals were familiar with operating the developed application and the devices (leap motion and Kinect), sign gesture capturing began. The Microsoft Kinect was placed a meter away from the signer and the leap motion was placed twenty (20) cm away from the signer's hands. Marin et al. [7] has revealed that Microsoft Kinect and Leap Motion Controller have different frame rates, therefore, during sign gesture capture, some few seconds were taken before the gesture is saved, so that the devices capture the current gesture. The individuals were guided by an interpreter of the Botswana Sign Language on how to execute the sign gestures. Each signer repeated the sign gesture hundred (100) times during the data acquisition process. Data needs to be repeated many times for it to work with classification algorithms [50]. The sign gestures were then categorized according to their meaning to make it easy for classification. The extracted features of a sign gesture were then saved to a Microsoft excel file. Table 2 depicts the list of Botswana Sign Language words used for this research.

Table 2 List of Botswana Sign Language static words captured in the dataset

Number	Words	Number of times to be captured
1	<i>Beautiful</i>	<i>1000</i>
2	<i>Fear</i>	<i>1000</i>
3	<i>Give</i>	<i>1000</i>
4	<i>Kiss</i>	<i>1000</i>
5	<i>Pray</i>	<i>1000</i>

3.7 Evaluation

After creating the dataset, it is important to validate it against existing sign language recognition algorithms in to determine if the dataset would assist in an automated Botswana sign language recognition system. The recognition performance of the recognition algorithms would then determine the quality of the dataset. In this section, we discuss the selected sign language recognition algorithms used to perform gesture classification and the criteria used to measure the performance of the algorithms when working on the dataset.

3.7.1 Gesture Classification

In the automatic sign language recognition research area, what is mostly different is the sign language dataset used, while the recognition algorithms remain the same. For this study, two commonly used recognition algorithms were used to evaluate the dataset: K Nearest Neighbor (KNN) and Support Vector Machines (SVM). Using Anaconda and the Python programming language libraries, the two algorithms were implemented. The Microsoft Excel file of the dataset containing Botswana sign language words was read as input. Data pre-processing was carried out on the dataset. The initial step involved division of the data into attributes and labels. All the attributes describing the features of the hand such as (finger direction, fingertip position and palm position) and (elbow positions and wrist positions) were stored in a Variable X. The meaning of the sign gesture which was stored in a column labelled “Word” which simply describes the meaning of a sign gesture. The features extracted from the two devices were merged together, those set of features describe a word or a phrase in Botswana sign language. The meaning of sign gestures was stored in a variable named Y which stores labels. The final pre-processing step was

to divide the data into training and test sets once the data was divided into attributes and labels. The training data was 60% and the test data was 40% of the dataset as is usually the norm.

3.7.1.1 Support Vector Machines

Support vector machines (SVM) is the most used classification algorithm, particularly in the field of recognition of gestures and sign language. The SVM classifier was implemented using the python programming language as depicted by Figure 14. The SVM classifier works by finding a hyperplane that separates classes and with as much margin as possible [51]. SVM algorithm is mostly used for dynamic and static hand gestures. One of the main motivations for using SVM is that it is considered to be highly accurate in high dimensional spaces. The developed dataset is highly dimensional, it has eighty-five (85) columns and five thousand (5000) rows. However, SVM does not work effectively when used in larger dataset. The selected kernel for the SVM classifier is 'linear' as our data is linearly separable. The number of rows from the dataset is five thousand four hundred and thirty-three (5433) and there are eighty-five (85) columns.

```
data.shape
(5433, 85)

X=data.drop('Word',axis=1)
y=data['Word']

from sklearn.model_selection import train_test_split
X_train, X_test,y_train,y_test=train_test_split(X,y,test_size=0.40)

from sklearn.svm import SVC
svclassifier=SVC(kernel='linear')
svclassifier.fit(X_train,y_train)
```

Figure 14 Python Code Snippet Showing the Implementation of the Support Vector Machines used to classify the Botswana sign language dataset

3.7.1.2 K -Nearest Neighbor

K-Nearest Neighbor (KNN) is a commonly used classification algorithms used in recognition of both static and dynamic sign gesture. For this research, the gestures captured are static, this means that for each of captured sign gesture there are no movements of the hands, fingers or arms. KNN classification algorithm uses Euclidean distance between an instance and class attributes as a measure of similarity. This means that when a sign gesture is performed in order to classify the

gesture, the KNN classification algorithm will find similar gestures to the one performed using the distance.

An example of how the algorithm works with respect to our dataset would be, similar sign gestures are grouped into classes. Words of the Botswana sign language represent the different classes in our dataset namely; *fear*, *pray*, *beautiful*, *kiss* and *give*. A newly captured sign gesture should belong to one of those classes. The algorithm works with the assumption that a newly captured sign gesture should be similar to those classes and uses the new sign gestures features to determine which of the classes is the new gesture a neighbor to, the closest neighbor will then be assigned as the new gestures class. We trained the algorithm by setting k (representing the number of nearest neighbors) to five (5) because there are five (5) classes. Figure 15 depicts implementation of the KNN algorithm using the Python programming language.

```
[24]: X=dataset.drop('Word',axis=1)
      y=dataset['Word']

[25]: from sklearn.model_selection import train_test_split
      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.40)

[26]: from sklearn.preprocessing import StandardScaler
      scaler = StandardScaler()
      scaler.fit(X_train)

      X_train = scaler.transform(X_train)
      X_test = scaler.transform(X_test)

[27]: from sklearn.neighbors import KNeighborsClassifier
      classifier = KNeighborsClassifier(n_neighbors=5)
      classifier.fit(X_train, y_train)

[27]: KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                           metric_params=None, n_jobs=None, n_neighbors=5, p=2,
                           weights='uniform')

[28]: y_pred = classifier.predict(X_test)
```

Figure 15 Python Code Snippet Showing the Implementation of the K Nearest Neighbor used to classify the Botswana sign language dataset

3.7.2 Performance Measure

The measure of performance for the dataset was based on the ratio of correct classification of sign gestures. Using anaconda and python programming language libraries, the created dataset is evaluated using classification algorithms, namely K Nearest Neighbor and Support Vector Machines. The python library contains a metric library which has classification report and confusion matrix. The performance details of classification models are given by the confusion matrix on a dataset where true values are known. A confusion matrix is a table that gives a summary of the prediction performance of classification models [52]. The confusion matrix gives details on how the employed classification algorithms have performed with regards to the prediction accuracy. The prediction accuracy is used to compare the results attained by other researchers to determine the quality of the data from the dataset. For automated sign language recognition systems to be robust, reliable and accurate it depends on the quality of the data collected [14]. The confusion matrix gives details about the number of correctly and incorrectly classified with count values broken down by each class. An example would be, the confusion matrix would give details of how the model was able to correctly and incorrectly predict the sign gesture of “pray” in percentage terms. The confusion matrix shows ways in which a classification model makes errors when making a prediction.

It is important to measure other aspects associated with the classification performance of a model such as recall, precision, specificity and accuracy. A confusion matrix of size $k \times k$ associated with a classifier, demonstrates the predicted actual /target classification, where k is the different number of classes.

3.7.2.1 True positive (TP)

Describes when a classification model predicts a positive and its actually true. An example would be when a model predicts that the sign gesture performed is “hello” and it is indeed the “hello” sign gesture that’s being performed.

3.7.2.2 True negative (TN)

Describes when a classification model predicts a negative and its actually true. An example would be when a model predicts that the sign gesture performed is not “hello” and it is indeed not the “hello” sign gesture that’s being performed.

3.7.2.3 False positive(**FP**)

Describes when a classification model predicts that the performed sign gesture is “*hello*” but it is definitely not “*hello*”.

3.7.2.4 False negative(**FN**)

Describes when a classification model predicts that the performed sign gesture is **not** “*hello*” but it is cc “*hello*”.

3.7.2.5 Accuracy

Accuracy of a classification model is defined as a percentage of the correctly classified training data from all the predictions made. Accuracy is one of the metrics used to evaluate the performance of a classification model and it is given by the following formula.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

3.7.2.6 Precision

Precision is a measure of exactness. This gives us an idea about when a sign gesture is “*hello*” how many times the classification model does correctly predicts that the sign gesture is definitely “*hello*”. Precision is given by the following formula.

$$Precision = \frac{TP}{TP + FP}$$

3.7.2.7 Recall

Recall is one of the metrics used to evaluate the performance of a classification model. Recall gives us an idea about how much the classification model correctly classifies positive cases of all actual positive cases. Recall is given by the following formula.

$$Recall = \frac{TP}{TP + FN}$$

4. Experiment Results and Discussion

In this chapter, we describe the created Botswana sign language dataset based on data from the Leap motion controller and Microsoft Kinect. The performance of sign language recognition algorithms on the collected dataset is evaluated.

4.1 Data Collection

We have collected a large dataset which consists of sign gestures of the Botswana Sign language. The sign gestures data was collected at the same time from the Microsoft Kinect and the Leap motion controller. The dataset consists of five (5) words from the Botswana sign language. Each of the ten (10) signers repeated the sign gesture a hundred (100) times, which means the resulting dataset should contain five thousand (5000) sign gestures. However, during the sign gesture capturing, there were some small variations resulting in some words being performed more than other words.

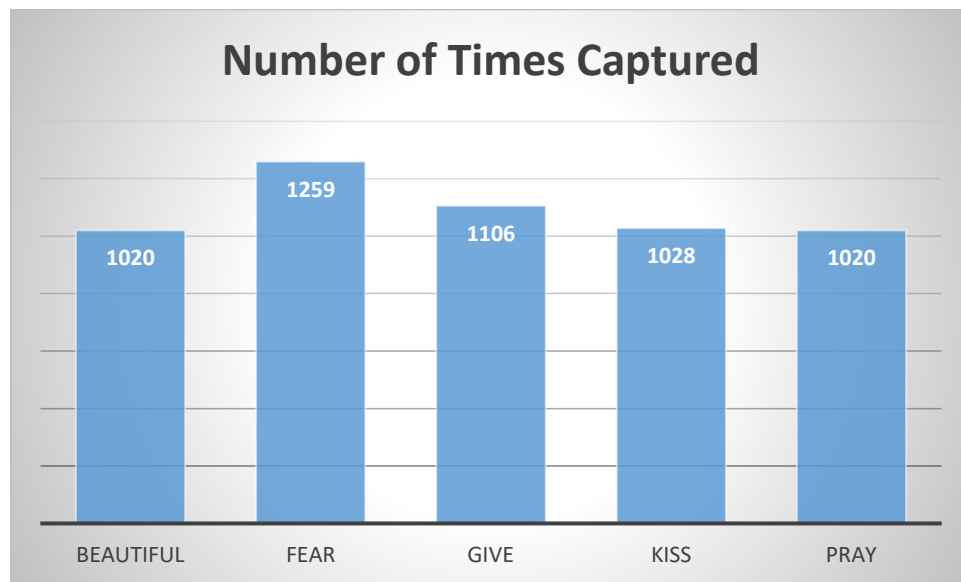


Figure 16 Chart showing the number of times each sign gestures was captured in the Botswana Sign Language dataset

All the captured sign gestures were performed using both hands. In comparison to other researches in other different sign languages where the researchers focused on creating datasets based on alphabets. Often alphabets need only one hand to perform the sign gesture and there is no major difference between alphabets of different sign languages. However, what's mostly different

between sign languages is that they have different sign gestures to convey the same meaning [2]. Figure 16 depicts the captured Botswana sign language words and the number of times each word was captured. A total number of five thousand four hundred and thirty-three (5433) gestures were recorded. The combined features of finger and arm data from the two hands made a total of eighty-five (85) attributes altogether. Table 3 shows images of the different sign gestures in the dataset and the number of times each one was captured.

Table 3 Sample of the collected sign gestures of the Botswana Sign Language

Number	Words	Sample	Number of times captured
1	<i>Beautiful</i>		1020
2	<i>Fear</i>		1259

3

Give



1106

4

Kiss



1028

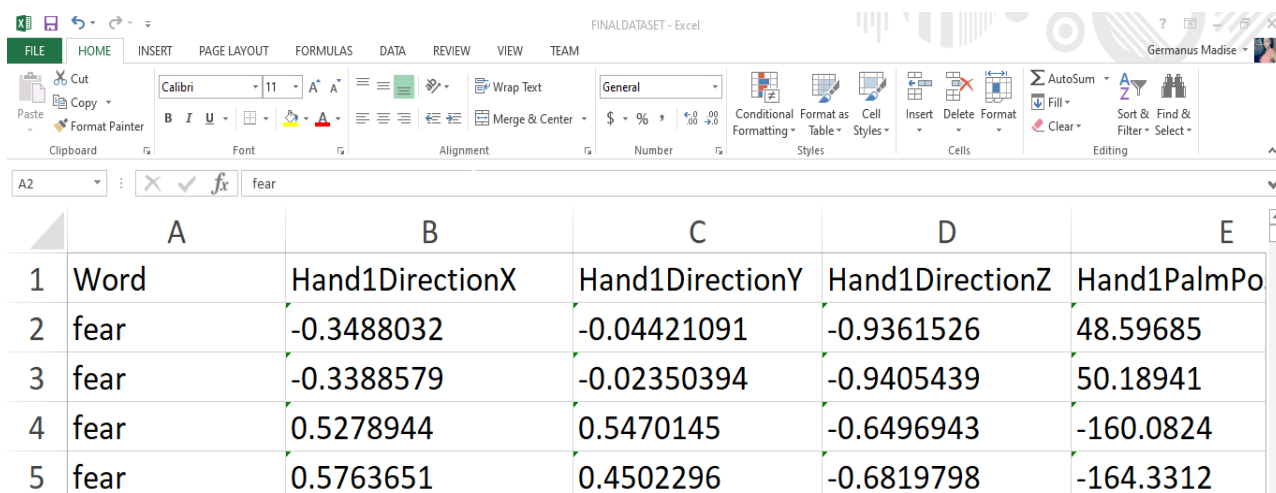
5

Pray



1020

All sign gestures captured were saved to a Microsoft Excel Comma Separated Value (CSV) file. Each column in the CSV file contains values of the concatenated hand gesture attributes. The attributes from Microsoft Kinect are 3-Dimensional data of right wrist, left wrist, right elbow and left elbow from both hands. The extracted features from the Leap motion controller are 3-Dimensional data of Finger direction, finger position and palm position from both hands. Figure 17 depicts the CSV file containing the Botswana Sign Language sign gestures. The first column of the file stores the list of different words describing a sign gesture. The next three columns depicted in Figure 17 (Hand1directionX, Hand1directionY and Hand1directionY) describes the 3-Dimensional hand direction of the left hand.



	A	B	C	D	E
1	Word	Hand1DirectionX	Hand1DirectionY	Hand1DirectionZ	Hand1PalmPo
2	fear	-0.3488032	-0.04421091	-0.9361526	48.59685
3	fear	-0.3388579	-0.02350394	-0.9405439	50.18941
4	fear	0.5278944	0.5470145	-0.6496943	-160.0824
5	fear	0.5763651	0.4502296	-0.6819798	-164.3312

Figure 17 Snapshot of Microsoft Excel CSV File Containing the Botswana Sign Language Dataset. The data is of different 3-Dimensional tracking data about the different sign gestures performed in the 3-Dimensional Space of the Leap motion controller and Microsoft Kinect

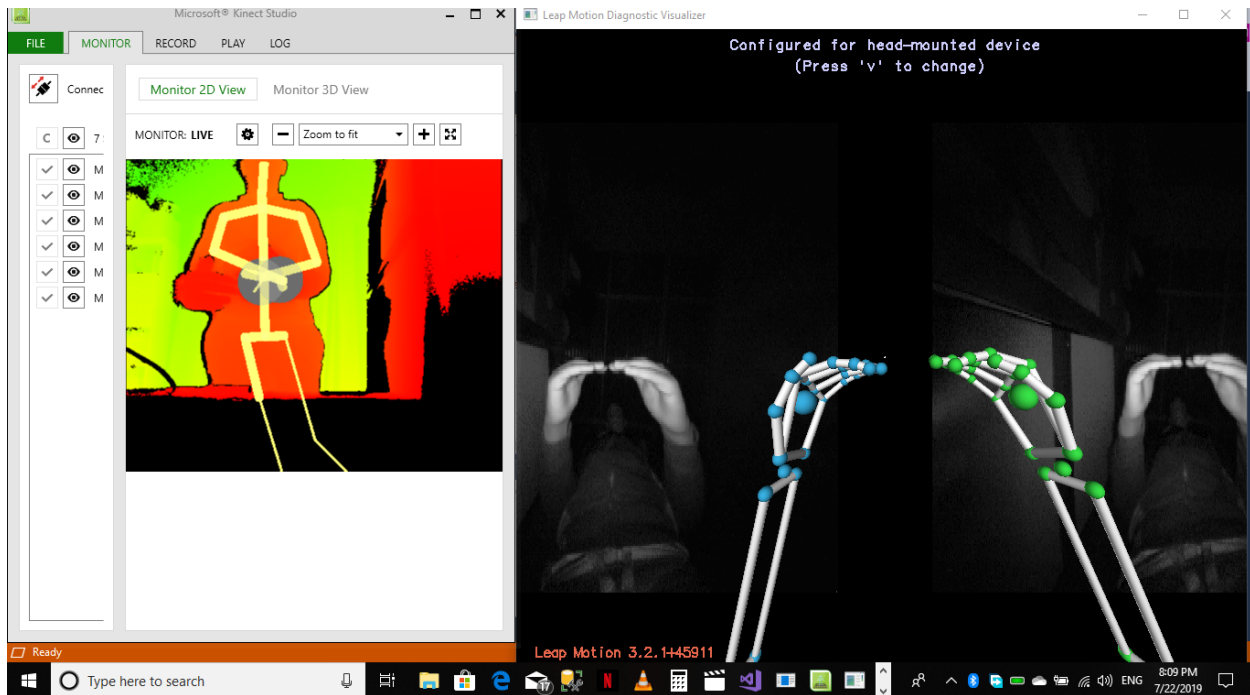


Figure 18 (Left) Image from Kinect and (Right) Leap motion controller during sign gesture capture of the word "Fear" of the Botswana Sign Language.

As depicted in Figure 18, sign gesture of the word “Fear” from the Botswana Sign Language was captured at the same time from the two sensors. As depicted in Figure 18, the image on the left from the Microsoft Kinect, the hands of the signer are not clearly visible which confirms that the Microsoft Kinect is unable to track fingers. The image from the Leap motion controller shows that the device is unable to track other body parts its only limited to the hand and fingers. An observation that was made during sign gesture capturing is that when a signer is wearing jewelry on their fingers sometimes the Leap Motion Controller mistakes the jewelry for a finger, this is supported by work done by Marin et al. [7]. Therefore, during sign gesture capturing all signers were required to remove anything from the hands that would obstruct the devices from having a good view of all the hands and fingers features. Additionally, the problem of occlusion was not fully addressed by this study as Figure 18. Figure 18 shows the field of view of the devices, in the event that the leap motion is unable to have a good view of the hand gestures the Microsoft Kinect data does not assist in getting the view of self-occluded hands or fingers.

Furthermore, the leap motion had a problem of not being able to detect all the fingers even when it is in the visible range of the device. If some fingers are not visible to the device, the Leap motion

controller programming library does not provide access to fingers or hands, this includes either self-occluded fingers or hands that do not appear during sign gesture capture. According to Marin et al. [18], failure to detect all the fingers leads to misclassification and poor recognition accuracy of machine learning algorithms. Before a sign gesture was saved, all the fingers needed to be visible to avoid cases of misclassification, for that reason it required some time to capture the sign gesture. The laptop used proved to be one of the factors that made it difficult to capture sign gestures at a faster rate. During sign gesture capture, the laptop was not time efficient when it came to capturing tracking data, then combining the features and finally saving the features to a file. The whole process made the participants tired, because they had to pose for a long time just to capture one gesture. To make it easy for them they took some breaks before capturing other gestures.

4.2 Gesture Classification

This section discusses the evaluation of performance of different classification models using the developed Botswana Sign Language dataset. Different classification algorithms were employed in order to determine if the developed dataset would be able to assist in, for example, automatically translating Botswana sign language to text.

4.2.1 Support Vector Machines

For this study, the Support Vector Machines (SVM) algorithm was implemented using python programming language. Depicted by Figure 19 is the results obtained when implementing the algorithm.

```

svclassifier.fit(X_train,y_train)

Out[67]: SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto deprecated',
  kernel='linear', max_iter=-1, probability=False, random_state=None,
  shrinking=True, tol=0.001, verbose=False)

In [68]: y_pred=svclassifier.predict(X_test)

In [69]: from sklearn.metrics import classification_report,confusion_matrix, accuracy_score
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
print(accuracy_score(y_test,y_pred))

[[411  0  0  0  0]
 [ 0 500  0  1  0]
 [ 0  0 464  0  1]
 [ 0  0  0 408  0]
 [ 0  0  0  0 389]]
      precision    recall  f1-score   support

 beautiful      1.00      1.00      1.00        411
      fear      1.00      1.00      1.00        501
       give      1.00      1.00      1.00        465
       kiss      1.00      1.00      1.00        408
       pray      1.00      1.00      1.00        389

 micro avg      1.00      1.00      1.00       2174
 macro avg      1.00      1.00      1.00       2174
 weighted avg      1.00      1.00      1.00       2174

0.9990800367985281

In [70]: 0.9986200551977921

```

Figure 19 Code Snippet of Implementation of Support Vector Machines

The results obtained from successful implementation of the SVM classification model show that the overall model accuracy obtained is **99.90%** using the developed Botswana sign language dataset based on concatenated features from the Leap Motion and Microsoft Kinect. The confusion matrix in Table 4 gives a detailed report on the performance of the SVM. Furthermore, Figure 20 gives a detailed report on the precision and recall of the SVM classification model.

Table 4 Confusion Matrix of Support Vector Machine classifier on the collected dataset

Confusion Matrix	beautiful	fear	give	kiss	pray
beautiful	411	0	0	0	0
fear	0	500	0	1	0
give	0	0	464	0	1
kiss	0	0	0	408	0
pray	0	0	0	0	389

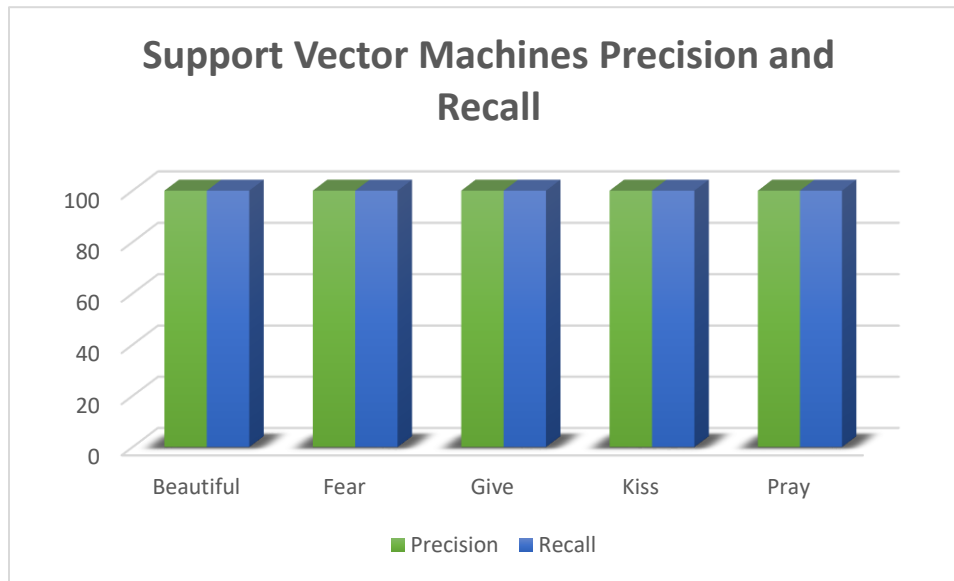


Figure 20 Support Vector Machines Precision and Recall report

The results show that the overall accuracy of the model was **99.90%** using the developed Botswana sign language dataset. The results of Table 4 show that, of the four hundred and eleven (411) sign gestures for the word *beautiful* all of them were correctly classified. From the results it is clear that the SVM classifier's precision is one hundred (100) percent because the model is able to correctly classify all the sign gesture for the word correctly. The recall for the word beautiful is one hundred (100) percent, it shows that the model is able to correctly reject sign gestures which are not of the word beautiful. From the results, we learn that, of the five hundred and one (501) instances of the sign gesture *fear* only one (1) was incorrectly classified as *kiss*. The results also show that of the four hundred and sixty-five (465) instances of the sign gesture *give* only one (1) was incorrectly classified as *pray*. We also learn that of the four hundred and eight (408) instances of the sign gesture *kiss*, all of them were correctly classified. All of the sign gestures for the word *pray* were correctly classified. A detailed performance report of the SVM classification model on each word of the Botswana Sign Language is given in Figure 21.

One of the reasons that can explain why some gestures are mistaken for others is that the sign gestures are similar or close. The leap motion can also be a contributing factor to misclassification of sign gestures because during data capture it misrepresented the sign gestures. Alexander [53]

pointed out that the leap motion controller sometimes struggles to represent the hand accurately, especially when two fingers are too close to each other or when some fingers are obscured .

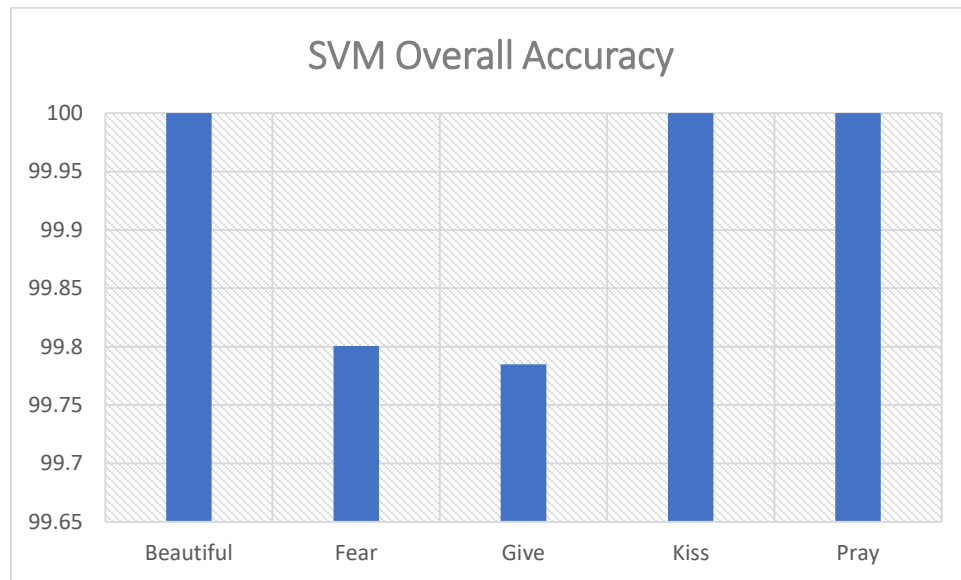


Figure 21 overall Performance of the Support Vector Machine on each word from The Botswana Sign Language

4.2.2 K Nearest Neighbor

For this study, the K Nearest Neighbor (KNN) algorithm was implemented using python programming language. Depicted by Figure 22 is the results obtained when implementing the algorithm.

```

: from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
  print(confusion_matrix(y_test, y_pred))
  print(classification_report(y_test, y_pred))
  print(accuracy_score(y_test, y_pred))

```

```

[[404  2  0  1  0]
 [ 0 524  0  0  3]
 [ 0  0 444  1  3]
 [ 0  0  0 411  0]
 [ 0  2  1  0 378]]

```

	precision	recall	f1-score	support
beautiful	1.00	0.99	1.00	407
fear	0.99	0.99	0.99	527
give	1.00	0.99	0.99	448
kiss	1.00	1.00	1.00	411
pray	0.98	0.99	0.99	381
micro avg	0.99	0.99	0.99	2174
macro avg	0.99	0.99	0.99	2174
weighted avg	0.99	0.99	0.99	2174

```

0.9940202391904324

```

Figure 22 Code Snippet of Implementation of KNN

The results obtained from successful implementation of the KNN classification model show that the overall model accuracy obtained is **99.40%** using the developed Botswana sign language dataset based on concatenated features from the Leap Motion and Microsoft Kinect. The confusion matrix in Table 5 gives a detailed report on the performance of the KNN. Furthermore, Figure 23 gives a detailed report on the precision and recall of the KNN classification model.

Confusion Matrix	beautiful	fear	give	kiss	pray
beautiful	404	2	0	1	0
fear	0	524	0	0	3
give	0	0	444	1	3
kiss	0	0	0	411	0
pray	0	2	1	0	378

Table 5 Confusion Matrix of K Nearest Neighbor classifier on the collected dataset

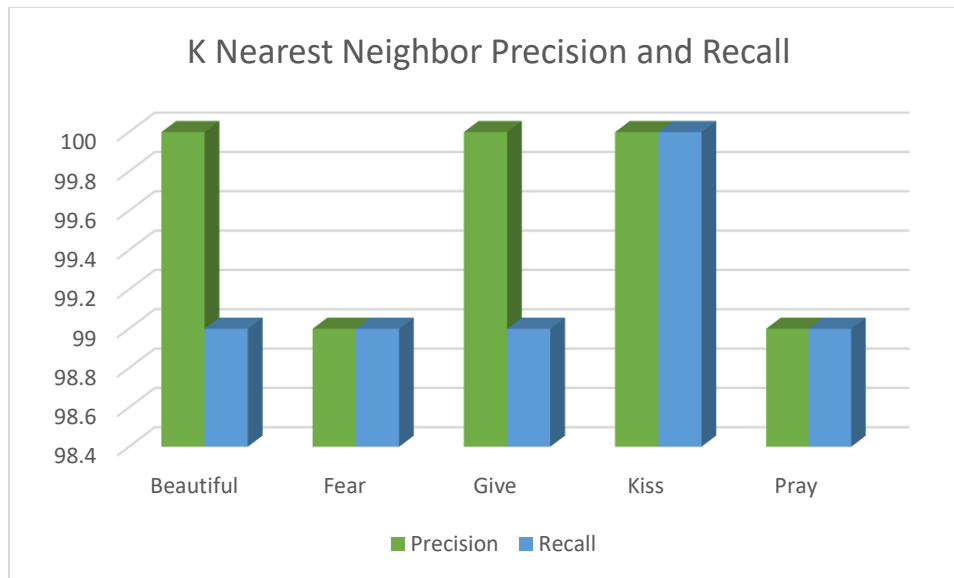


Figure 23 KNN Precision and Recall report

The results indicate that the KNN classification algorithm managed to achieve an overall accuracy of **99.40%**. The dataset has five thousand four hundred and thirty-three (5433) records of which 60% is used for training and 40% is used for testing, resulting in two thousand one hundred and seventy-four (2174) samples. The results show that of the four hundred and seven (407) instances of the sign gesture *beautiful*, only four hundred and four (404) were correctly classified. In this regard, two (2) of the instances were misclassified as the sign gesture *fear* and the other instance was misclassified as the sign gesture *kiss*. This work shows that, the KNN algorithm was able to correctly classify five hundred and twenty-four (524) of five hundred and twenty-seven (527) instances of the sign gesture *fear*. However, the results also show that three (3) of the instances of the sign gesture *fear* were incorrectly classified as *pray*. From the results we learn that of the four hundred and forty-eight (448) instances of the sign gesture *give* only four hundred and four (444) instances were correctly classified. The three (3) other instances of the sign gesture were incorrectly classified as *pray* and one (1) instance was incorrectly classified as the sign gesture *kiss*. For the sign gesture *kiss* the algorithm showed better results, as for all four hundred and eleven (411) instances were correctly classified. Only three hundred and seventy-eight (378) of three hundred and eighty-one (381) instances of the sign gesture were correctly classified, two (2) of the instances were incorrectly classified as *fear* while one (1) instance was incorrectly classified as *give*. A detailed performance report of the KNN classification model on each word of the Botswana Sign Language is given in Figure 24.

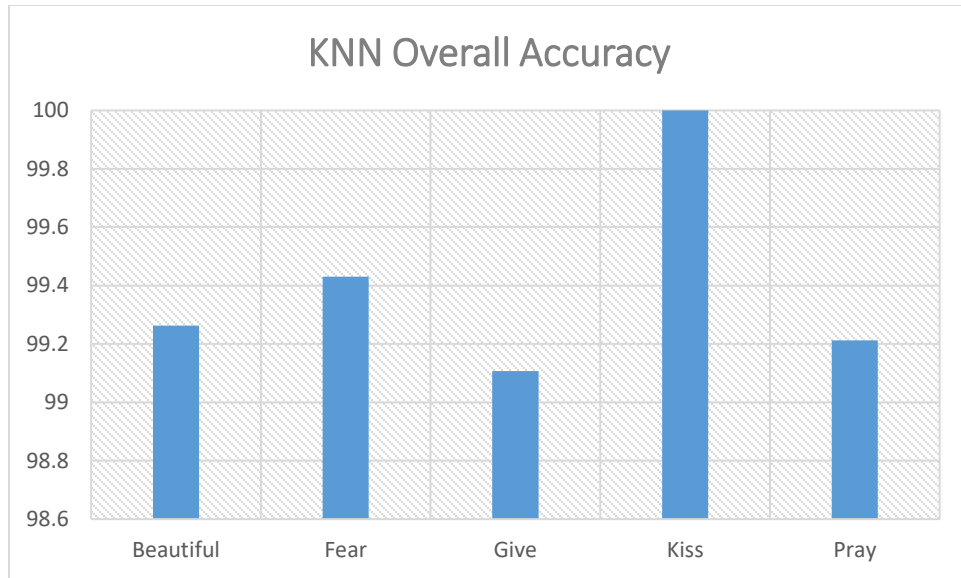


Figure 24 overall Performance of the Support Vector Machine on each word from The Botswana Sign Language

4.3 Comparative Analysis

In this section we discuss the performances of the two classification models on our developed Botswana sign language dataset as compared to each other. We also consider this results in the view of related work as a way of gauging the quality of the dataset here proposed.

4.3.1 Self-Comparative Analysis of the SVM and the KNN within this study

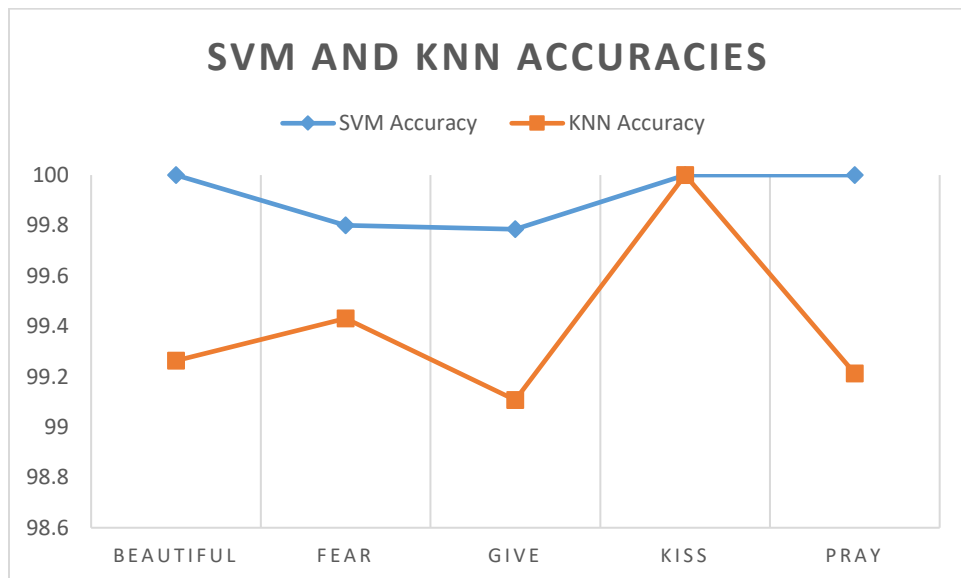


Figure 25 Accuracies achieved by SVM and KNN in classification of the collected Botswana Sign Language dataset

With regards to the sign gesture *beautiful*, the Support Vector Machine was able to correctly classify all instances and was able to reject any other which is not of the sign gesture. However, results from the K Nearest Neighbor classifier show that two (2) were classified as *kiss* and one (1) was classified as *fear* which shows that Support vector outperformed the K Nearest Neighbor when classifying the sign gesture *beautiful*. Support Vector Machine Classifier was able to recognize five hundred (500) instances of the word *fear* and only one (1) was incorrectly classified as *kiss*. The K Nearest Neighbor was again outperformed by Support Vector Machines with five hundred and twenty-four (524) correctly classified sign gestures and only three (3) were incorrectly classified.

The results show that the Support Vector Machine classifier was able to recognize four hundred and sixty-four (464) instances correctly and one (1) was incorrectly classified as *kiss*. The K Nearest Neighbor classifier was again outperformed by the Support Vector Machines with four hundred forty-four (444) correctly classified instances and one (1) classified as *kiss* and three (3) were classified as *pray*. Both classifiers incorrectly classified *give* as *pray*, this might be because of the similarity of the sign gestures. Also, one of the contributing factors could be the fact that the sign gestures were performed by non-skilled individuals which leads to incorrect representation of sign gestures. Incorrect representation of sign gestures leads to errors in classification.

Additionally, the results from both classifiers correctly classified all instances of the sign gesture for *kiss*. The Support Vector Machines Classifier correctly classified all three hundred and eighty-nine (389) instances of the word *pray*. However, the K Nearest Neighbor correctly classified three hundred and seventy-eight (378) instances while two (2) were classified as *fear* and the other one (1) was classified as *give*. From this study we learn that in the overall performance of classification of our developed dataset based on tracking data of the Microsoft Kinect and Leap motion controller, Support Vector Machines performed better than K Nearest Neighbor in the accuracy rate. The K Nearest Neighbor made more errors in the classification of each Botswana sign language words.

We can therefore deduce that the difference in performance of the algorithms on the Botswana Sign Language Dataset is negligible because it is within the one percentile.

4.3.2 Comparative Analysis of the Results to Related Works

This research work is compared with other related research work in different sign languages for purposes of estimating the quality of the dataset and establishing a yardstick for further development of the research. There are different data acquisition approaches adopted by researchers to tackle the problem of automatically translating gestures to text. Results may vary depending on the classification model used, size of dataset and sign language used. This is to mean that work done on other sign languages may not be directly applicable to another sign language. This is because sign language is different from country to country and it is never an international language. Therefore, a comparative analysis of sign language is for the most part subjective as it may be peculiar to individual research. Nevertheless, this section puts this research in context of global research on sign language as it compares our results with a sample of related research work.

Naidu et al. [54], proposed hand gesture recognition of the Indian sign language using the Leap motion controller. Forty-five (45) overall gestures were captured, of which twenty-six (26) were alphabets, nine (9) words and ten (10) numbers of the Indian Sign language. The maximum recognition accuracy that was achieved was 90%. From the research conducted the authors were only focused on capturing hand poses and failed to include other body parts. However, Naidu et al [54] pointed out that to make a robust automated sign language recognition system body parts like elbows, arms, facial features and lip movement should be considered. In that regard, our research's results show a better recognition accuracy when compared to Naidu et al [54] results, more that other body parts were considered. Moreover, our study used two sensors for data acquisition while they used a single sensor for their data acquisition. In line with the ideas from Kumar et al. [2], that recognition accuracy can be improved when Microsoft Kinect is used together with the leap motion controller for data acquisition, it can be concluded to be true because our research achieved an average of 99% accuracy compared to their 90% recognition accuracy.

Marin et al. [7], presented a sign gesture recognition model based on tracking data of the Microsoft Kinect and Leap motion controller. The two devices captured American Sign Language (ASL) alphabets. The features captured from the two devices were combined and fed into a Support vector Machine (SVM) classifier for recognition of the sign gestures. The SVM classifier achieved an overall accuracy 91.28%. Contrary to their research, our approach was to access tracking data from the Microsoft Kinect, they instead accessed the raw image from the device and used machine learning algorithms to extract hand and fingers from the image. The process requires complex

computations and is often affected by background environment. The ideas by Kumar et al. [4], suggested that using tracking data from the devices is more accurate than accessing raw images, this proved to be true as this study achieved better results using the same SVM classifier. Their research focused on finger spelling. However, Kumar et al. [5] believes that systems that automatically translate sign gestures to spoken language cannot be built solely on finger spelling and alphabets. Our research was focused on static words of the Botswana sign language which require both hands and more complex finger poses as opposed to alphabets that require the use of a single hand to perform.

Chong et al. [49], focused at developing a complete recognition for American Sign Language (ASL) which consist of ten (10) digits and twenty six (26) letters. Majority of the ASL letters in their dataset are static and some are dynamic. This research results, revealed that the sign language recognition rates for the twenty-six (26) letters using a Deep Neural Network (DNN) and Support Vector Machine (SVM) achieved an overall recognition rates of 93.81% and 80.30%, respectively. Their research was better in the sense that they combined both static and dynamic sign gestures, however, our study proved to be better at recognition with an average recognition rate of 99% using support vector machines classifier.

5. Conclusion and future works

5.1 Introduction

In this study we address the lack of research work on automatic sign language recognition of Botswana sign language which can be attributed to the lack of sign language dataset. This study addressed this problem by developing a Botswana sign language dataset based on the tracking data derived from the Microsoft Kinect and Leap motion controller. The use of two sensors was motivated by research work which revealed that robust automatic recognition systems are built using more than a single sensor. The benefit of using two sensors is to improve the accuracy of extracting finger and hand poses accurately. During the sign acquisition, tracking data was captured simultaneously from the two sensors. The collected dataset consisted of five thousand four hundred and thirty-three (5433) rows and eighty-five (85) columns, only five (5) static words of Botswana sign language were used for the dataset. In order to evaluate the collected dataset, we used existing classification algorithms to perform recognition. The algorithms used are Support Vector Machines and K Nearest Neighbor they achieved an overall accuracy of 99.90% and 99.40% respectively. By combining tracking data from the Microsoft Kinect and Leap motion controller we achieved better recognition accuracy than when using a single sensor. In future the dataset will be extended to include dynamic gestures and including facial expressions. Although, high recognition rates are achieved there are a number of limitations which are discussed in sub section 5.2.

5.2 Limitation of our work

It is important to note that sign language is not a universal language as it is different from country to country nor is it a visual form of spoken language. The main aim of this study was to develop a Botswana sign language dataset based on data from two sensors being the Leap motion controller and Microsoft Kinect. The motivation of developing such a dataset was due to the fact that, to the best of our knowledge research in automatic sign language recognition of Botswana sign language is lacking and it can be attributed to the lack of sign language dataset. Such a dataset is required in order to bridge the communication gap between the hearing and the hearing-impaired Batswana by assisting in translating sign gestures to text. The results from previous related research showed

that using more than one sensor improved the recognition accuracy hence the use of the two sensors.

The chosen words (*beautiful, pray, fear, give, kiss*) of the Botswana sign language are in English because their sign gestures are represented in English words. All the chosen words for this research are all static gestures, this means that no movements are made when the sign gestures are performed. However, not all the words were included for this research the Botswana sign language has other words that require movements (dynamic gestures) of the hands, fingers, lips, arms and facial expression. One of the major limitations of this study is the lack of dynamic gestures in the developed Botswana sign language dataset and this renders the developed dataset incomplete. In real-life situations individuals use both static and dynamic gestures to communicate. Although, the developed dataset is rendered incomplete, most of related studies focused on incomplete dataset because there is a lot more research that needs to be conducted in the area of automatic sign recognition of sign language. The dataset has achieved positive results when compared to existing sign language recognition algorithms which were used to classify the data with an average of 99% accuracy using SVM and KNN classifier.

One of the contributions of this research is the set of features combined for the developed dataset, however, most of the sign gestures for Botswana sign language require the movements of elbows, wrist, fingers, hands and the use of facial expressions. Some sign gestures have similar finger and hand poses but the only thing that differentiates the gesture is the facial expressions. Not including the facial expression as part of the features is one other limitation of this research.

Another limitation of this study is that it failed to fully address the issue of occlusion. Occlusion occurs when all or some fingers or hands are hidden from the view of the camera. According to Kumar et al. [4] self-occlusion is one of the major factors that lead to misclassification of sign gestures. [2], [4], [5], [18] many researchers have suggested the use of more than one sensor so that when one sensor fails to accurately capture a sign gesture the other one might have a good view of the gesture. In this study, two sensors were used to capture different sets of features. The Microsoft Kinect was used to capture position of the elbows and wrist while the leap motion controller was used to capture the position of the palms, hands and fingers. If the Microsoft Kinect could not deliver a good view of the wrists and elbows the leap motion controller could not assist because the leap motion is limited to tracking fingers and hands only. Furthermore, if the Leap

motion could not have a good view of the hands and fingers the Microsoft Kinect could not assist because it is not able to track small body parts. However, Kumar et al. [2] accessed the raw image from the Microsoft Kinect and used machine learning algorithms to extract the hand and fingers. Using the raw image from the Microsoft Kinect is not as accurate as using the tracking data from the device [7]. For this study, tracking data derived from the two devices was used.

One of the challenges which were experienced during the acquisition of the sign gesture was that the Leap motion software struggled to represent the hand accurately especially when the fingers and hands are too close to each other. The Microsoft Kinect also had a challenge of accurately representing the arm position during data collection. The other challenge is that the laptop that was used kept freezing when capturing the sign gestures which proved to be one of the factors that slowed down the progress of the research.

5.3 Future work

In this study better recognition accuracy rate was achieved compared to existing studies, however, there are several issues to be considered in the future. This study did not fully address the problem of occlusion, in the future two Leap motion controllers will be used to simultaneously capture hands and fingers at two different viewing angles, if one device fails to accurately capture sign gesture data from the other device will be used. Furthermore, two Microsoft Kinect will be used to capture arm movements at different viewing angles. Another future work would be expanding the dataset to include lip and facial expressions features. Additionally, the dataset will be extended to include dynamic sign gestures and including sentences.

References

- [1] F. Ronchetti, F. Quiroga, and L. Lanzarini, "LSA64 : An Argentinian Sign Language Dataset," *Congr. Argentino Ciencias la Comput.*, pp. 794–803, 2016.
- [2] P. Kumar, H. Gauba, P. Pratim Roy, and D. Prosad Dogra, "A multimodal framework for sensor based sign language recognition," *Neurocomputing*, vol. 259, pp. 21–38, 2016.
- [3] G. A. Rao and P. V. V Kishore, "Selfie video based continuous Indian sign language recognition system," *Ain Shams Eng. J.*, p. , 2017.
- [4] P. Kumar, H. Gauba, P. P. Roy, and D. P. Dogra, "Coupled HMM-based multi-sensor data fusion for sign language recognition," *Pattern Recognit. Lett.*, vol. 86, pp. 1–8, 2017.
- [5] P. Kumar, P. P. Roy, and D. P. Dogra, "Independent Bayesian classifier combination based sign language recognition using facial expression," *Inf. Sci. (Ny).*, vol. 428, pp. 30–48, 2018.
- [6] D. Ryumin, D. Ivanko, A. Axyonov, I. Kagiroy, A. Karpov, and M. Zelezny, "Human Robot Interaction with Smart Shopping Trolley Using Sign Language: Data Collection," *2019 IEEE Int. Conf. Pervasive Comput. Commun. Work. PerCom Work. 2019*, pp. 949–954, 2019.
- [7] G. Marin, F. Dominio, and P. Zanuttigh, "Hand gesture recognition with leap motion and kinect devices," *2014 IEEE Int. Conf. Image Process. ICIP 2014*, pp. 1565–1569, 2014.
- [8] M. R. Islam, U. K. Mitu, R. A. Bhuiyan, and J. Shin, "Hand gesture feature extraction using deep convolutional neural network for recognizing American sign language," *2018 4th Int. Conf. Front. Signal Process. ICFSP 2018*, pp. 115–119, 2018.
- [9] C.-H. Chuan, E. Regina, and C. Guardino, "American Sign Language Recognition Using Leap Motion Sensor," *2014 13th Int. Conf. Mach. Learn. Appl.*, no. December, pp. 541–544, 2014.
- [10] D. Naglot and M. Kulkarni, "Real time sign language recognition using the Leap Motion Controller," *Proc. Int. Conf. Inven. Comput. Technol. ICICT 2016*, vol. 2016, pp. 1–5, 2016.
- [11] C. Chansri and J. Srinonchat, "Hand Gesture Recognition for Thai Sign Language in Complex Background Using Fusion of Depth and Color Video," *Procedia Comput. Sci.*, vol. 86, no. March, pp. 257–260, 2016.
- [12] S. P. Das, A. K. Talukdar, and K. K. Sarma, "Sign Language Recognition Using Facial Expression," *Procedia Comput. Sci.*, vol. 58, pp. 210–216, 2015.
- [13] S. Rajaganapathy, B. Aravind, B. Keerthana, and M. Sivagami, "Conversation of sign language to speech with human gestures," *Procedia Comput. Sci.*, vol. 50, pp. 10–15, 2015.
- [14] M. Simão, P. Neto, and O. Gibaru, "Using data dimensionality reduction for recognition of incomplete dynamic gestures," *Pattern Recognit. Lett.*, vol. 0, pp. 1–7, 2017.
- [15] K. M. Lim, A. W. C. Tan, and S. C. Tan, "Block-based histogram of optical flow for isolated sign language recognition," *J. Vis. Commun. Image Represent.*, vol. 40, no. Part B, pp. 538–545, 2016.
- [16] J. Forster, O. Koller, C. Oberdörfer, Y. Gweth, and H. Ney, "Improving continuous sign language recognition: Speech recognition techniques and system design," *Work. Speech ...*, pp. 41–46, 2013. "Finger detection for sign language recognition," *Int. MultiConference Eng. Comput. Sci.*, vol. I, pp. 0–4, 2009.

- [17] J. Ravikiran, K. Mahesh, S. Mahishi, R. Dheeraj, S. Sudheender, and N. V Pujari, "Finger detection for sign language recognition," *Int. MultiConference Eng. Comput. Sci.*, vol. I, pp. 0–4, 2009.
- [18] C. P. Vogler, "American Sign Language Recognition: Reducing the Complexity of the Task with Phoneme-Based Modeling and Parallel Hidden Markov Models," *Diss. Abstr. Int. B Sci. Eng.*, vol. 64, no. 4, 2003.
- [19] V. Athitsos *et al.*, "The American Sign Language Lexicon Video Dataset," 2008.
- [20] H. Chandel and S. Vatta, "Occlusion Detection and Handling: A Review," *Int. J. Comput. Appl.*, vol. 120, no. 10, pp. 975–8887, 2015.
- [21] H.-D. Yang, "Sign Language Recognition with the Kinect Sensor Based on Conditional Random Fields," *Sensors*, vol. 15, no. 1, pp. 135–147, 2014.
- [22] C. Oz and M. C. Leu, "American Sign Language word recognition with a sensory glove using artificial neural networks," *Eng. Appl. Artif. Intell.*, vol. 24, no. 7, pp. 1204–1213, 2011.
- [23] T. Shanableh and K. Assaleh, "User-independent recognition of Arabic sign language for facilitating communication with the deaf community," *Digit. Signal Process. A Rev. J.*, vol. 21, no. 4, pp. 535–542, 2011.
- [24] S. M. Shohieb, H. K. Elminir, and A. M. Riad, "Signs World Atlas; a benchmark Arabic Sign Language database," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 27, no. 1, pp. 68–76, 2015.
- [25] F. Gustin, I. Rendulic, N. Miskovic, and Z. Vukic, "Hand gesture recognition from multibeam sonar imagery," *IFAC-PapersOnLine*, vol. 49, no. 23, pp. 470–475, 2016.
- [26] K. Iwasako, M. Soga, and H. Taki, "Development of finger motion skill learning support system based on data gloves," *Procedia Comput. Sci.*, vol. 35, no. C, pp. 1307–1314, 2014.
- [27] A. Z. Shukor, M. F. Miskon, M. H. Jamaluddin, F. Bin Ali Ibrahim, M. F. Asyraf, and M. B. Bin Bahar, "A New Data Glove Approach for Malaysian Sign Language Detection," *Procedia Comput. Sci.*, vol. 76, no. Iris, pp. 60–67, 2015.
- [28] F. Pezzuoli, D. Corona, M. L. Corradini, and A. Cristofaro, "Development of a Wearable Device for Sign Language Translation," pp. 115–126, 2018.
- [29] J. E. López-Noriega, M. I. Fernández-Valladares, and V. Uc-Cetina, "Glove-Based sign language recognition solution to assist communication for deaf users," *2014 11th Int. Conf. Electr. Eng. Comput. Sci. Autom. Control. CCE 2014*, 2014.
- [30] A. S. Elons, M. Ahmed, H. Shedid, and M. F. Tolba, "Arabic sign language recognition using leap motion sensor," *Proc. 2014 9th IEEE Int. Conf. Comput. Eng. Syst. ICCES 2014*, pp. 368–373, 2014.
- [31] Y. Lan, J. Li, and Z. Ju, "Data fusion-based real-time hand gesture recognition with Kinect V2," *Proc. - 2016 9th Int. Conf. Hum. Syst. Interact. HSI 2016*, pp. 307–310, 2016.
- [32] M. Ahmed, M. Idrees, Z. Abideen, R. Mumtaz, and S. Khalique, "Deaf Talk Using 3D Animated Sign Language," *2016 SAI Comput. Conf.*, pp. 330–335, 2016.
- [33] S. Aliyu, M. Mohandes, M. Deriche, and S. Badran, "Arabie sign language recognition using the Microsoft Kinect," *13th Int. Multi-Conference Syst. Signals Devices, SSD 2016*, pp. 301–306, 2016.
- [34] K. Lai, J. Konrad, and P. Ishwar, "A gesture-driven computer interface using Kinect," *Proc. IEEE Southwest Symp. Image Anal. Interpret.*, pp. 185–188, 2012.
- [35] C. Sun, T. Zhang, B. K. Bao, and C. Xu, "Latent support vector machine for sign language recognition with Kinect," *2013 IEEE Int. Conf. Image Process. ICIP 2013*

- [36] F. Dominio, M. Donadeo, and P. Zanuttigh, "Combining multiple depth-based descriptors for hand gesture recognition," *Pattern Recognit. Lett.*, vol. 50, pp. 101–111, 2014.
- [37] Sulfayanti, Dewiani, and A. Lawi, "A real time alphabets sign language recognition system using hands tracking," *Proc. - Cybern. 2016 Int. Conf. Comput. Intell. Cybern.*, pp. 69–72, 2017.
- [38] S. Bessa Carneiro, E. D. F. M. De Santos, T. M. A. De Barbosa, J. O. Ferreira, S. G. S. Alcalá, and A. F. Da Rocha, "Static gestures recognition for Brazilian Sign Language with kinect sensor," *Proc. IEEE Sensors*, 2017.
- [39] E. Pramunanto, S. Sumpeno, and R. S. Legowo, "Classification of hand gesture in Indonesian sign language system using naive bayes," *Proc. - 2017 Int. Semin. Sensor, Instrumentation, Meas. Metrol. Innov. Adv. Compet. Nation, ISSIMM 2017*, vol. 2017-Janua, pp. 184–191, 2017.
- [40] M. W. Cohen, N. Ben Zikri, and A. Velkovich, "Recognition of continuous sign language alphabet using leap motion controller," *Proc. - 2018 11th Int. Conf. Hum. Syst. Interact. HSI 2018*, pp. 193–199, 2018.
- [41] L. E. Potter, J. Araullo, and L. Carter, "The Leap Motion controller: A view on sign language," *Proc. Aust. Comput. Interact. Conf. Augment. Appl. Innov. Collab. - Oz-CHI '13*, no. February 2016, pp. 175–178, 2013.
- [42] C.-H. Chuan, E. Regina, and C. Guardino, "American Sign Language Recognition Using Leap Motion Sensor," *2014 13th Int. Conf. Mach. Learn. Appl.*, pp. 541–544, 2014.
- [43] P. Kumar, R. Saini, S. K. Behera, D. P. Dogra, and P. P. Roy, "Real-time recognition of sign language gestures and air-writing using leap motion," *Proc. 15th IAPR Int. Conf. Mach. Vis. Appl. MVA 2017*, vol. 1, pp. 157–160, 2017.
- [44] L. Shao, G. Wetzstein, and R. Konrad, "Hand movement and gesture recognition using Leap Motion Controller Stanford EE 267, Virtual Reality, Course Report, Instructors," pp. 2–6, 2015.
- [45] M. Deriche, S. Aliyu, and M. Mohandes, "An Intelligent Arabic Sign Language Recognition System using a Pair of LMCs with GMM Based Classification," *IEEE Sens. J.*, vol. 19, no. 18, pp. 1–12, 2019.
- [46] M. Mohandes, S. Aliyu, and M. Deriche, "Prototype Arabic Sign language recognition using multi-sensor data fusion of two leap motion controllers," *12th Int. Multi-Conference Syst. Signals Devices, SSD 2015*, pp. 1–6, 2015.
- [47] M. Sreejith, S. Rakesh, S. Gupta, S. Biswas, and P. P. Das, "Real-time hands-free immersive image navigation system using Microsoft Kinect 2.0 and Leap Motion Controller," *2015 5th Natl. Conf. Comput. Vision, Pattern Recognition, Image Process. Graph. NCVPRIPG 2015*, vol. 0, no. Figure 1, pp. 1–4, 2016.
- [48] V. Kiselev, M. Khlamov, and K. Chuvilin, "Hand Gesture Recognition with Multiple Leap Motion Devices," *2019 24th Conf. Open Innov. Assoc.*, pp. 163–169, 2019.
- [49] T. W. Chong and B. G. Lee, "American sign language recognition using leap motion controller with machine learning approach," *Sensors (Switzerland)*, vol. 18, no. 10, 2018.
- [50] P. Haque, B. Das, and N. N. Kaspy, "Two-Handed Bangla Sign Language Recognition Using Principal Component Analysis (PCA) and KNN Algorithm," *2nd Int. Conf. Electr. Comput. Commun. Eng. ECCE 2019*, pp. 1–4, 2019.

- [51] A. Borysova, “Leap Motion Controller for sign language recognition : A review of the literature.”
- [52] Khamid, A. D. Wibawa, and S. Sumpeno, “Gesture recognition for Indonesian Sign Language Systems (ISLS) using multimodal sensor leap motion and myo armband controllers based-on naïve bayes classifier,” *Proc. - 2017 Int. Conf. Soft Comput. Intell. Syst. Inf. Technol. Build. Intell. Through IOT Big Data, ICSIIT 2017*, vol. 2018-Janua, pp. 1–6, 2018.
- [53] L. Alexander, “Sign Language Recognition using Leap Motion,” no. March, pp. 263–269, 2016.
- [54] C. Naidu and A. Ghotkar, “Hand gesture recognition using Leap Motion Controller,” *Ijsr*, vol. 5, no. 10, pp. 436–441, 2016.