CORRELATION-BASED SUBSET EVALUATION OF FEATURE SELECTION FOR DYNAMIC MALAYSIAN SIGN LANGUAGE

SUTARMAN

DOCTOR OF PHILOSOPHY

(COMPUTER SCIENCE)

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's full name : SUTARMAN

Date of birth : 27rd January 1972

Title : CORRELATION-BASED SUBSET EVALUATION OF FEATURE SELECTION FOR DYNAMIC MALAYSIAN SIGN LANGUAGE

Academic Session : Semester 2 2015/2016

I declare that this thesis is classified as:

☐	**CONFIDENTIAL**	(Contains confidential information under the Official Secret Act 1972)*
☐	**RESTRICTED**	(Contains restricted information as specified by the organization where research was done)*
☐	**OPEN ACCESS**	I agree that my thesis to be published as online open access (Full text)

I acknowledge that Universiti Malaysia Pahang reserve the right as follows:

1. The Thesis is the Property of University Malaysia Pahang
2. The Library of University Malaysia Pahang has the right to make copies for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified By:

Sutarman

Passport Number: A2020571

Date:

Assoc. Prof. Dr. Mazlina Abdul Majid

Date:

SUPERVISOR'S DECLARATION

We hereby declare that we have checked this thesis and in our opinion, this thesis is adequate in terms of scope and quality of the award of the degree of Doctor of Philosophy in Computer Science.

Signature

Name of Supervisor : Assoc. Prof. Dr. Mazlina Abdul Majid

Position : Associate Professor

Date :

Signature

Name of Co-Supervisor : Prof. Dr. Jasni Mohamad Zain

Position : Professor

Date :

STUDENT'S DECLARATION

I hereby declare that the work on this thesis is my own except for the quotations and summaries which have been duly acknowledged. The thesis has not been accepted in any degree and is not concurrently submitted to the award of another degree.

Signature	
Name	: SUTARMAN
Position	: Ph.D. Student
Date	:

CORRELATION-BASED SUBSET EVALUATION OF FEATURE SELECTION FOR DYNAMIC MALAYSIAN SIGN LANGUAGE

SUTARMAN

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Doctor of Philosophy in Computer Science

Faculty of Computer Systems and Software Engineering
UNIVERSITI MALAYSIA PAHANG

SEPTEMBER 2016

DEDICATION

I would like to dedicate this thesis to my parents, my beloved wife Subandiyah, S. Pd., for her love and support, my children Hammam ZM, Nur Aqilah Afifah, Haya Aribah Sholehah, the beauties of my life, my brother as well as my friends who have always supported me in completing this research.

ACKNOWLEDGEMENTS

First of all, I would like to thank Allah S.W.T. for giving me the strength and knowledge to finish this thesis and for blessing me with many great people who have been my greatest support in life. Shalawat and Salam for the Prophet Muhammad S.A.W.

I wish to express my sincere and deepest thanks to my supervisor, Dr. Mazlina Abdul Majid for her guidance, help and encouragement during the three-year period of my Doctor of Philosophy program. Her invaluable help with constructive comments and suggestions throughout the simulation and thesis works have contributed toward the success of this research. Thank you also to my co-supervisor Prof. Dr. Jasni Mohamad Zain, for her best support and knowledge on this topic.

I would like to extend my gratitude to Prof. Bambang Hartadi, Ph.D, MM, CA, CPA., chancellor of University of Technology in Yogyakarta (UTY) who has given me the opportunity to pursue my doctoral studies.

Last but not least, my deepest gratitude to my beloved parents, my beloved wife Subandiyah for her deeply love and endless support, prayers and encouragement. To those who indirectly contributed to this research, your kindness means a lot to me. Thank you very much.

ABSTRACT

Sign language is used for communication to the deaf and speech impaired. For communication between the common man and the deaf, sign language interpreter is needed for understanding natural language and vice versa. Sign Language Recognition (SLR) aims to translate sign language into text so that the communication between the deaf and the general public can be done comfortably. Research in Sign Language Recognition (SLR) has been widely done by researchers from many various countries using different datasets. In the existing work of Sign Language Recognition, most researchers divide the process in four main steps: image acquisition, pre-processing, features extraction and classification. The success for the classification process is determined by many factors. One factor is the quality of the data or information held. The process of data model extraction will be more difficult if the information held is irrelevant or contains redundancies, or if the data obtained contains high noise. Thus by adding processes before classification methods such as feature selection methods can provide better data input in the classification process, it is expected to improve the performance of the method of classification. Feature selection potential is used in SLR. Currently, there is no research work that used Feature Selection on Sign Language Recognition. In this study, the Correlation-based Feature Subset Evaluation (CfsSubsetEval) and Artificial Neural Network (ANN) has been proposed, in order to improve the accuracy rate on the recognition of sign language. The data samples tested were 15 dynamic signs taken from the Malaysian Sign Language (MySL). Pre-processing in this study was based on tracking the joints on a skeleton feature for generating 3D coordinates X, Y, Z. The sample of 3D data coordinates of X, Y, and Z axis is a value relative to the torso and head. In this study, the images has been captured using a kinect sensor based skeletal algorithms. The feature extraction was done by normalizing the position and size of the user, by taking eight out of 20 joints that contribute in identifying the movement of the hands; left hand, right hand, left wrist, right wrist, left elbow, right elbow, torso and head. CfsSubsetEval and Artificial Neural Network have been compared with Consistency-based Subset Evaluation (CSE) and Correlation-based Attribute Evalualtion (CorrelationAttributeEval) for performance analysis on result accuracy. In this study, spherical coordinate conversion process and segmentation frame using mean function were used. The experiments have achieved 95.56 % in accuration rates for Correlation-based Feature Subset Evaluation (CfsSubsetEval).

ABSTRAK

Bahasa isyarat digunakan sebagai Bahasa komunikasi untuk orang pekak dan pertuturan terjejas. Untuk komunikasi antara manusia biasa dengan orang pekak, jurubahasa bahasa isyarat diperlukan untuk bahasa tabii dan sebaliknya. Pengecaman Bahasa Isyarat bertujuan untuk menterjemahkan bahasa isyarat ke dalam teks supaya komunikasi antara orang pekak dan orang awam boleh dilakukan dengan selesa. Penyelidikan dalam Pengecaman Bahasa Isyarat telah wujud secara meluas oleh penyelidik dari berbagai negara yang menggunakan set data yang berbeza. Dalam penyelidikan semasa Pengecaman Bahasa Isyarat, kebanyakan penyelidik membahagikan proses dalam empat langkah utama: perolehan imej, pra-pemprosesan, pengekstrakan dan klasifikasi. Kejayaan proses klasifikasi ditentukan oleh banyak faktor. Salah satu faktor adalah kualiti data atau maklumat yang dipegang. Proses pengekstrakan model data akan menjadi lebih sukar jika maklumat tidak relevan atau mengandungi lebihan, atau jika data yang diperolehi mempunyai kandungan hingar yang tinggi. Dengan demikian penambahan kaedah pemilihan fitur sebelum kaedah klasifikasi boleh memberi input data yang lebih baik dan sekaligus dijangka dapat meningkatkan prestasi kaedah klasifikasi. Kaedah Pemilihan fitur berpotensi untuk digunakan dalam Pengecaman Bahasa Isyarat. Pada masa ini, tidak ada penyelidikan yang menggunakan Kaedah Pemilihan fitur pada Pengecaman Bahasa isyarat. Dengan itu, penggunaan Kaedah Korelasi-Ciri Subset Penilaian (CfsSubsetEval) dan Rangkaian Neural Buatan dicadangkan dalam kajian ini untuk meningkatkan kadar ketepatan pada Pengecaman Bahasa Isyarat. Sampel data yang diuji diambil dari 15 perkataan yang bercorak dinamik dari Bahasa Isyarat Malaysia (BIM). Pra-pemprosesan dalam kajian ini adalah berdasarkan kepada konsep mencari sendi pada ciri tulang rangka untuk menghasilkan 3D koordinat X, Y, Z. Sampel koordinat data 3D paksi X, Y, dan Z adalah nilai yang relatif antara bahagian badan dada (*torso*) dan kepala. Di dalam kajian ini, imej dicapai menggunakan alatan Kinect dan algoritma rangka. Pengekstrakan ciri dilakukan dengan menormalkan kedudukan dan saiz pengguna dengan mengambil lapan daripada 20 sendi yang menyumbang dalam mengenal pasti pergerakan tangan; tangan kiri, tangan kanan, pergelangan tangan kiri, pergelangan tangan kanan, siku kiri, siku kanan, bahagian badan dada dan kepala. Kaedah Korelasi-Ciri Subset Penilaian (CfsSubsetEval) dan Rangkaian Neural Buatan telah dibandingkan dengan konsistensi Subset Penilaian (CSE) and korelasi Atribut Penilaian (CorrelationAttributeEval) untuk mengkaji kadar ketepatan kaedah yang dicadangkan. Hasil penyelidikan mendapati kadar ketepatan sebanyak 96.56% pada Korelasi-ciri Subset penilaian (CfsSubsetEval).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

D_i	The number of occurrences of the i^{th} attribute value combination
f	Activation function
M_i	The cardinality of the majority class for the i^{th} attribute value combination
N	The total number of instances in the data set.
$\overline{r_{cf}}$	The average feature-class correlation
$\overline{r_{ff}}$	The average feature inter-correlation
α	Determine learning rate
s	Attribute subset
δ_k	The error of hidden layer
r	Polar Angle
θ	Polar Angle
φ	Azimuthal Angle
μ	(statistics) population mean
W_{ji}	Connection weights
$\forall$	Universal Quantifier
$\sum$	Sigma
$\neq$	Not equal
®	Register
W_0	Threshold

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
AI	Artificial Intelegent
BPNN	Backpropagation Neural Network
CFS	Correlation-based Feature Selection
CON	Consistency-based Subset Evaluation
DNNs	Deep Neural Networks
DTW	Data Time Warping
ESL	Ethiopian Sign Language
GF	Gabor Filter
HMM	Hidden Markov Model
ISL	Indonesian Sign Language
JSL	Japanies Sign Language
k-NN	K-Nearest Neighboar
KTBM	Kod Tangan Bahasa Melayu
MySL	Malaysian Sign Language
PSL	Persian Sign Language
SGONG	Self-Growing and Self-Organized
SIBI	Sistem Isyarat Bahasa Indonesia
SIFT	Scale Invariants Feature Transform
SLR	Sign Language Recogntition
WEKA	Waikato Environment for Knowledge Analysis
CSE	Consistency-based Subset Evaluation
CfsSubSetEval	Correlation-based SubSet Evaluation
CorrAttributEval	Correlation-based Attribute Evaluation

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

This chapter introduces the research background and the motivation for this study, followed by overview of research study, scope of study, thesis contributions, and lastly the thesis organization.

1.2 BACKGROUND AND MOTIVATION

Sign language is the language used by the deaf and mute. It is a combination of form and hand movement, orientation and movement of the body between the hands, arms, body, and facial expressions to fluidly express a speaker's thoughts (Mekala Gao Fan Davari and Eng, 2011). However, communication with normal (or non-speech impaired) people is a major handicap for them since normal people do not understand sign language. Sign Language Recognition (SLR) is needed for realizing a human oriented interactive system, which can aid interactions between the speech impaired and the general public. For communication between an ordinary person with a deaf person, a translator is usually required to translate the sign language into natural language and vice versa (Quan and Jinye, 2008). Sign Language Recognition (SLR) aims to translate the sign language into the visual text so that the communication between the deaf, and the general public would be more comfortable. Generally, most people are not familiar with sign language and will have difficulty in communicating with the speech impaired. However, by translating natural text to sign language, the software can aid communication and provide interactive training for people to learn sign language. Research in Sign Language Recognition (SLR) has been widely done by researchers from many various countries using different datasets. In the existing work of Sign Language Recognition, most researchers divide the process in four main steps (Khan and Ibraheem, 2012a; Mohandes, 2013): image acquisition (input image from camera, videos or data glove), pre-processing (convert movie frame to indexed image + filtering), features extraction, and classification or recognition as illustrated in Figure 1.1

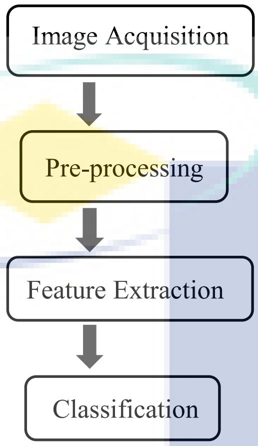


Figure 1.1 Sign Language Recognition Steps Diagram

There are several methods to acquire data in image acquisition such as the data glove (Q. Chen El-Sawah Joslin and Georganas, 2005; Gao Wu Wang and Jiyong Gao Jiangqin, Wu Chunli, Wang, 2000; Maraqa Abu-Zaiter and Manar Maraqa, 2008; Swee Salleh Ariff Ting and Seng, 2007), video using RGB sensor (Akmeliawati Ooi and Ye Chow Kuang, 2007; Al-Rousan Assaleh and Tala'a, 2009; Geer, 2004; Karabasi Bhatti and Shah, 2014; Karami Zanj and Sarkaleh, 2011; Lee Tsai and Lee Cheng-Yueh, 2009) Using the attached sensors, the joint angles and spatial positions of the hand can be measured directly from the glove. However, the data glove and its attached wires are still inconvenient and rigid for users to wear. The use of video RGB sensor is simpler than the data glove but video RBG is unable to provide information from users, such as the position and size of the user. In addition, the accuracy of the results using RGB sensor video is more influenced by the complex background, lighting variations, and other skin color with hand objects, in addition to the system requirements such as speed, recognition time, robustness and computational efficiency.

Currently, there is image acquisition method using the camera with RGB and depth sensor. This is based on the existing advantages such as: the RGB and depth cameras, in general, are well suited for sign language recognition. It offers 3D data from the environment without a complicated camera setup and efficiently extracts the user's body parts, allowing for recognition of not just hands and head, but also other parts such as elbows that can be of further help in distinguishing between similar signs. The technology that uses the RGB camera and depth sensor such as the Kinect sensor. Althought, the cost of kinect sensor is still too expensive for regular users. Several researchers have used the camera with RGB and depth sensor such as Celebi et al., (2013), Capilla, (2012), Sakhalkar et al., (2014).

After the data has been acquired, it is described via features. The features selected to often depend on sign language elements that are detected. Feature extraction is done to get quantities, which showed the specificity of the data processed. The feature extraction is one of the most important and influential in the accuracy of the recognition results.

The final stage during the process of SLR is the classification. Sign classification method is used to recognize the sign. The recognition process is effected by proper selection of suitable features and classification of parameter algorithm.

Currently, there are several methods of classification used is the Hidden Markov Model (HMM) and researchers who use such as (Yona Falinie A Gaus Wong Chin Porle and Chekima, 2013; Yona Falinie Abdul Gaus and Wong, 2012; Lang Block Rojas and Simon Lang Rojas Raúl, 2011; Sandjaja and Marcos, 2009; Swee et al., 2007; Vogler and Metaxas, 1999; Wong Sainarayanan Abdullah Chekima Jupirin and Gaus, 2013), Data Time Warping (DTW) and researchers who use such as (Capilla, 2012; Celebi et al., 2013; Iqbal Purnomo and Purnama, 2011; Jangyodsuk and Conly, 2014; Mauridhi Purnama and Purnomo, 2011; Reyes Domínguez and Escalera, 2011), Artificial Neural Network (ANN) and researchers who use such (Adithya R and Gopalakrishnan, 2013; Admasu and Raimond, 2010; Akmeliawati et al., 2007; Bailador Tröster and Roggen, 2007; Karami et al., 2011; Maraqa et al., 2008; Munib Habeeb Takruri and Al-Malik, 2007; Sakhalkar et al., 2014; K. Sharma and Garg, 2014).

The success for the classification process is determined by many factors. One factor is the quality of the data or information held. The process of data model extraction will be more difficult if the information held is irrelevant or contains redundancies, or if the data obtained contains high noise (Hall and Holmes, 2003). Thus by adding process before classification methods such as feature selection methods can provide better data input in the classification process, it is expected to improve the performance of the method of classification.

Therefore, the selection process requires attributes to be sorted; the good attributes and those that are not relevant. Algorithm based at the level of correlation data, and the level of data consistency are an algorithm for the feature's selection that has been developed.

The aim of feature selection is to choose a subset of features for improving prediction accuracy or decreasing the size of the structure without significantly decreasing prediction accuracy of the classifier built using only the selected features (Dash and Liu, 1997; Koller, D. and Sahami, 1996).

Basically, feature selection is a technique of selecting a subset of relevant/important features by removing most irrelevant and redundant features (Jhon Kohavi and Pfleger, 2015) from the data for building an effective and efficient learning model (Dash and Liu, 1997).

1.3 PROBLEM STATEMENT

There are several methods to acquire data in image acquisition such as the data glove, video using RGB sensor, video using RGB and depth sensor. There are some problems in image acquisition data such as information about the user (the position and size of the user), the complex background, lighting variations, and other skin color with hand objects, the lighting conditions.

The use of video RGB sensor is simpler than the data glove but video RBG is unable to provide information from users, such as the position and size of the user. The use of RGB video and depth sensor can be the solution, because this method can provide information about the user (the position and size of the user).

In the dynamic of sign language, there are many variations of the sign such as a model different movement that involve the hands, fingers, wrists, elbows. Therefore, the number of joints in the skeletal features are important parameters in the sign language translators. Previous studies using six joints such as (Capilla, 2012; Celebi et al., 2013) and Sakhalkar et al., (2014) used seven joints. By using six joints and seven joints, one cannot handle sign language that has more complex characteristics, such as the movement of the wrist and fingers. Currently, there is no dataset of dynamic Malaysia sign language developed using skeletal-based features.

Xu and Yang (2006) have proposed feature selection in the classification process in SLR. They proposed Hill Climbing approach and Random Walk approach, to select the features. They claimed that both algorithms were easy-implemented but reasonable and efficient. Therefore, there are opportunities, the feature selection can be used in SLR. Currently, there is no research work on Sign Language Recognition using Correlation-based Feature Subset Evaluation (CfsSubsetEval).

The quality of data or information are the success factors in the classification process. The process of data model extraction will be more difficult if the information held is irrelevant or contains redundancies, or if the data obtained contains high noise. Adding filtering process before classification methods can provide better data input in the classification process, is expected to improve the performance of the method of classification.

Thus, the research questions that arise, here are:

a. Does the number of skeleton joints give an impact to accuracy of dynamic Malaysian Sign Language recognition?
b. Does the Feature Selection with Artificial Neural Network give any impact to the accuracy of Dynamic Malaysian Sign Language Recognition?

1.4 RESEARCH OBJECTIVES

The objectives of this research are:

a. To acquire data (image acquisition) and skeletal feature of Sign Language Recognition (SLR) using depth and color sensor feature in Kinect.

b. To develop the Features Selection method using Correlation-based Feature Subset Evaluation (CfsSubsetEval).
c. To combine CfsSubsetEval and Artificial Neural Network for improving the accuracy of Dynamic Malaysian Sign Language Recognition.

1.5 RESEARCH SCOPES

The scopes of study are:

a. Image Acquisition Data collected from the Kinect camera, using skeletal data tracking with six, seven and eight joint positions.
b. Total data samples are 375 that consist of 15 dynamic signs that often been used in daily communication in Malaysian, from 5 people. Each sign in captured 5 times by the same people in order to collect the dynamic sign.
c. Features Selection process evaluation using Correlation-based Feature Subset Evaluation (CfsSubsetEval), Consistency-based Subset Evaluation (CSE), Correlation-based Attribute Evalualtion (CorrelationAttributeEval. The algorithm to search for a subset uses Best First and Ranker.

1.6 THESIS CONTRIBUTIONS

The work carried out in this thesis seeks to produce a key contribution to the body of knowledge in Sign Language Recognition System. The contributions of this thesis aim to improve the recognition process in SLR using three main ideas:

1. Image acquisition and skeletal feature using RGB and depth sensor to obtain joint positions. This study uses 8 of the 20 joints.
2. Proposed Feature Selection method using Correlation-based Feature Subset Evaluation (CfsSubsetEval).
3. Combining the Correlation-based Feature Subset Evaluation (CfsSubsetEval) with Artificial Neural Network for improving the accuracy of Dynamic Malaysian Sign Language Recognition.

1.7 THESIS ORGANIZATION

This thesis consists of five chapters, structured as follows:

Chapter two gives an account of the literature on Sign Language, Malaysian Sign Language, image acquisition, skeleton feature, feature selection and Sign Language Recognition methods.

Chapter three explains the research methodology used to achieve the research objectives. It starts with the research strategy, dataset collection, feature extraction, feature selection and followed by classification using Artificial Neural Network.

The chapter four, explains the proposed method. It starts with CFsSubsetEval of feature selection with detailed explanation about an algorithms and flowchart of the feature selection, then there is a classification process using ANN with detailed explanation about the architecture and the parameter of Neural Network.

In Chapter five, an explanation of experiments are conducted to evaluate CfsSubsetEval with Artificial Neural Network, CSE, CorrelationAttributeEval on different number of joints (6, 7 and 8 joints). In this chapter, the experiment and the results are described and comparison are made between CfsSubsetEval, CSE and CorrelationAttributeEval in terms of accuracy.

The chapter six, this chapter presents the conclusions, recommendations, and future work for the research.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter reviews existing research studies on sign language recognition system. The first review is on the theory of Sign Language and Malaysian Sign Language (MySL). The second is image acquisition; data glove approaches and vision based approaches. The third is of feature selection; correlation-based feature selection and consistency-based subset evaluation. Then this chapter will discuss the methods used for the recognition system; Artificial Neural Networks (ANN), Hidden Makrov Model (HMM) and Dynamic Time Warping (DTW). Next, existing studies comparing the methods are discussed and the research gaps in sign language recognition system is identified. Finally, a discussion of existing literature in sign language recognition system using Artificial Neural Network is presented.

2.2 THEORY FOR SIGN LANGUAGE AND MALAYSIAN SIGN LANGUAGE (MYSL)

2.2.1 Sign Language

The term sign language is similar to the term language;, there are many of both spread in the various territories of the world. Just like languages, sign language has been developed a long time back. It has its own language grammar and vocabulary, and thus considered a real language (Braem,1995).

Sign Language is a gesture language which visually transmits sign patterns using hand shapes, orientation and movements with the hands, arms or body, facial expressions and lip-patterns to convey word meanings instead of acoustic sound patterns. Sign languages that exists around the world is different, each with its own vocabulary and gestures. Some examples are American Sign Language (ASL), German Sign Language (GSL), British Sign Language (BSL), Malaysian Sign Language (MySL) and many others. This type of language is commonly used in deaf communities, including interpreters, friends, and families of the deaf, as well as people who have difficulties of hearing themselves. However, these languages are not commonly known outside of these communities, and therefore communication barriers exist between the deaf and hearing people. Sign language communication is multimodal; it involves not only hand gestures (i.e. Manual signing) but also non-manual signals.

Gestures in sign language are defined as specific patterns or movements with the hands, face or body to make our expressions. Gestures are a form of movement through body language or non-verbal communication. Hand gestures are classified under several categories such as controlling gestures, conversational gestures, manipulative gestures and communicative gestures (Kelly, 2010).

Hand gestures are classified into two types; static and dynamic gestures (Athavale and Deshmukh, 2014). Static hand gestures are defined as the orientation and position of hand in the space during a period of time without any movement while if a movement is detected in the aforementioned time duration, it is called dynamic gestures. Dynamic hand gestures include gestures involving body parts like waving of hand while static hand gestures include single formation without movement like jamming the thumb and forefinger to form the "ok" symbol in a static pose which represents the static gesture.

Until the late 1960s, sign language is generally not considered a real language. Instead, as Boyes Braem states, most linguists never seriously analyzed them and assume they can just set loosely connected movements to express a simple relationship (Lang Rojas and Block-Berlitz, 2012).

William C. Stokoe, a professor and lecturer in English at Gallaudet University in the United States concentrated upon the analysis of American Sign Language (ASL) and was among the first to explore methods of communication for the deaf with up-to-date means of linguistics. His work, however, was approached with scepticism; even deaf people and hearing people with sign language as their mother tongue deemed themselves backward and not considered a real language that can also be used for the necessary communications in higher education (Braem, 1995).

At the same time, two other researchers, namely Ursula Bellugi and her husband Edward Klima, began the study of sign language where they filmed the movement in ASL and analyzed them (McKee, 2014). Both of them have never had contact with the deaf before learning, and their goal was to analyze the sign language from a neutral point without a prior opinion. Their conclusion was that ASL has its own grammar, and the most basic features of a language. After the first results from the study presented a modern sign language, it was immediately accepted by both the deaf and the hearing.

Dynamic hand gestures done intentionally for communication are called conscious dynamic gestures, whereas unintentional (unaware) gestures carried out during casual physical activity is known as the unconscious dynamic gesture. 35 % of human communication consists of verbal communication, and 65 % are non verbal gesture based communication (Athavale and Deshmukh, 2014).

Sign language also has its own syntax and grammar. A common misconception in sign language is that it is patterned after the vocally produced languages of that country, and that the signs manually produce English words. Sign Language has the phonology, morphology, syntax and grammar that is independent of spoken language. Morphological structures in sign language have simultaneous so different morphemes within a word simultaneously superimposed on each other rather than strung, as those of spoken language usually. This is one of the main differences between signed and spoken language. For example, manual signs are conveyed sequentially, where each sign comes one after the other. However, in addition to being conveyed sequentially, each manual sign can occur in parallel to another manual sign performed by the other hand, as well as actions such as facial expressions or head and body movements. The linguistic characteristics of sign languages, therefore, differ greatly from those of spoken languages.

The sign language is the fundamental communication method among people who suffer from hearing defects. In order for an ordinary person to communicate with deaf people, a translator is usually needed to translate sign language into natural language and vice versa. Recognition of sign language is needed to realize the human-oriented interactive system, which can interact as normal communication. The function within a translator system transcribes symbols in sign languages into plain text that can help with real-time communication and may also provide interactive training for people to learn a sign language.

In Sign Language Recognition system most researchers have divided it in four main phases (Khan and Ibraheem, 2012a): image acquisition (input image from camera, videos or data glove), pre-processing (convert movie frame to indexed image + filtering), features extraction, and classification or recognition.

2.2.2 Malaysian Sign Language (MySL)

In Malaysia, the standard sign language is a Malaysian Sign Language used as an everyday communication for the deaf community. Malaysian Sign language has a different dialect based upon the state in Malaysia (Masitry Majid Toh Sutarman and Herawan, 2013). Malaysian Sign Language (MySL) is designed by the deaf naturally based on factors such as daily communication and lifestyle of the deaf and mute in Malaysia (Wong et al., 2013). MSL is not a spelled out of oral language or letters in gesture form.

In Penang, this sign language is called as a Penang Sign language (PSL), and this was mostly exploited by the elder people. Selangor Sign languages (SSL) is well recognize as Kuala Lumpur Sign language (KLSL). Selangor Sign Language is originally from America Sign Language (ASL), where some of the sign symbols of American Sign Language (ASL) is used to develop the Malaysian Sign Language (MySL) with the certain sign being different from the American Sign Language because of the cultural and social norms in Malaysia (R. G. Gordon Jr., 2005).

Malaysian Sign Language (Bahasa Isyarat Malaysia, or BIM), is the sign language used every day in many parts of Malaysia. BIM has many dialects and differs from state to state. Malaysian Sign Language was born when the Malaysian Federation of the Deaf was established in 1998, and the usage has expanded among deaf leaders and participants. American Sign Language (ASL) has had a strong influence on BIM, but the two are considered different languages. BIM in turn has been the basis for Indonesian Sign Language (Masitry et al., 2013).

Kod Tangan Bahasa Malaysia or Manually Coded Malay (KTBM) was created by hearing educators and linguists in between 1980 and 1986 and remains the only sign language recognized by the Malaysian Ministry of Education. Other sign languages in use in Malaysia include Penang Sign Language (PSL) and Selangor Sign Language (SSL or KLSL). These two sign languages began in 1980 before MSL/BIM when Penang, Selangor and Kuala Lumpur became popular destinations for employment opportunities, entertainment, disability benefits.

Additionally, every parent of deaf children has their own signs called home signs to make a gestural communication. The use of such home signs among Peranakan or ethnic Chinese users of BIM may be behind the controversy over the supposed influence of Chinese Sign Languages, which does not seem to be well documented and may merely be based on ethnic stereotyping.

Mamat et al., (2014) developed multimedia applications that are capable to translate from Malaysian Text to MCMSL and helps students to learn sign language effectively and fun.

The development of MySL database is highly important in order to fulfill the needs of the researcher, especially for the research of MySL. The MySL database has been proposed and presented in detail by Maarif et al., (2012). The development of the MSL database is expected to support the research in MSL recognition. For the classification, they proposed database, the MSL is classified into One Hand, Two Hands, Static, and Dynamic. The following are some examples of dynamic Malaysian Sign Language:

- Assalamualaikum, as show in Figure 2.1 - Right hand 'A' while the thumb is imposed on the right edge of the forehead and then moved forward.

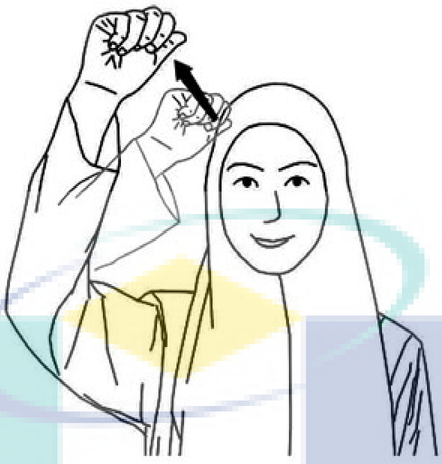


Figure 2.1 The example of ”Assalamualaikum” sign

- Siapa, as show in Figure 2.2 - Raise the right hand 'D', and put the finger as clockwise around the lips.

Figure 2.2. The example of ”Siapa” sign

- Selamat, as show in Figure 2.3 - Right hand 'B', fingertips imposed on the right edge of the forehead and then moved forward.

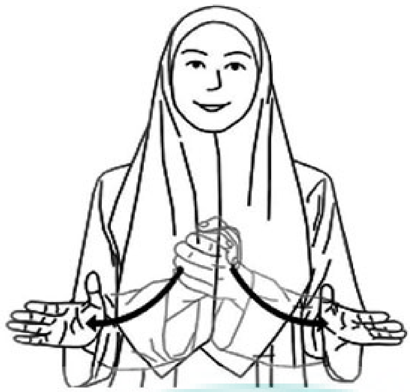


Figure 2.3. The example of ”Selamat” sign

- Apa kabar (as show in Figure 2.4 - make sign 'Apa' (1) and then both sides palms make a signal as 'K', and then moved forward (2)).

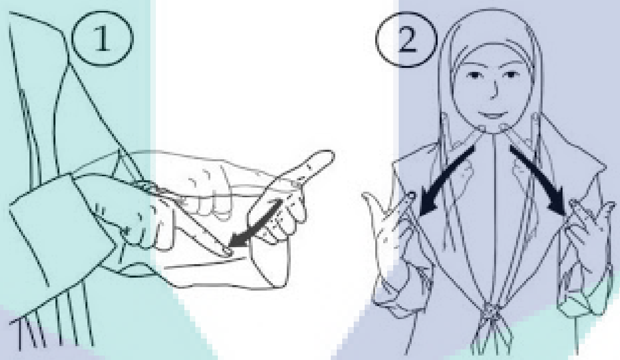


Figure 2.4. The example of ”Apa khabar” sign

- Kamu (as show in Figure 2.5 - Right hand 'A', with the thumb upright and point forward).

Figure 2.5. The example of ”Kamu” sign

- Selamat Pagi as show in Figure 2.6 - Make sign "selamat" – (1), then make sign "pagi" – (2)

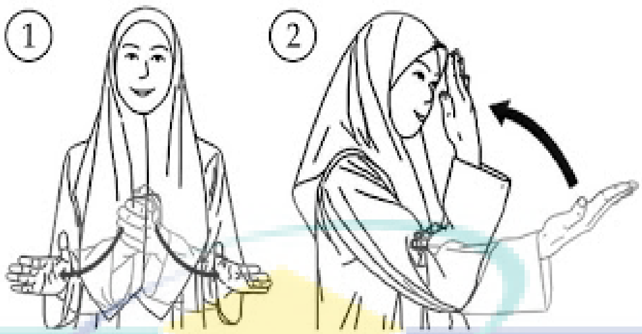


Figure 2.6 The example of ”Selamat Pagi” sign

- Selamat Petang, as show in Figure 2.7 - Make sign "selamat" - (1), then make sign "petang" – (2)

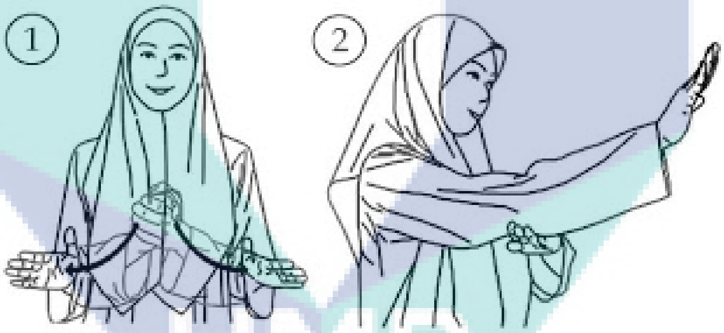


Figure 2.7 The example of ”Selamat Petang” sign

2.3 IMAGE ACQUISITION

Generally, image acquisition in image processing can be defined as a way to retrieve a picture or video from a source, usually a hardware-based source, so it can be passed through to whatever process that needs to occur afterward. Image acquisition should be mostly perfect as possible for efficient sign language recognition. Suitable input device should be selected for data acquisition. There are many input devices that have been developed for data acquisition, able to recognize movement, they are categorized into two: data glove-based and vision-based approach .

2.3.1 Data Glove Approaches

One of the traditional sign capturing methods is data glove approach **Error! Reference source not found.**. These methods employ mechanical or optical sensors attached to a glove that transforms finger flexions into electrical signals to determine the hand posture (Xingyan, 2003). This method requires the glove must be worn and a wearisome device with a load of cables connected to the computer, which will hamper the naturalness of user-computer interaction (Mitra and Acharya, 2007).

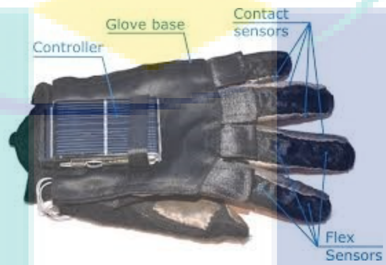


Figure 2.8 The example of data glove

Source http://enabletalk.com/prototype.html

The data glove-based methods use sensor devices for digitizing hand and finger motions into multi-parametric data. The extra sensors make it easy to collect hand configuration and movement. However, the devices are quite expensive and bring much cumbersome experience to the users (Garg Aggarwal and Sofat, 2009).

Researchers who have used the data glove method such as (Gao et al., 2000; Maraqa et al., 2008), used sensor devices for digitizing hand and finger motions for multi parametric data. However the devices are quite expensive and inconvenient to the user and it is difficult in unconstrained background. Extracting skin regions from a range of color has also been done. But the drawback is that it cannot be completely done because skin color depends on each signer or each situation.

Swee et al., (2007) used the Hidden Markov Models (HMM) method to develop a system that can convert the Malay sign language into speech so that has the capability in verifying approximately 25 words using 2 data gloves plus accelerometers.

Oz and Leu, (2011) used a sensory glove called the Cyber glove^{TM} (as shown in Figure 2.9) and a Flock of Birds 3-D motion tracker to extract the gesture features. The results show that the recognition accuracy of the system is about 90 %.

Figure 2.9 The CyberGlove II from the Immersion Corporation

Source: Oz and Leu (2011)

According Mitra and Acharya (2007), the disadvantage of the method that requires the glove to be worn, which is a wearisome device with many cables connected into the computer that will hamper the naturalness of the user's computer.

Some of the weaknesses of data glove-based approach: the data glove can be quite expensive. While it is possible to utilize cheaper data-gloves, they are much more susceptible to noise, and reduce the number of sensors that leads to loss of important information about the hand. This led to a loss of accuracy in sign translation. Furthermore, the data-glove however is less comfortable for signer.

2.3.2 Vision Based Approaches

In the vision-based methods, the system requires only a camera depicted in Figure 2.10 to capture the image required for the natural interaction between human and computers and no additional devices are needed (Khan and Ibraheem, 2012b). It is more natural and useful for real time applications. This approach is simple, but there are many challenges from generated movements such as complex background, lighting variations, and other skin color objects with the hand objects, in addition to the requirements of systems such as speed, recognition time, robustness, and computational efficiency. In order to create a database for a gesture system, the gestures should be selected with their relevant meaning, in which each gesture may contain multi samples (Mitra and Acharya, 2007) to increase the accuracy of the system.

Figure 2.10 Examples of data vision

Source: Mitra and Acharya (2007)

Vision-based method is widely deployed for sign language recognition. Sign gestures are captured by a fixed camera in front of signers. The extracted images convey posture, location and motion features of the fingers, palms and face. Next, an image-processing step is required in which each video frame is processed in order to isolate the signer's hands from other objects in the background (Garg et al., 2009).

The latest computer vision technologies and the advanced computer hardware capacity make real-time, accurate and robust hand tracking and gesture recognition promising. One of the latest technologies of computer vision is Kinect Camera. Kinect is a motion sensing input device developed by Microsoft, which consists of a webcam-style add-on peripheral. It enables the users to control and interact with the console/computer through gestures and spoken commands.

Figure 2.11 shows the physical appearance and sensor components of the first generation of the Kinect device. The color sensor is a RGB camera. An infrared (IR) emitter emits infrared light beams, and the reflected IR beams from the environment come back to the IR depth sensor (Corporation, 2013). The distances between different objects and the sensor are obtained based on the reflected beams. A multi-array microphone containing four microphones can be used for capturing sound. The tilt motor is capable of vertically tilting the sensor bar with a range of ±27º.

The video streams use the VGA resolution (640×480 pixels) with 8 bits per channel for the RGB video and 11 bits for the depth video. The maximum frames per second (fps) can reach 30 fps. The angular field of view of the sensor is 43º vertically and 57º horizontally. The optimal sensing distance ranges from 1.2 meters to 3.5 meters (Corporation, 2015).

Microsoft Kinect was initially developed as a peripheral device for use with the XBOX 360™ gaming console. Its three sensors, i.e. RGB, audio, and depth are able to detect movements, to identify user faces/speeches, and allow the players to play games using only their own body as the controller. In contrast to previous attempts at gesture or movement-based controls, there is no need of wearing accessories in order for the device to track the users. Although its initial goal was for the gaming purpose, Microsoft Kinect opened the door to a wide number of useful applications in the field of computer vision, such as action recognition, gesture recognition, virtual reality, and robotics (Corporation, 2015).

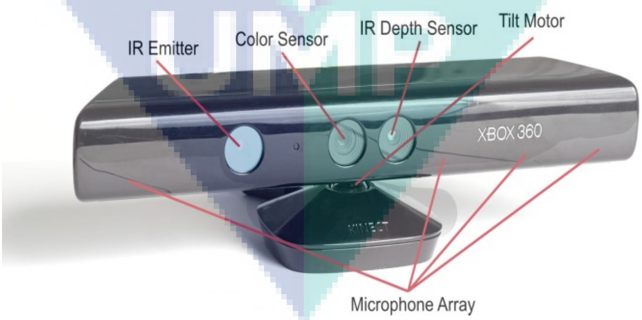


Figure 2.11 The Kinect device

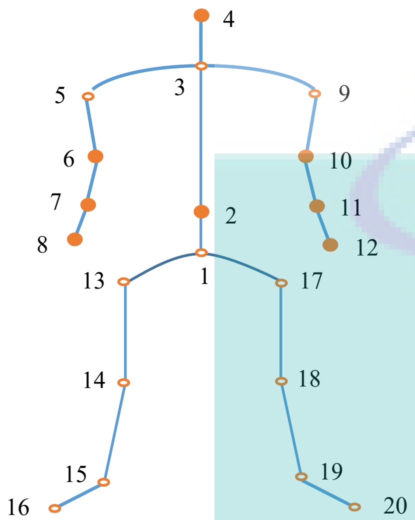
Source: Corporation (2015)

It features an RGB camera along with a microphone array and a depth sensor using an infrared projector, and is thus capable of tracking a user's full body independently from conditions of lighting. Kinect for XBOX 360™ is changing the game. Besides its application on XBOX360™, there are several drivers that allow Kinect to be used on PC and Mac. Driver OpenNI was released in December 2010 by PrimeSense, the manufacturer of Kinect's camera technology. It includes a feature-rich open source framework licensed under the GNU lesser General Public License, version 3, and can be combined with closed source middleware called NITE for skeletal tracking and recognition of hand gestures. Kinect as an XBOX360™, motion sense game controller, can be used to garner video with depth information as well as track the skeletal movements of the game. Now the Kinect can also be used to recognize body motion without connecting to an XBOX because it can be connected to a normal PC to collect information.

Vision based gesture recognition is mainly divided into static hand gesture recognition and dynamic hand gesture recognition (Athavale and Deshmukh, 2014). The dynamic hand gesture recognition can be divided into hand type based dynamic hand gesture recognition and the palm of the hand trajectory based dynamic hand gesture recognition. The former can be regarded as the continuous static gesture sequences, while the latter mainly studies the meaning of hand trajectory (Y. Wang Yang Wu Xu Li and Wang Cheng Wu, Xiaoyu Xu, Shengmiao Li, Hui Xu, Shengmiao Li, Hui, 2012).

The latest computer vision technologies and the advanced computer hardware capacity make real-time, accurate and robust hand tracking and gesture recognition promising. One of the latest technologies of computer vision is the Kinect Camera. Kinect is a motion sensing input device developed by Microsoft, which consists of a webcam-style add-on peripheral. It enables the users to control and interact with the console or computer through gestures and spoken commands. The Microsoft Kinect camera can produce depth and RGB streams for an object at a cost that is lower than using the conventional range sensor approach.

Microsoft Kinect sensor is used in Shotton et al., (2011) to obtain joint positions. Kinect SDK tracks 3D coordinates of 20 body joints given in Figure 2.12. in real time (30 frames per second). Since the machine learning algorithm uses depth images to predict joint positions, the skeleton model is quite robust to color, texture, and background. To track users, skeletal tracking must first be enabled in Kinect. The information about the tracked users is provided in the form of an array of Skeleton objects present within the frame. The skeletons in a frame can be chosen to either 'tracked' or 'position only'. If ‘tracked’ is chosen, the tracking state is provided by Kinect. On the other hand, a skeleton with tracking the state of 'position only' gives information on the position of the user, without the details of the joints.



1. Hip Center	11. Wrist Left
2. Spine / Torso	12. Hand Left
3. Shoulder Center	13. Hip Right
4. Head	14. Knee Right
5. Shoulder Right	15. Ankle Right
6. Elbow Right	16. Foot Right
7. Wrist Right	17. Hip Left
8. Hand Right	18. Knee Left
9. Shoulder Left	19. Ankle Left
10. Elbow Left	20. Foot Left

Figure 2.12. Kinect 20 Joint positions

Source: Shotton et al. (2011)

The main component of the Kinect is the time-of-flight camera that measures the distance of any given point from the sensor by using the time taken by IR light to reflect from the object. Along with this, the surface curvature is modeled by projecting an IR grid and obtaining the deformation information from the grid.

Several researchers who have been using vision-based methods such as Sanchez-Nielsen et al., (2003) suggested a real time vision system that uses a fast segmentation method, by using minimum features to identify hand posture in order to speed up the recognition process. Imagawa et al., (2000) presented a local feature recognizer for the sign language recognition system. The Vision based (Brashear Starner Lukowicz and Junker, 2003; Stergiopoulou and Papamarkos, 2009; Tin Hninn, 2009) method, it only required one or two cameras, but the challenges are, it needs to be background invariant, lighting insensitive, allows person and camera independent to achieve real time performance (Garg et al., 2009). Shimada et al., (2001) have studied the extraction based vision features of the silhouette shape of a hand with a match in hand with 3D Computer Graphics (CG) in a simple background.

The framework created as part of this work by Lang et al., (2011) was called Dragonfly that draws gestures on the fly, and mainly consists of two classes that users could use in assimilation with their software. The depth camera serves as an interface to OpenNI, i. e. it updates the camera image and reports the data of the skeleton joints of the body parts, and the data set for recognition.

Agarwal and Thakur, (2013) presented a sign language recognition system by using depth images captured by Kinect camera. The generated feature matrix was then trained using a multi-class SVM classifier. They have made a sign language recognition system that works faster in comparison to other techniques that are based on tracking, hand shape analysis or ones that are developed using high level features.

Karabasi et al., (2014) proposed a model for recognizing Malaysian Sign Language through image processing techniques and converting the visual information into textual information at real-time. They proposed a system based on mobile devices that take user video from the camera, perform image processing and gesture recognition on that image to understand the gesture of the opposite person and then give a textual representation of that gesture.

Gaus et al., (2011) proposed a method to identify hand gesture trajectory in constrained environment. The method consists of three modules: collection of input images, skin segmentation and feature extraction. To reduce processing time, they compared the absolute difference between two consecutive frames then chose the one with highest value. The experimental results show up to 80 % of accuracy in identifying the forms of the gesture trajectory.

Biswas and Basu, (2011) proposed a method to recognize human gestures using a Kinect® depth camera. The camera views the subject in the front plane and generates a depth image of the subject in the plane towards the camera. The results of accuracy could be improved by making use of the skin colour information from the colour camera. This allows the efficient use of depth camera to successfully recognize multiple human gestures.

2.3.3 Skeletal Features

The Skeletal features can be extracted from a skeleton data generated from RGB-D sensor data. The skeleton data includes the 3D coordinates of all skeleton joints and sometimes also information related to the joint angle (X. Chen, 2014).

Simple and effective features are essential to achieve an accurate recognition system. The skeleton extraction is essential for general shape representation and will affect system performance and algorithms complexity. This is especially true for the features concerning both position and motion of joints to determine human body's pose and motion.

In addition, depending on the different body size of the performers, the same sign language by different performers will have different coordinate representations. Therefore in order to directly use the 3D coordinates, it is essential to register the joint coordinates into a common coordinate system. The sources of the skeleton model are mainly the data from a motion capture system and RGB-D sensors. In order to make these skeleton coordinates comparable, all skeletons are rotated into the same orientation and the root joint of the skeleton is translated to the origin, which causes the coordinates of the hip joints of the same actor to overlap regardless of the posture.

For skeletons extracted with an RGB-D sensor, usually only the 3D joint coordinates are available without the rotation and translation information. To transform all skeletons into the same orientation and to translate the root joints to the origin, one skeleton can be selected as a common basis, with its orientation considered as a reference for the other skeletons to be transformed into (X. Chen, 2014).

The skeletons from different sources are often in different formats, as illustrated in Figure 2.13. The other two skeletons are extracted from Kinect data but by different software. The skeleton is from the Microsoft Kinect SDK (Corporation, 2015) and has 20 joints. The rightmost skeleton is from the PrimeSense Natural Interaction Middleware

(NiTE) with 15 joints (Rusu, 2015). Although the number of joints differ, the essential joints are available in all formats. These include the hands, feet, elbows, knees and root.

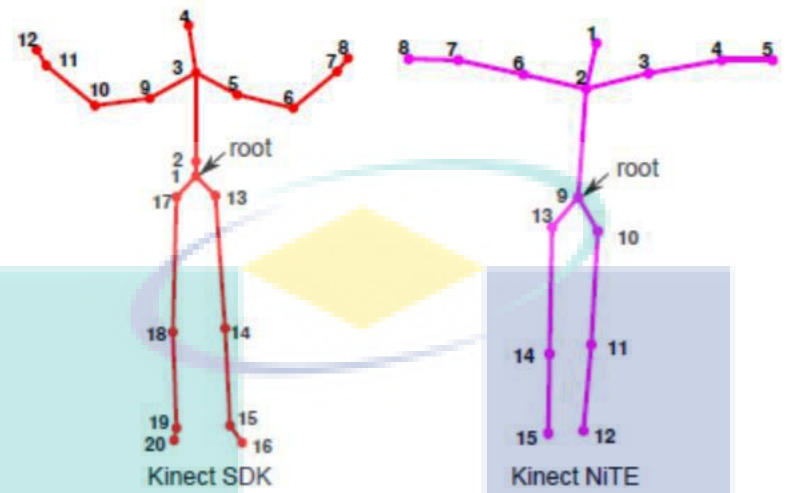


Figure 2.13. Skeleton models from different sources

Source: Rusu (2015)

Capturing image using RGB camera and depth sensor has been found important for sign language recognition (Capilla, 2012; Celebi et al., 2013; Sakhalkar et al., 2014) due to it has the capability to capture 3D data from the environment and efficiently extracted part of the user's body.

In addition, these capturing method allows to capture other part of the body such as elbow which is important in distinguishing between the similar signs. Therefore, in this research, both camera and depth sensor capturing method will be used. Both characteristics can only be found in Kinect, thus, Kinect is chosen as the image acquisition method.

2.4 FEATURE EXTRACTION

Under different conditions, the performance of different feature detectors will be significantly different. The features should be efficiently and reliably extracted to find shapes and robustly irrespective of changes in illumination levels, position, orientation and size of the object in a video/image.

Feature extraction is done to get quantities which shows the specificity of the data processed. The feature extraction is one of the most important and influential in the accuracy of the recognition results. The feature vector thus obtained using any one of the feature extraction methods is used for training the classifier. Thus feature extraction is the most crucial step of sign language recognition since the inputs to the classifier are the feature vectors obtained from this step (Gilorkar and Ingle, 2014).

There are several feature extraction that has been used by researchers such as: Scale invariant Feature Transform (SIFT) used by Lowe, (2004); the Haar-like feature is applied for hand detection by Q. Chen et al., (2008); Principal Component Analysis (PCA) (Nasser Dardas and Georganas, 2011).

2.4.1 Normalized 3D joint positions (NP)

The skeleton model is a constitution of joints represented by 3D coordinates, which contains very rich raw information about the posture. However, the joint coordinates are closely related to the circumstances in which the skeleton model is generated. The coordinate system varies in uncontrolled recording environments, which directly influences the joint coordinate values. Next, even in the same coordinate system, multiple instances of the same gesture performed by the same actor are likely to have different coordinate values due to the translation and rotation. Moreover, attributable to the different body sizes of the performers, and the same gesture by different performers will have different coordinate representations.

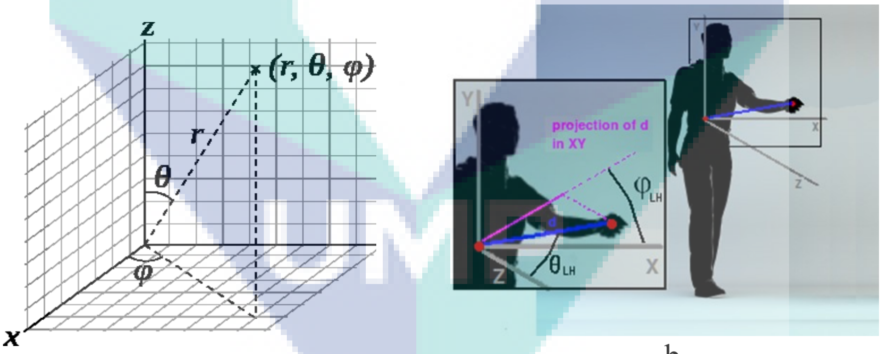
2.4.2 The Normalized Variations of User Size.

The description should be the same no matter if the user is tall or short and translators must be able to produce output the right word in every case. Although the dictionary allows several samples for the same sign (meaning that we can have the same sign described for different size of users), it is almost impossible to add the sample of all size of users in the dictionary.

Instead of directly storing the Cartesian coordinates X,Y, and Z (which can be obtained using OpenNI/NITE), the normalization all the joint coordinates with respect to the Torso position. The position remains always constant throughout the frame signs, will be used is correct, to make the position-invariant systems. Instead of using Cartesian coordinates X, Y, and Z, the spherical coordinates TORSO are stored as the origins. According to Capilla, (2012b), the spherical coordinates are better than Cartesian, therefore in this study uses spherical coordinates.

In mathematics, a spherical coordinate system is a coordinate system for three-dimensional space where the position at a point is specified by three numbers: the radial distance from that point from a fixed origin, its polar angle measured from a fixed zenith direction, and the azimuth angle of its orthogonal projection of a reference plane that passes through the origin and is orthogonal to the zenith, measured from a fixed reference direction within that plane (Hazewinkel, 2001; Weisstein, 2015).

Figure 2.14 (a) show three numbers or values and Figure 2.14 (b) shows the correspondence of these three values in the system.



a b

Figure 2.14. (a). Spherical coordinates (r, θ, φ). (b) Equivalence of these values in the system.

Radial distance r is expressed by d, and d is the vector between torso and concerned joints (θ and φ) are the angle that describes the direction of the vector 3D. The joint set $J = \{EL, ER, HL, HR, WL, WR\}$ and considering T as torso, set the distance $D = \{d_{EL}, d_{ER}, d_{HL}, d_{HR}, d_{WL}, d_{WR}\}$ and set orientations.

$\Theta = \{\theta_{EL,}\theta_{ER,}\theta_{HL,}\theta_{HR,}\theta_{WL,}\theta_{WR}\}$ and $\Phi = \{\varphi_{EL,}\varphi_{ER,}\varphi_{HL,}\varphi_{HR,}\varphi_{WL,}\varphi_{WR}\}$ are follows:

$$\sum_{i=1}^{n} D(i) = \sqrt{(J(i)_x - T_x)^2 + (J(i)_y - T_y)^2 + (T_z - J(i)_z)^2} \tag{2.1}$$

$$\sum_{i=1}^{n} \Theta(i) = atan2\left(\sqrt{(J(i)_x - T_x)^2 + (J(i)_y - T_y)^2}\ , (T_z - J(i)_z)\right) \tag{2.2}$$

$$\sum_{i=1}^{n} \Theta(i) = atan2\left((J(i)_y - T_y), (J(i)_x - T_y)\right) \tag{2.3}$$

where n is the number of joints from J.

2.4.3 Invariant to user's size

In a sign language translator, the system must be able to translate the receipt of the user either tall or short, so that the translator can produce output with the right word in every case.

Figure 2.15 shows the normalization of all the relative distance d by a factor defined by the distance between the Head and Torso joints (d_{HD}). This factor shows the size of the users and all distances D that can be normalized in accordingly with this value.

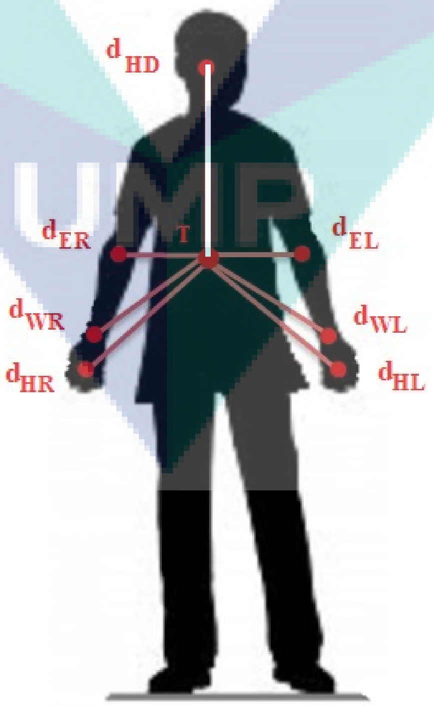


Figure 2.15 Set of distances D sizes

The set of distances $D = \{d_{EL}, d_{ER}, d_{HL}, d_{HR}, d_{WL}, d_{WR}\}$, the distance D_{norm} normalization is obtained as follows Eq. (2.4):

$$\sum_{i=1}^{n} D_{norm}(i) = \frac{D(i)}{d_{HD}} \tag{2.4}$$

where n is the number of the distance D, and d_{HD} is the distance Head-Torso as in Figure 2.15 - line of white. The angles θ and φ do not need to be normalized for expressing direction and the direction remains the same after normalization.

2.5 CLASSIFICATION METHODS

There are many methods used for the classification of sign language, some of which are Data Time Warping (DTW), Hidden Markov Models (HMM) and Artificial Neural Network (ANN).

2.5.1 Dynamic Time Warping (DTW)

Dynamic Time Warping (DTW) was introduced in the 1960s (Bellman and Kalaba, 1959). It is an algorithm for measuring the similarity between two sequences, which may vary in time or speed. For instance, similarities in walking patterns would be detected, even if in one video, the person is walking slowly and if in another video, he or she is walking more quickly, or even if there are accelerations and decelerations during one observation.

DTW has been used in video, audio and graphics applications. In fact, any data that can be turned into a linear representation can be analyzed with DTW (a well- known application has been automatic speech recognition). By using DTW, a computer is able to find an optimal match between two given sequences (i.e., signs) with certain restrictions. The sequences are "warped" non-linearity in the time dimension to determine a measure of their similarity independent of certain non-linear variations in the time dimension. Li and Greenspan, (2007) used compound gesture models, in which the temporal endpoints of a gesture were estimated by DTW, and a bounded search were experiments containing nine different gestures and five subjects. The resulting average recognition rates were 93.30 % for single scale and 88.10 % for multiple scales continuous gestures.

The Microsoft Kinect XBOX360^{TM} is proposed to solve the problem of sign language translation (Capilla, 2012). By using the tracking capability of this RGB-D camera, a meaningful 8-dimensional descriptor for every frame is introduced here. In addition, an efficient Nearest Neighbor DTW and Nearest Group DTW is developed for fast comparison between sign languages. The list of tracked joints at every frame is reduced from 15 to six position of the joints for a dictionary of 14 homemade signs. The introduced system achieved an accuracy of 95.24 %.

Some researchers have applied Dynamic Time Warping (DTW) recognition methods. Iqbal, Purnomo, and Purnama (2011), used DTW for the Indonesian Sign Language based-sensor accelerometer and sensor flex. The experiments were conducted to recognize the 50-word (classes) of Sistem Isyarat Bahasa Indonesia (SIBI). The selected words are only implied by one hand, i.e., the right hand. The results show that the DTW method used can recognize words with an average accuracy of 95.60 %.

Jangyodsuk and Conly, (2014) conducted an American Sign Language (ASL) recognition experiment on Kinect sign data using DTW for sign trajectory similarity and Histogram of Oriented Gradient (HoG) for hand shape representation. The results show an improvement over the original work achieving an 82.00 %.

Celebi et al., (2013) developed gesture recognition using skeleton data and proposed weighted DTW method that weight joints by optimizing a discriminant ratio. They demonstrated the recognition performance of the proposed weighted DTW with respect to the conventional DTW and the state-of-the-art. They have observed that only six out of the 20 joints contribute in identifying a hand gesture: left hand, right hand, left wrist, right wrist, left elbow, right elbow. Thus, the proposed method has an accuracy of 96.70 %.

Iqbal et al., (2011) presented DTW for the Indonesian Sign Language based-sensor accelerometer and sensor flex. The experiments were conducted to recognize the

50-word (classes) of Sistem Isyarat Bahasa Indonesia (SIBI). The results show that the DTW method used can recognize words with an average accuracy is 95.60 %.

The method proposed in Reyes et al., (2011) used DTW costs to be computed between and within class variations to find a weight for each body joint. These weights are global weights in the sense that there is only one weight computed for a body joint. They presented a gesture recognition approach for depth video data based on a novel Feature Weighting approach within the Dynamic Time Warping framework.

The Data Time Warping algorithm was used to compare two signs regardless of their length. By doing so, the system was able to deal with different speeds during the execution of two different samples for the same sign, but sometimes the algorithm can wrongly produce output a positive similarity coefficient.

2.5.2 Hidden Markov Model (HMM)

Hidden Markov Models (HMMs) are learnable finite stochastic automates. They are considered as a specific form of dynamic Bayesian networks. A Hidden Markov Model consists of two stochastic processes. The first stochastic process is a Markov chain that is characterized by states and transition probabilities. The states of the chain are externally not visible, therefore "hidden." The second stochastic process produces emissions observable at each moment, depending on a state-dependent probability distribution (Dymarski, 2011).

HMM is used in robot movement, Bioinformatics, speech and gesture recognition. This model has two advantages for sign recognition: the ability to model linguistic rules and its ability to classify continuous gestures within a certain assumption (Ng and Ranganath, 2002).

Some researchers have applied Hidden Markov Models (HMM) on the recognition methods. Sandjaja and Marcos, (2009) used the HMM for the training and testing phase in Filipino Sign Language numbers. The feature on extraction could track 92.3 % of all objects. The recognizer could also recognize Filipino Sign Language numbers with an average of 85.52 % accuracy.

Wong et al., (2013) presented the development of a software-based Malaysian Sign Language recognition system using Hidden Markov Model. Ninety different gestures were used and tested in this system. Skin segmentation based on YCbCr colour space was implemented in the sign gesture videos to separate the face and hands from the background. Their system has achieved a recognition rate of 72.22 %.

Jiang et al., (2008) presented multilayer architecture in sign language recognition for the signer independent Chinese Sign Language recognition, in which classical DTW, and HMM are combined within an initiative scheme. In the two-stage hierarchy, they defined the confusion sets and introduced the DTW/ISODATA algorithm as the solution to build confusion sets in the vocabulary space. The experiments showed that the multilayer architecture in sign language recognition increases the average recognition time by 94.2 % and the recognition accuracy is 4.66 % more than the HMM-based recognition method.

Vogler Dimitris, (2004) used Parallel Hidden Markov models (PaHMMs) for American Sign Language recognition. He used two channels for the right and left hands, assuming any word can be broken down into fundamental phonemes the same as words in speech recognition. A single channel of the HMM model was tested for a small vocabulary number (22 signs) with the result showing an 87.88 % accuracy rate.

Gaus et al., (2013) presented extraction of suitable feature vector as well as the analysis and performance comparison of the feature vectors using Hidden Markov Model (HMM) for Malaysian Sign Language (MySL). They have tested the system to recognize 112 MSL and found that the union feature vector gives the best recognition rate, which is 83.00 %.

The recognition algorithm of dynamic and combined gestures, which is based on multi-feature fusion was proposed by Liang et al., (2015). Steps of the process are in image segmentation stage. The algorithm extracts the interested regions of gestures in colour and depth map by combining the depth information. Then, to establish a Support Vector Machine (SVM) model for static hand gesture's recognition, the algorithm fuses weighted Hu invariant moments of the depth map into the Histogram of Oriented

Gradients (HOG) of the color image. Finally, a HMM toolbox supporting multi-dimensional continuous data input is adopted to do the training and recognition.

Swee et al., (2007) developed the system that is able to recognize 25 common words signing in Bahasa Isyarat Malaysia (BIM) by using the HMM method. Both hands were involved to perform the BIM with all the sensors connected wirelessly to a PC with a Bluetooth module.

Lang et al., (2011) used the HMM for Sign Language Recognition using Kinect. The sign language recognition framework makes use of Kinect, a depth camera developed by Microsoft and PrimeSense, which features easy extraction of important body parts. The framework also offers an easy way of initializing and training new gestures or signs by performing them several times in front of the camera. The results show a recognition rate of > 97.00 % for eight out of nine signs when they are trained by more than one person.

Auephanwiriyakul et al., (2013) developed an automatic Thai sign language translation system using Scale Invariant Feature Transform (SIFT) and Hidden Markov Models (HMMs). The best correct classification rate for this case is around 74 % on the average.

Nobuhiko Tanibata, (2002) proposed a method to obtain hand features from sequence of images, where a person performs the Japanese Sign Language (JSL) in a complex background to recognize the JSL word. They used a sequence of the hand features as an input to HMM. They made an experiment with real images of a professional JSL interpreter and recognized 65 JSL words successfully.

Zhang et al., (2012) proposed a method based on HMM-Neuro Fuzzy method for the modeling and scoring for Golf-Swing. Kinect is used to capture the 3D skeleton coordination of a golfer while a swing is performed. The results showed that the proposed methods can be implemented to identify and score the golf swing effectively with up to 80 % accuracy rate.

Gaus and Wong, (2012) applied Hidden Markov Model (HMM) to recognize the input gesture. The gesture to be recognized is separately scored against different states of HMMs. They used Kalman Filter to identify the overlapping hand-head or hand-hand region. The model with the highest score indicates the corresponding gesture. In the experiments, they tested the system to recognize 112 Malaysian Sign Language, and the recognition rate was about 83 %.

The recognition algorithm of dynamic and combined gestures, which is based on multi-feature fusion, was proposed by Liang et al., (2015). They used an Hidden Markov Models (HMM) toolbox to support multi-dimensional continuous data input, which was adopted to do the training and recognition. Experimental results show that the algorithm can not only overcome the influence of skin object, multi-object moving and hand gestures interference in the background, but also real-time and practical in Human-Computer interaction. Hidden Markov Models (HMMs) are learnable finite stochastic automates. They are considered as a specific form of dynamic Bayesian networks. Hidden Markov Models has two advantages for sign recognition: the ability to model linguistic rules and its ability to classify continuous gestures within a certain assumption.

2.5.3 Artificial Neural Networks (ANN)

Artificial Neural Network (ANN) is a branch of Artificial Intelligence (AI) and has been accepted as a new computing technology in computer science fields. AI is defined as intelligence exhibited by an artificial entity to solve complex problems, and such a system is generally assumed to be a computer or machine (K. Kumar and Thakur, 2012). The ANN refers to a network or circuit of biological neurons. It is composed of interconnecting neurons that are used to solve biological neural problems or an artificial intelligence problem.

According to Schalkoff (1997), ANN may be defined as structures comprised of densely interconnected adaptive simple processing elements (called artificial neurons or nodes) that are capable of performing massive parallel computations for data processing and knowledge representation.

Although ANNs are drastic abstractions of the biological counterparts, the idea of ANNs is not to replicate the operation of the biological systems but to make use of what is known about the functionality of the biological networks for solving complex problems.

An ANN involves a network of simple processing elements (artificial neurons) which can exhibit complex global behavior, determined by the connections between the processing elements and element parameters. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. In most cases, an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. The utility of Artificial Neural Network models lies on the fact that they can be used to infer a function from observations. This is particularly useful in applications where the complexity to the data or task makes the design of such a function by hand impractical. The tasks to which artificial neural networks are applied are classification, including pattern and sequence recognition, novelty detection and sequential decision making.

According to Principe et al., (2000), Artificial Neural Network (ANN) is an information processing paradigm that is inspired by biological nervous systems, such as the brain. The key element of the paradigm of Artificial Neural Network is a novel structure of the information processing system. It consists of a number of interconnected processing elements (neurons) that are very large, working simultaneously to solve specific problems.

Architecture of Neural Network

The biological neural systems have inspired the creation of the computation process that has performance which is identical with the human nervous system. As with the biological Neural Networks, mathematical model of neural network connects a number of inputs and outputs from an adaptive system which is organized in layers of processing elements interconnected (S. Kumar, 2004).

The basic structure of Artificial Neural is a neuron (as show in Figure 2.16).

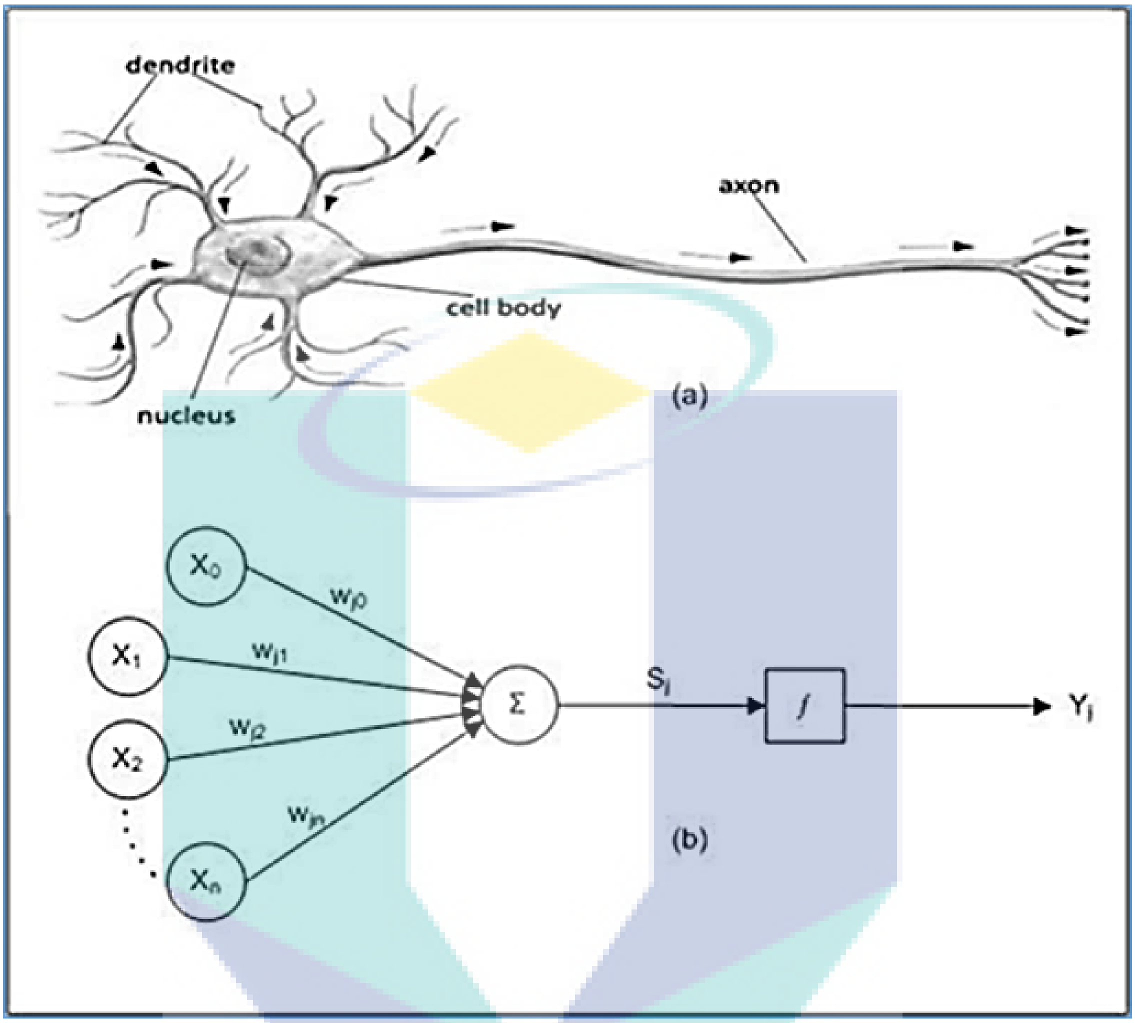


Figure 2.16. The analogy of biological cell and artificial neuron

Source: S. Kumar (2004).

As shown in Figure 2.14(b), the neural model usually consists of:

i. Input (*x*) to receive the signal
ii. Connection weights (*wji*) to store the information
iii. Threshold (*w0*) to set the bias value
iv. Processing element (∑) and activation function (*f*)
v. Output (Yj) present the results of information processing to the next cell

The neuron (as shown in Figure 2.14) can be modelled by the following mathematical equations:

$$s_j(x) = \sum_{i=1}^{n} w_{ji}x_i + w_0x_0 \tag{2.5}$$

$$y_j(x) = f(s_j(x)) \tag{2.6}$$

If $Sj>0$, then $Yj = 1$; if $Sj<0$, then $Yj = 0$; where Yj is the output of the processing element, wi is the input connection weight and xi is the number of input neuron.

This can be of three layers: input, hidden, and output. Input neurons are designated to receive external stimuli that have been presented to the network. Outputs from the network are generated as a signal of output neuron. Hidden neurons compute the intermediate functions and their states are not accessible to the external environment. A layer is connected to the other weight through weight-labelled link. The output of the neuron is typically altered by a transfer function. The most common functions are binary threshold, linear threshold, and sigmoid function and they may differ from neuron to neuron within the network. Neural network is trained by using time series data in order to capture the nonlinear characteristic of the data (Yu et al., 2005).

ANN with a single layer has a limitation in pattern recognition. This limitation can be overcome by adding one or more hidden layers between the input and output layer. Although the use of more than one hidden layer has more benefits for some cases, however it takes a long time to perform the training process. So generally people try a network with one hidden layer first (Siang, 2012).

Backpropagation

Backpropagation trains the network in order to obtain a balance between the ability of the network to recognize the patterns on the training process as well as the network's ability to provide the correct response to the similar input pattern (but not identical) with the pattern that is used during training process.

Figure *2.17* shows the architecture of multi-layer perceptron neural network with n inputs (plus bias), a hidden layer that consists of p units (plus bias), and m output units. vji is the connection weight from the input unit xi to the hidden layer unit zj. vj0 is the weight that connecting the bias from input unit to a hidden layer unit zj). wkj is the connection weight from the hidden layer unit zj to the output yk (wk0 is the weight that connecting the bias from hidden layer to the output zk).

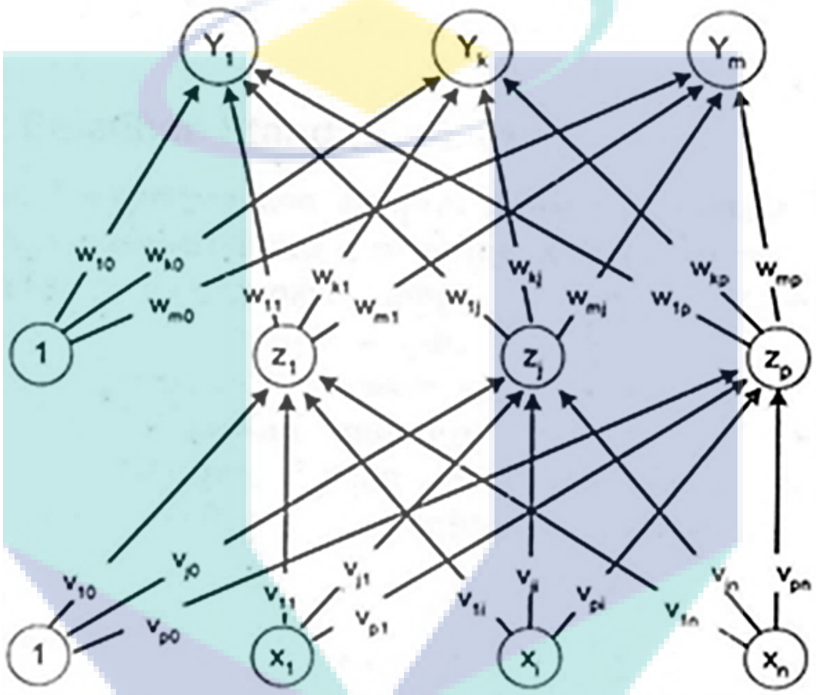


Figure 2.17. The architecture of multilayer perceptron neural network

Source: Siang (2012)

Activation function used in backpropagation must meet several requirements, namely: continuous function, differentiable, and monotonic. The activation functions commonly used is the Log-sigmoid activation function with range of (0,1), as shown in Figure 2.18.

$$f(x)=\frac{1}{1+e^{-x}} \tag{2.7}$$

$$f'(x)=f(x)(1-f(x)) \tag{2.8}$$

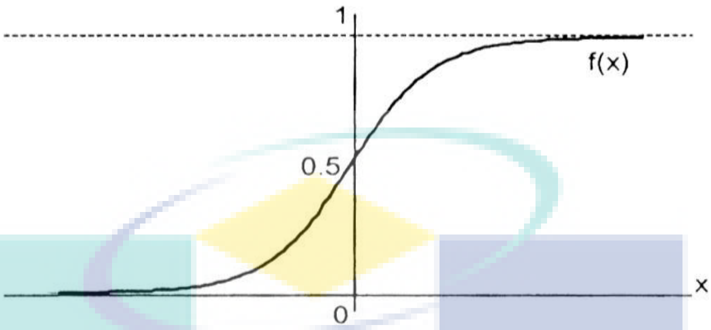


Figure 2.18 Log-sigmoid activation function

Source: Siang (2012)

Another function that is commonly used is the Hyperbolic-Tangent Sigmoid function which is similar to the log-sigmoid function, but with the range [-1,1].

$$f(x)=\frac{2}{1+e^{-x}}-1 \tag{2.9}$$

$$f'(x)=\frac{(1+f(x))(1-f(x))}{2} \tag{2.10}$$

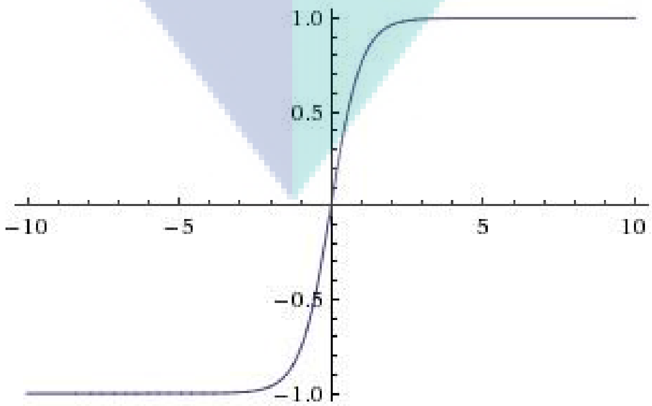


Figure 2.19. Hyperbolic tangent sigmoid activation function

Source: http://cs231n.github.io/neural-networks-1

The graph function is shown in Figure 2.19. Log-sigmoid function has a maximum value = 1. Then for the targets > 1, the inputs and outputs must be transformed first so all patterns have the same range as the activation function that is used. The alternative way is the sigmoid activation function is only used in the layer which is not the output layer. The activation function that is used in the output layer has an identity function: $f(x) = x$.

Backpropagation is a supervised learning technique used for training artificial neural networks. It was first described by Paul Werbos in 1974, and further developed by David E. Rumelhart, Geoffrey E. Hinton and Ronald J. Williams in 1986.

Supervised learning can be illustrated through the block diagram of Figure 2.20(a) and has as objective in minimization of the mean square error E(t), given in (2.9), where the index t represents the number of training epochs (one complete presentation of all training examples, n = 1, 2,…, N, where N is the total number of examples, called an epoch) (Temel, 2010).

$$E(t) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}e(n)^2 \quad (2.11)$$

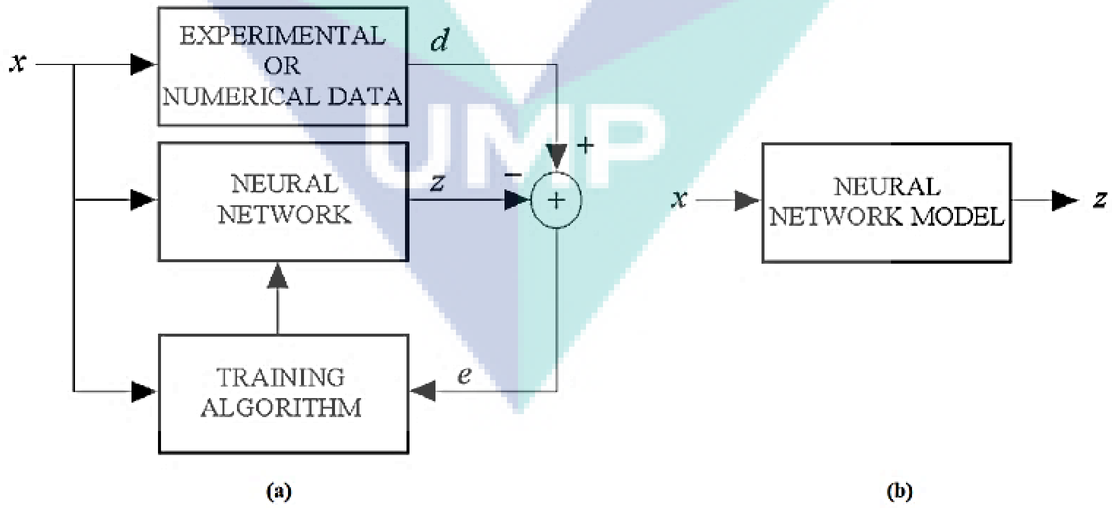


Figure 2.20. (a) Block diagram of supervised learning; (b) neural network "black box" model

Source: Temel (2010)

It is most useful for feed-forward networks (networks that have no feedback, or simply, that have no connections that loop). The term is an abbreviation for "backwards propagation of errors". Backpropagation Many researchers highlight the success of using neural networks in sign language recognition. An Artificial Neural Network (ANN) consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation.

Backpropagation algorithm can be divided into two process, training algorithm and simulation. The training process includes 3 phases. The first phase is the forward phase. The input pattern is calculated forward from the input layer to the output layer using the specified activation function. The second phase is a backward phase. The difference between the outputs of the network with the desired target is the error that has occurred. Then the errors are propagated backwards, starting from the connection that relates directly to the units in the output layer. The third phase is modification to reduce the weight of the error.

The three phases are repeated constantly until the termination condition is met. Generally, the termination condition that is commonly used is the number of iterations (epoch) or error (one complete presentation of all training examples, n = 1, 2,…, N, where N is the total number of examples, called an epoch) (Silva etal., 2010b). Iteration will be terminated if the number of iterations performed has exceeded the maximum number of iterations that has been specified, or if the error occurred smaller than the tolerance limits which is allowed. The simulation process only used forward phase. The training algorithm by using standard architecture of Back-propagation is presented below:

Training Algorithm:

0. Initialize weights. Determine learning rate (α).
1. Determine the value of error tolerance or threshold value (if use error tolerance as termination condition); or set the number of iteration/epoch (if use epoch as termination condition).
2. While stopping condition is not met do step 2 to step 9.
3. For each pair of training patterns, do step 3-to-8.

Forward phase

4. For each input unit x_i (from the first unit to unit n in the input layer) sends the input signal to all units on the next layer (to hidden layer).
5. For each unit in the hidden layer z_j (from the first unit to unit p; i = 1, ..., n; j = 1,..., p), the outputs are calculated by applying the Log-sigmoid activation function to sum the weighted input x_i, then send to next layer.

$$z_j = z_{jo} \sum_{i=1}^{n} x_i x_{ji} \tag{2.12}$$

$$f(z) = \frac{1}{1 + e^{-z_i}} \tag{2.13}$$

6. For each unit in the output layer y_k (from the first unit to unit m; j=1,…, p; k=1,…, m) the outputs are calculated by applying the activation function to sum the weighted input z_j.

$$y_k = zw_{ko} \sum_{j=1}^{p} z_j w_{kj} \tag{2.14}$$

$$f(y) = \frac{1}{1 + e^{-y_k}} \tag{2.15}$$

Backward phase

7. For each output y_k (from the first unit to unit m; j=1,…,p; k=1,…,m) calculate the error of output layer (δk) by applying target t_k. send δ_k to the previous layer and calculate weights and bias (Δwjk and Δw0k) between output layer and hidden layer:

$$\delta_k = (t_k - y_k) f'(y_k) = (t_k - y_k) y_k (1 - y_k) \tag{2.16}$$

calculate weights Δwjk with learning rate α.

$$\Delta w_{jk} = \alpha\delta_k z_j \tag{2.17}$$

$$\Delta w_{0k} = \alpha\delta_k \tag{2.18}$$

8. For each unit in the hidden layer (from the first unit to unit p; i=1,…,n; j=1,…,p; k=1,…,m) calculate the error of hidden layer (δ_k). Then calculate weights and bias (Δv_{ij} and Δv_{0j}) between hidden layer and input layer:

$$\delta_j = \left(\sum_{k=1}^{m} \delta_k\, w_{jk}\right) f'\left(w_{jk} + \sum_{i=1}^{m} x_i v_{ij}\right) \tag{2.19}$$

$$\Delta v_{ij} = \alpha\delta_j z_i \, \Delta v_{0j} = \alpha\delta_j \tag{2.20}$$

<u>Modification Weights Phase</u>

9. For each output unit y_k (from the first unit to unit m) update the weights and bias (j=0,…,p; k=1,…,m;) and the new weights and bias is:

$$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk} \tag{2.21}$$

from the first unit to unit p in the hidden layer update the weights and bias (i=0,…, n; j=1,…, p):

$$v_{jk}(new) = v_{0j}(old) + \Delta w_{ij} \tag{2.22}$$

10. Check the stopping condition.

There are some researchers who had applied ANN method of the sign language recognition such as: Hasan and Mishra, (2010), they used a Multilayer Perceptron (MLP) neural network to recognize static hand-finger gestures of the yubimoji, the Japanese Sign Language syllabary. Signal inputs from the data glove interface were taken separately for each static yubimoji gesture. Each input was fed as input of MLP, after the network was trained 10 times and tested for 41 gestures. Generally, only 18 of the static gestures were successfully recognized. One of the reasons was attributed to the data glove's inability to measure gesture directions, particularly yubimoji gestures with similar finger configurations.

Karami et al., (2011) used Wavelet transforms and ANN for Persian Sign Language (PSL) recognition. The system was implemented and tested using a data set of 640 samples of Persian sign images, 20 images for each sign. The experimental results show that the system can recognize 32 selected PSL alphabets with an average classification accuracy of 94.06 %.

Munib et al., (2007) developed a system for automatic translation of static gestures of alphabets and signs in American Sign Language and used Hough transform and neural networks, which are trained to recognize signs. Experiments revealed that the system was able to recognize selected ASL signs with an accuracy of 92.30 %.

Adithya et al., (2013) presented an ANN based method for automatically recognizing the fingerspelling in Indian sign language. The signs are identified by the features extracted from the hand shapes. They used skin colour based segmentation for extracting the hand region from the image. The results show that the system produced a recognition rate of 91.11 %.

Sharma et al., (2014) presented a system for automatic recognition of Indian sign language of numeric signs using neural network and K-Nearest Neighbor (kNN) classification techniques. The result from these experiments achieved up to 97.10 % accuracy.

Dogic and Karli (2013) developed a system for the Bosnian sign language with the use of digital image processing methods providing a system that teaches a multilayer neural network using a back-propagation algorithm. Training was done using cross validation method for better performance thus, an accuracy of 84.00 % was achieved.

Sharma and Garg (2014) presented a technique which begins by conversion of coloured image into gray scale image, binarization, edge detection and segmentation using morphological operations like filtering, thinning and dilation for comparing the Back-propagation Neural Network with Support Vector Machine (SVM). Database used comprises of real hand images captured using a camera. Training database comprises of 30 images for each alphabet making a total of 780 images. Features used are area, extent, eccentricity, centroid and orientation. The accuracy using Backpropagation Neural Network came out to be 96.00 %-97.00 % and accuracy using SVM came out to be 92.00 %-93.00 %.

A Neural Network based method for automatically recognizing the hand gestures of Indian sign language is presented by Tavari and Deorankar, (2014). It has been experimented with gesture images captured by a web camera and achieved the satisfactory results with accuracy of 91.66 %.

Maraqa et al., (2008) used two recurrent neural network's architectures for static hand gestures to recognize the Arabic Sign Language (ArSL). Elman (partially) Recurrent Neural Networks and fully Recurrent Neural Networks were used. A digital camera and a coloured glove were used for input image data. For the segmentation process, the colour model was used. Segmentation divides the image into six color layers, five for fingertips and one for the wrist. Thirty features are extracted and grouped to represent a single image, expressed by the fingertips and the wrist with angles and distances between them. This input feature vector is the input to both Neural Network systems. A total of 900 coloured images were used for the training set and 300 coloured images for testing purposes. The results showed that the fully recurrent neural network system (with recognition rate of 95.11 %) is better than the Elman neural network (89.67 % recognition rate).

Asriani and Susilawati (2010) used the Backpropagation Neural Networks (BNN) method for the Indonesian Sign Language (ISL) recognition. The success rate for static hand gesture recognition achieved in this study was 69.00 %.

Lungociu (2011) proposed a Neural Network based approach to the sign language recognition. An experiment was provided and directions to further improve his work were also emphasized. The results showed that the system produced a recognition rate of 80.00 %.

Admasu and Raimond (2010) used the Gabor Filter (GF) together with Principal Component Analysis (PCA) for extracting features from the digital images of hand gestures for the Ethiopian Sign Language (ESL), while the ANN was used for recognizing the ESL from extracted features and translation into Amharic voice. The experimental results showed that the system produced a recognition rate of 98.53 %.

Hninn and Maung (2009) used real time 2D hand tracking to recognize hand gestures for the Myanmar Alphabet Language. Digitized photograph images were used as input images and the Adobe Photoshop filter was applied for finding the edges of the image. By employing histograms of local orientation, this orientation histogram was used as a feature vector. MATLAB toolbox was used for system implementation. Experiment results showed that the system can achieve a 90.00 % recognition average rate.

Bailador et al., (2007) presented a Continuous Time Recurrent Neural Networks (CTRNN) real time hand gesture recognition system using a tri-axial accelerometer sensor and wireless mouse to capture the 8 gestures used. The work was based on the idea of creating specialized signal predictors for each gesture class, in which a standard Genetic Algorithm (GA) was used to represent the neuron parameters. Each genetic string represents the parameter of a CTRNN. Two data sets were applied; one for isolated gestures, with a recognition rate of 98.00 % for the training set and 94.00 % of the testing set. For the second data set, for capturing gestures in a real environment, the recognition rate was 80.50 % for training and 63.60 % for testing.

Stergiopoulou and Papamarkos, (2009) conducted a study on the static hand gesture recognition based on Neural Gas Self-Growing and Self-Organized (SGONG) network. An input image using a digital camera for the detection of the hand area of YCbCr color space was applied, and the threshold technique was used to detect skin tones. They used the competitive Hebbian learning algorithm, which begins studying with two neurons. As the neurons grow, the grid will detect the exact shape in the hand, with the specified number of fingers raised, however, in some cases the algorithm might lead to false classification. This problem is solved by applying the average finger length ratio. This method has the disadvantage that two fingers may be classified into the same class. This problem has been overcome by choosing the most likely combinations of fingers.

This system can recognize the 31 movements that have been established with a recognition rate of 90.45 % and 1.5 sec.

Paulraj et al., (2009) presented a simple method for translating Kod Tangan Bahasa Melayu (KTBM) into a voice signal based on subject head and two-hand gestures. Different gesture signs made by different subjects are captured using a USB web camera in the RGB video stream format with a screen bit depth of 24 bits and a resolution of 320 X 240 pixels. Experimental results demonstrated that the recognition rate of the proposed neural network models was about 81.07 %.

Nguyen et al., (2015) proposed Principal Component Analysis (PCA) and an ANN method for recognition. The focus of their research is the resizing method which can classify different gestures, and applying PCA for feature extraction, with low computational cost features for identification. The experimental results showed that the system produced a recognition rate of 94.30 %.

Tang et al., (2015) applied Deep Neural Networks (DNNs) to automatically learn features from hand posture images that are insensitive to movement, scaling, and rotation. They proposed a two-stage HPR system for Sign Language Recognition using a Kinect sensor. Experiments verified that the proposed system worked quickly and accurately and achieved a recognition accuracy as high as 98.12 %.

Sakhalkar et al., (2014) presented a method for gesture recognition using neural networks for American Sign Language. The image acquisition was done with the help of skeletal tracking using a Kinect camera. They used the information about the seven joints. The efficiency in the system was found out to be 85.71 %.

Akmeliawati et al., (2007) presented an automatic visual-based sign language translation system. They proposed automatic sign language translator that provides a real-time English translation of the Malaysian Sign Language. The Sign Language Translator can recognize both finger spelling and sign gestures that involve static and motion signs. The trained Neural Networks were used to identify the signs to translate into English.

To summarize, some of the sign language recognition methods shows a comparison of the methods that have been applied by researchers in sign language recognition.

Table 2.1 Comparison between the existing method for Sign Language Recognition

No	Classification Method	Author	No of Data Set	Type of Data Set	Capture Method	Accuracy
1.	Feature extraction from 2D gesture	(Y. F. B. A. Gaus et al., 2011)	112 signs	Malaysian Sign Language	Skin segmentation YCbCr colour space	80.00 %
2.	Artificial Neural Networks (NN)	(Akmeliawati et al., 2007).	59 samples	Malaysian sign language	Video Camera	90.00 %
3.	Artificial Neural Networks (NN)	(Paulraj et al., 2009)	14 signs, 140 samples	Kod Tangan Bahasa Melayu (KTBM) - Hand	USB web camera	81.07 %
4.	Hidden Markov Models (HMM)	(Swee et al., 2007).	25 words	Malay Sign Language	Data Glove	-
5.	Artificial Neural Networks (NN)	(Bailador et al. (2007)	8 signs		Tri-Axial Acceleration Sensor	94.00 %
6.	Artificial Neural Networks (NN)	(Maraqa et al., 2008)	30 samples	Arabic Sign Language	Digital camera	95.00 %
7.	*Artificial Neural Networks (NN)*	(Maung, 2009).	33 samples	Myanmar Sign Language	Digital camera	90.00 %
8.	Artificial Neural Networks (NN)	(Admasu and Raimond, 2010).	34 signs, 170 samples	Ethiopian Sign Language	Digital camera	98.53 %.
9.	Artificial Neural Networks (NN)	(Lungociu, 2011)	14 character	Recognized characters	web camera,	80.00 %.
10.	Wavelet transform and Artificial Neural Networks (NN)	(Karami et al., 2011).	10 gesture	Persian Sign Language	Digital camera	90.45 %
11.	Hidden Markov Models (HMM)	(Lang et al., 2012).	9 signs	America Sign Language	Kinect Depth Camera	97.00 %
12.	HMM and Neuro-Fuzzy	(Zhang et al. 2012).	200 images	Postures Of Golf Swing	Kinect	80.00 %
13.	Dynamic Time Warping (DTW)	(Capilla, 2012)	14 sign, 225 samples	America Sign Language	Kinect (six joints)	95.24 %.
14.	Dynamic Time Warping (DTW)	(Celebi et al., 2013)	8 gesture 28 samples	Right Hand Push Up and Left Hand	Kinect (six joints)	96.70 %

A comparison of the advantages and disadvantages of sign language recognition methods has been made between ANN and HMM. It has been found that different ANN systems are used at different stages of recognition systems according to the nature to the problem, its complexity and the environment available. Additionally, since true human gestures are continuous, introducing an isolated system can significantly disrupt the natural flow of human interaction, and it does not have as much value in the reality of sign recognition. The success of a fully automated sign recognition system relies on solving current problems associated with continuous gesture recognition. HMM classifier is also proven to be interesting for sign recognition due to its ability to model words based on sets of predefined states.

The researchers have used various methods for the sign language recognition such as Hidden Markov Models (HMM), Data Time Warping (DTW), Artificial Neural Network (ANN). The highest rate of accuracy 98.53 % was achieved by Admasu and Raimond (2010), with the ANN method.

2.6 FEATURE SELECTION

According Yu and Liu, (2003), Feature Selection is frequently used as a preprocessing step to machine learning, wherein a subset of the features available from the data are selected for application of a learning algorithm. It is a process of choosing a subset of original features so that the feature space is optimally reduced according to a certain evaluation criterion. The best subset contains the least number of dimensions that most contribute to accuracy, to discard the remaining, unimportant dimensions (Isabelle Guyon and Elisseeff, 2003).

Selection of attributes subset-based is algorithms that perform an analysis of the subset of attributes that are generated, either at random or by genetic algorithms (Goldberg, 1989) or be greedy, or known as the Best First Search (Mark a Hall, 2000).

2.6.1 Feature Selection Process

Feature selection is the combination of the two operational components: evaluation of feature subsets; and searching for feature subsets. During the feature subset search, features are evaluated and assigned a value of evaluation measure; in heuristic search, this measure is evaluated to guide the search further or to terminate the search. When the feature selection process is completed, the evaluation measure values are assessed and the best evaluated feature is selected (Scully and Jensen, 2011).

a. Feature Selection – Evaluation

The task of feature evaluation is to assess the quality of a feature. A feature is commonly a subset of conditional attributes. A feature subset can be a singleton (single element), a subset (multiple elements), the full set of conditional attributes, or the empty set. The approaches toward evaluating singletons and multielement subsets are the cause for much research effort, giving differing algorithm methodologies, measures and performance results.

Feature evaluation types can be grouped into the following categories (Scully and Jensen, 2011):

- Instance-based Evaluation

Problems with Instance Based Evaluation are Search spans on both attribute and instance space. This can lead to larger time and space complexity over attribute-based methods

- Attribute-based Evaluation
    - Single or Singleton Attribute Evaluation (Univariate)
    - Multiple Element or Subset Attribute Evaluation (Multivariate).

Singleton attribute evaluation methods measure individual attributes with or without consideration of the decision attributes. The project explored information-based approaches in extension of rough-set subset evaluation approaches.

Singleton is unable to determine subset dependency, which limits its potential effectiveness on datasets showing conjunctive rule associations. However, information theory based measures seem to be effective otherwise. The reduced attribute search space, in comparison to subset evaluation, means a less highly time complex search process is necessary (Scully and Jensen, 2011).

- Subset Attribute Evaluation

Subset attribute evaluation methods that evaluate the subsets of attributes most commonly with respect to the decision attribute. The quantity of possible subsets to be evaluated is the power-set of the conditional attributes, unification the decision attribute to each of those subsets. Problems with Subset Attribute Evaluation: the dependency evaluation is commonly used in subset evaluation. It carries an increase in time complexity over singleton attribute evaluation in its increase feature space. This increase in feature space can also result in an increase in space complexity in the tracking of feature subsets evaluations (Scully and Jensen, 2011).

Aggarwal and Amrita, (2013) divided the feature selection methods into three categories named filter, wrapper and hybrid (embedded) method.

b. Filter methods

According to Yu et al. (2003), Dash et al. (2002) filter methods are based on performance evaluation functions calculated directly from the training data such as distance, information, dependency, and consistency, correlation and select features subsets without involving any learning algorithm.

- Consistency-based Subset Evaluation (CSE).

Evaluates the worth of a subset of attributes by the level of consistency in the class values when the training instances are projected onto the subset of attributes. Consistency of any subset can never be lower than that of the full set of attributes, hence the usual practice is to use this subset evaluator in conjunction with a Random or Exhaustive search which looks for the smallest subset with consistency equal to that of the full set of attributes (Liu and Setiono, 1996).

Several approaches to attribute subset selection use class consistency as an evaluation metric (Dash and Liu, 2003). These methods look for combinations of attributes whose values divide the data into subsets containing a strong single class majority. Usually the search is biased in favor of small feature subsets with high class consistency. The consistency-based subset evaluator uses Liu and Setiono's (Dash and Liu, 2003) consistency metric:

$$\text{Consistency}_s = 1 - \frac{\sum_{i=0}^{J} |D_i| - |M_i|}{N} \tag{2.23}$$

where s is an attribute subset, J is the number of distinct combinations of attribute values for s, $|D_i|$ is the number of occurrences of the ith attribute value combination, $|M_i|$ is the cardinality of the majority class for the ith attribute value combination and N is the total number of instances in the data set. Consistency-based Subset Evaluation Algorithm shown in Figure 2.21.

```
Input: Data D, feature set S
Output: Consistent Feature Subset Evaluation
1. δ = inConCal(S,D)
2. T = S
/* subset generation */
3. For all feature f in S
4. Sj = S − f /* remove one feature at a time */
5. j ++
6. For all Sj
7. if (Sj is legitimate Λ inConCal(Sj,D) _ δ)
8. if inConCal(Sj ,D) < inConCal(T ,D)
9. T = Sj
/* recursion */
10. A (Sj , D)
```

Figure 2.21 Consistency-based Subset Evaluation Algorithm

Source: Dash and Liu (2003)

It starts with the full set of features S^0, removes one feature from S_j^{l-1} in turn to generate subsets S_j^l where l is the current level and j specifies different subsets at the lth level. If $\mathrm{U}(S_j^l) > \mathrm{U}(S_j^{l-1})$, S_j^l stops growing (its branch is pruned); otherwise, it grows to level $l + 1$, i.e., one more feature could be removed.

The legitimacy test is based on whether a node (subset) is a child node of a pruned node. A node is illegitimate if it is a child node of a pruned one (which is already found to be illegitimate). Each node is represented by a binary vector where 1's stand for presence of a particular feature in that subset and 0's for its absence. The test is done by checking the distance between the child node under consideration and pruned nodes. If the distance with any pruned node is 1 (i.e., the difference of the two representative binary vectors is 1), the child node is the child of the pruned node. Notice that by this way at every level we are able to determine all the illegitimate nodes.

Lei and Govindaraju (2005) presented a comparative study of features commonly used in on-line signature verification. A consistency model is developed by generalizing the existing feature-based measure to distance-based measure. Experimental results showed that the simple features like X, Y coordinates, the speed of writing and the angle with the X-axis are amongst the most consistent.

Data sets with numeric attributes are first discretized using the method of Fayyad and Irani (Tan LIM and LAI, 1993). The modified forward selection search described at the start of this section is used to produce a list of attributes, ranked according to their overall contribution to the consistency of the attribute set.

- Correlation-based Feature Selection (CFS).

According to Hall and Smith (1999) and Hall (2000), Correlation-based Feature Selection (CFS) is among the first methods that evaluates subsets of attributes rather than individual attributes. At the heart of the algorithm is a subset evaluation heuristic that takes into account the usefulness of individual features for predicting the class along with the level of inter-correlation among them. The heuristic (Eq. (2.24)) assigns high scores to subsets containing attributes that are highly correlated with the class and have low inter-correlation with each other.

$$\text{Merit}_s = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k+1)\,\overline{r_{ff}}}} \tag{2.24}$$

where Merit S is the heuristic "Merit" of a feature subset s containing k features, $\overline{r_{cf}}$ is the average feature-class correlation, and $\overline{r_{ff}}$ is the average feature inter-correlation. The numerator can be thought of as giving an indication of how predictive a group of features are, the denominator shows of how much redundancy there is among them. The heuristic handles irrelevant features as they will be poor predictors of the class. Redundant attributes are discriminated against as they will be highly correlated with one or more of the other features. Due to independent treatment of attributes, CFS cannot identify the interacting features strongly such as in a parity problem. However, it has been shown that it can identify useful attributes under moderate levels of interaction (Mark Hall and Smith, 1999).

Correlation-based Subset Evaluation (CfsSubsetEval):

Evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them. Subsets of features are highly correlated with the class while having low intercorrelation are preferred (M. Hall, 1999).

In general, a feature/attribute is good if it is relevant to the class concept but is not redundant to any of the other relevant features. If adopted the correlation between two variables as a goodness measure, the above definition becomes that of a feature is good if it is highly correlated to the class but not highly correlated to any of the other features. In other words, if the correlation between a feature/attribute and the class is high enough to make it relevant to (or predictive of) the class and the correlation between it and any other relevant features/attributes does not reach a level so that it can be predicted by any of the other relevant features/attributes, it will be regarded as a good feature/attribute for the classification task. The problem of attribute selection requires a suitable measure of correlations between attributes and a sound procedure to select attributes based on this measure. There exist broadly two approaches to measure the correlation between two random variables. One is based on classical linear correlation and the other is based on information theory (Yu and Liu, 2003).

The algorithm of Correlation-based Feature Selection (CFS) mainly consists of two parts for achieving the goal of reducing dimensionality of the original feature space (as shown in Figure 2.22).

In the first part (line 1- 4), the algorithm removes irrelevant features with poor prediction ability to target class. In the second part (line 5-9), the algorithm eliminates redundant features that are inter-correlated with one of more other features.

Finally, the remaining selected features are significant features that contain indispensable information about the original feature set. Given a data set with a number of input features and a target class, the algorithm firstly calculates the mutual information between features and class.

The algorithm then ranks the features in descending order according to their degrees of association to the target class. Once the importance of the input features are ranked, these terms whose information measure are greater than zero are kept; which means those removed features are totally irrelevant to target class and the remaining ones are predictive.

In the second part, it starts with calculating the intercorrelated strengths of each pair of features. The total amount of mutual information for each feature is acquired by adding all mutual information measures together that relate to that feature. For adjusting the discriminative power of mutual information performed on feature-to-feature and feature-to-class to the same level, the factor w and its value is equal to the mean of summation of feature-to-class information divided by the mean of summation of feature-to-feature information. By multiplying w to each feature-to-class measure, both feature-toclass and feature-to-feature reach to the same important rank.

Finally, the differences of them are computed and features values are greater than zero, which means the selected features are the most “significant features” that restrain indispensable information of the original feature space.

```
1   // Remove irrelevant features
2   Input original data set D that includes features X and target class Y
3   For each feature Xi
    Calculate mutual information SU(Y; Xi)
4   Sort SU(Y; Xi) in descending order
5   Put Xj whose SU(Y; Xi) > 0 into relevant feature set RXY
6   // Remove redundant features
7   Input relevant feature set RXY
8   For each feature Xj
    Calculate pairwise mutual information
    SU(Xj; Xk) ∀j ≠ k
9   SXY = Σ (SU(Xj; Xk))
10  Calculate means μR and μS of RXY and SXY ,
    respectively. w = μS /μR
11  R = w·RXY - SXY
12  Select Xj whose R > 0 into final set F
```

Figure 2.22. Correlation-based Feature Selection Algorithm

Source: Chou Yen Luo Pissinou and Makki (2007)

Correlation-based Attribute Evalualtion (CorrelationAttributeEval)

Evaluates the worth of an attribute by measuring the correlation (Pearson's) between it and the class. Nominal attributes are considered on a value by value basis by treating each value as an indicator. An overall correlation for a nominal attribute is arrived at via a weighted average (Mark a Hall, 2000).

Attribute subset selection on the basis of relevance analysis is one way to reduce the dimensionality. Relevance analysis of attribute is done by means of correlation analysis, which detects the attributes (redundant) that do not have significant contribution in the characteristics of whole data of concern. After which the redundant attribute or attributes strongly correlated to some other attribute is disqualified (Tiwari, 2010).

There are four basic steps in a typical feature selection method (as show in Figure 2.23):

1. a generation procedure to generate the next candidate subset for evaluation,
2. an evaluation function to evaluate the candidate subset,
3. a stopping criterion to decide when to stop, and

4. a validation procedure to check whether the subset is valid.

The generation procedure uses a search strategy to generate subsets of features for evaluation. It starts (i) with no features, (ii) with all features, or (iii) with a random subset of features. In the first two cases features are iteratively added/removed, whereas in the last case, features are either iteratively added/removed or produced randomly thereafter during search. An evaluation function measures the goodness of a subset produced by some generation procedure, and this value is compared with the previous best. If it is found to be better, then it replaces the previous best subset. An optimal subset is always relative to a certain evaluation function (i.e., an optimal subset chosen using one evaluation function may not be the same as that using another evaluation function). Without a suitable stopping criterion the feature selection process may run unnecessarily long or possibly forever depending on search strategy. Generation procedures and evaluation functions can influence the choice for a stopping criterion. Examples of stopping criteria based on a generation procedure include: (i) whether a predefined number of features are selected, and (ii) whether a predefined number of iterations reached. Examples of stopping criteria based on an evaluation function include: (i) whether further addition (or deletion) of any feature produces a better subset, and (ii) whether an optimal subset (according to some evaluation function) is obtained. The feature selection process halts by outputting the selected subset of features which is then validated.

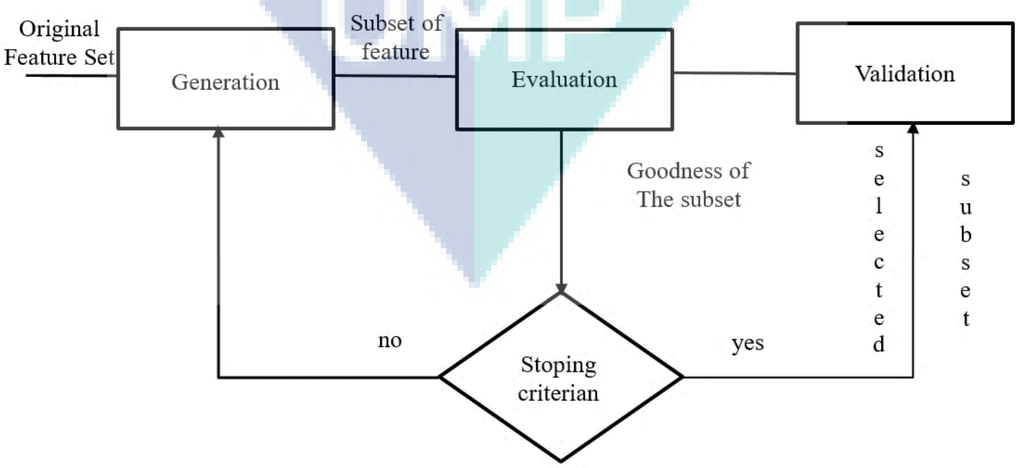


Figure 2.23. The flowchart Feature Selection Algorithm

Source: Dash and Liu (1997)

Sakar et al., (2012) proposed a method called KCCAmRMR (Kernel Canonical Correlation Analysis based minimum Redundancy–Maximum Relevance) which explores and uses all the correlated functions (covariates) between variables to compute their unique (conditional) information about the target. The experimental results show that KCCAmRMR method can choose better set of features than mRMR and opens a promising alternative way in feature selection using the renowned CCA and kernel methods.

Santiago-Ramírez et al., (2012) proposed a strategy for the recognition and another for face tracking, both are based on correlation filters. Correlation filter is constructed by averaging the Fourier transforms of the training images with non-linearity factor k, which emphasizes the common features of the training images. This strategy achieved approximately 95 % effectiveness in recognition.

Monirul Kabir et al. (2010) presented a new Feature Selection (FS) algorithm based on the wrapper approach using Neural Networks (NNs). The algorithm used a constructive approach involving correlation information in selecting features and determining NN architectures.

c. Wrapper methods

Kohavi et al. (1997a) required one predetermined learning algorithm and used its estimated performance as the evaluation criterion. Generally, the wrapper method achieves better performance than the filter method, but tends to be more computationally expensive than the filter approach. Also, the wrappers yield feature subsets optimized for the given learning algorithm only the same subset may thus be bad in another context.

d. Hybrid approach

Das, (2001); Haindl, Somol, Ververidis and Kotropoulos, (2006); Sebban and Nock, (2013) combined the advantages of more than one of the listed approaches. Hybrid algorithms have recently been proposed to deal with high dimensional data. These algorithms mainly focus on combining filter and wrapper algorithms to achieve best possible performance with a particular learning algorithm with the time complexity comparable to that of the filter algorithms.

2.6.2 Search Method on Feature Selection

The task of search in feature selection is to select features for evaluation until a stopping criterion is met. Stopping criterion might be when evaluation has reached its goal state or the search has exhausted all possible features (Scully and Jensen, 2011). These methods search the set of all possible features in order to find the best set of features. Three search methods which includes BestFirst, GreedyStepwise and Ranker (Jantawan and Tsai, 2014).

- BestFirst

This searches the space of attribute subsets by greedy hill climbing augmented with a backtracking facility. Setting the number of consecutive non-improving nodes allowed the controls of the level of backtracking done.

The BestFirst search starts with an empty set of features and generates all possible single feature expansions. The subset with the highest evaluation is chosen and expanded in the same manner by adding single features. If expanding a subset results in no improvement, the search drops back to the next-best unexpanded subset and continues from there. Given enough time a best first search will explore the entire feature subset space, so it is common to limit the number of subsets expanded that result in no improvement.

Best first search was used in the final experiments as it gave slightly better results for some cases than hill climbing. Figure 2.24 presents the algorithm of the search subset using the Best First.

1. *Begin with the OPEN list containing the start state, the CLOSED list empty, and BEST □□start state.*
2. *Let s = arg max e(x) (get the state from OPEN with the highest evaluation)*
3. *Remove s from OPEN and add to CLOSED*
4. *If e(s) ≥ e (BEST), then BEST ← s*
5. *For each child t of s that is not in the OPEN or CLOSED list, evaluate and add to OPEN*
6. *If BEST changed in the last set of expansions, goto 2*
7. *Return BEST*

Figure 2.24. Algorithm BestFirst search method

Source: Mark a Hall (2000)

- GreedyStepwise

Performing a greedy forward or backward search through the space of attribute subsets. May start with no/all attributes or from an arbitrary point in the space. Stops when the addition/deletion of any remaining attributes results in a decrease in evaluation. Can also produce a ranked list of attributes by traversing the space from one side to the other and recording the order that attributes are selected (T. Zhang, 2011).

- Ranker

Ranker method is ranked attributes by their individual evaluations Use in conjunction with attribute evaluators (ReliefF, Gain Ratio, Entropy etc.) with the parameter generate ranking (true or false), number to select, and thres hold values is set threshold by which attributes can be discarded. Default value results in no attributes are discarded. Use either this option or number to select to reduce the attribute set (Dinakaran and Thangaiah, 2013).

In general, the subset search method is a method of searching a subset of attributes in the hill climbing technique plus the backtracking. A specified parameter to set how many nodes can be increased in a row, which is used to control the level of backtracking.

Xu and Yang (2006) have proposed feature selection in the classification process in SLR. They proposed Hill Climbing approach and Random Walk approach, to select the features. They claimed that both algorithms were easy-implemented but reasonable and efficient. Therefore, there are opportunities, the feature selection can be used in SLR

2.7 DISCUSSION

There are several methods to acquire data in image acquisition such as the data glove, video using RGB sensor, using the attached sensors, the joint angles and spatial positions of the hand can be measured directly from the glove. However, the data glove and its attached wires are still hassle and rigid for users to wear. The use of video RGB sensor is simpler than the data glove but video RBG is unable to provide information from users, such as the position and size of the user. In addition, the accuracy of the results using RGB sensor video is more influenced by the complex background, lighting variations, and other skin color with hand objects, in addition to the system requirements such as speed, recognition time, robustness and computational efficiency.

There is image acquisition method using the camera with RGB and depth sensor. This is based on the existing advantages such as: the RGB and depth cameras, in general, are well suited for sign language recognition. It offers 3D data from the environment without a complicated camera setup and efficiently extracts the user's body parts, allowing for recognition of not just hands and head, but also other parts such as elbows that can be of further help in distinguishing between similar signs. Another advantage is the independency of lighting conditions, as the camera uses infrared light. All body parts are detected equally well in a dark environment. The technology that uses the RGB camera and depth sensor such as the Kinect sensor. Althought, the cost of kinect sensor is still too expensive for regular users.

2.8 CONCLUSIONS

In vision-based SL recognition, the key factor is the accurate and fast hand tracking and segmentation. Some researchers have done the development of methods to capture data (image acquisition) using media such as data glove, data vision (video camera, digital camera, web camera, kinect). Most studies are interested in using kinect for the data capture media for sign language, with a skeleton-based algorithm. This is based on the existing advantages such as: the Kinect and depth cameras, in general, are well suited for sign language recognition. It offers 3D data from the environment without a complicated camera setup and efficiently extract the user's body parts, allowing for recognition of not just hands and head, but also other parts such as elbows that can be of further help in distinguishing between similar signs. Another advantage is the independency of lighting conditions, as the camera uses infrared light. All body parts are detected equally well in a dark environment and there is no need for the user to wear special coloured gloves or wired gloves.

Microsoft Kinect sensor is used to obtain joint positions. Kinect SDK tracks 3D coordinates of 20 body joints in real time (30 frames per second). Researchers have developed sign language recognition method using kinect with joint skeleton tracking. The number of joint positions that have been used are six joints and seven joints.

The number of joints is an important parameter in a sign language translators, uaing of six joints and seven joints can not handle the sign language that has more complex characteristics, such as the movement of the wrist and fingers. Therefore in this studies is interested in using eight joints.

Attribute evaluation algorithms based correlation and consistency chosen because in addition to fast; the evaluation of these algorithms also produced attributes that are significant to the accuracy.

In this study, combination of the feature selection method with Artificial Neural Network is proposed. The methodology of this thesis is shown and discussed in the following chapter.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 INTRODUCTION

Chapter 3 will discuss the research methodology used for each case studies, and discusses briefly in the following sequence: the research framework, image acquisition using kinect, feature extraction, feature selection, classification using ANN. The conclusions that can be made as a result of using this methodology are discussed at the end of the chapter.

3.2 THE RESEARCH FRAMEWORK

There are several stages in the research strategy as shown in Figure 3.1, methodology phase. Each activity in the methodology had listed the methods involved.

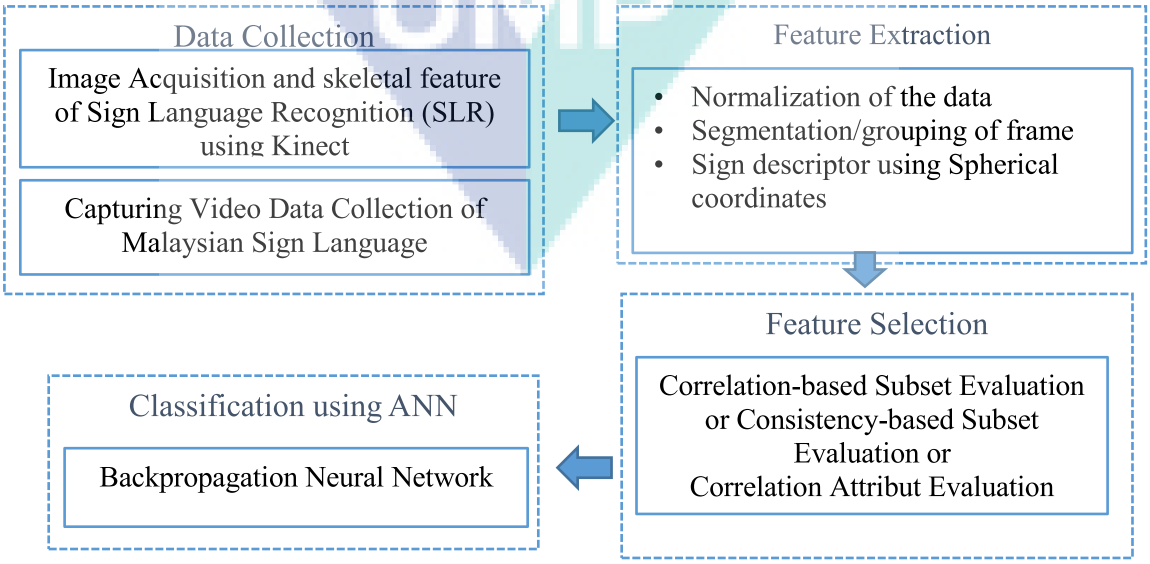


Figure 3.1 The Research Framework

3.3 DATASET COLLECTION

The first step in developing the sign dataset is by studying the vocabulary of sign language in the Bahasa Isyarat Malaysia dictionary. Then, 15 different words were chosen as shown in Table 3.1. These 15 words were chosen because they have variations of movement in sign language and often been used in communication. A sign language expert whom is a teacher at Special Education School, in Kuantan has been selected to perform all the 15 words in sign language movement. The performance was captured by camera video (as shown detail in the appendix D). Figure 3.2 shows the example of "kamu" sign out of 15 sign language sign.

Figure 3.2 The example of video for "kamu" sign

The ordinary people in learning sign language to use a dictionary have experience difficulty because the dictionary is included pictures and little explanation. Figure 2.7 shows an example of sign on dictionary "Selamat Petang" make sign "selamat" - (1), then make sign "petang" – (2).

The purpose of collection the teacher performance MSL is to guidance or reference for the five people who will do performance sign language in data collection. By referring to the video that has been recorded is expected to make it easier for the five people who will take the data and the results can be more accurate.

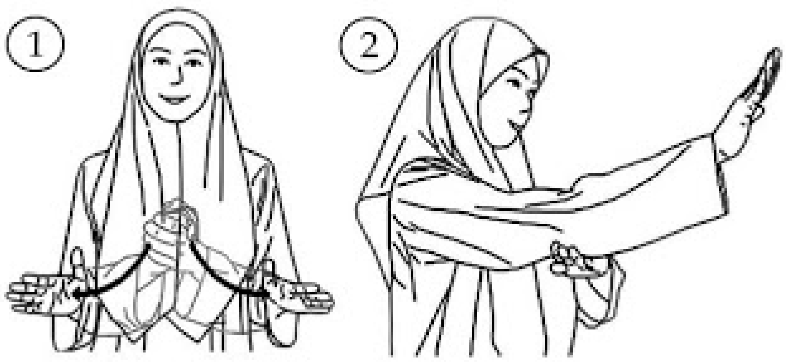


Figure 3.3 The example of ”Selamat Petang” sign

Table 3.1 The list of 15 Dynamic Malaysian Sign Language

No	Sign language	No	Sign language
1.	Apa khabar	9.	Mengapa
2.	Assalamu'alaikum	10.	Pagi
3.	Bahasa isyarat	11.	Petang
4.	Belajar	12.	Saya
5.	Cikgu	13.	Selamat
6.	Gembira	14.	Siapa
7.	Kamu	15.	Terima kasih
8.	Makan		

3.3.1 Image Acquisition and skeletal data tracking using kinect

Pre-processing is based on skeleton tracking the joints to produce 3D coordinates X, Y, Z. The studies were interested in using eight joints, left hand, right hand, left wrist, right wrist, left elbow, right elbow, torso and head as shown in Figure 3.6. After learning some sign language from sign language dictionary, there are six joints out of the 20 to be significant for the description of the mark: both hands, wrists and elbows, while Head and Torso joints are needed for normalization and modelling the steps and signs. There is no point in tracking the other joints such as the shoulders, knees, feet, etc., because they have remained almost static during the implementation of the mark. Adding these joints to mark the descriptor will be the same as adding redundant information.

Skeletal Tracking allows Kinect to follow the actions of the people. The joints of the tracked users in space can be located, and their movements can be followed. The skeletal tracking recognizes users in standing or sitting position, and facing the sensor of the camera. In the default range mode, Kinect is able to detect people standing between

0.8 meter and 4.0 meter. The practical range is 1.2 meter to 3.5 meter, which allows for the hand movement of the user (Corporation, 2013).

3.3.2 Capturing Data Collection Malaysian Sign Language

The process of capturing the data collection is done by five people (as shown in Appendix C). Each person was asked to perform 15 signs and every sign was repeated 5 times (15 Signs, 5 people and repeated 5 times). The total of data samples were 375 in Malaysian Sign Language. Sample data in the form of X, Y, and Z was obtained from Kinect, by considering only 6,7 and 8 joints in the skeleton feature.

Figure 3.4 shows user interface design, which was used to capture the video of Malaysian sign language. The interface consists of parts such as RGB camera, depth Sensor, Record, File Name.

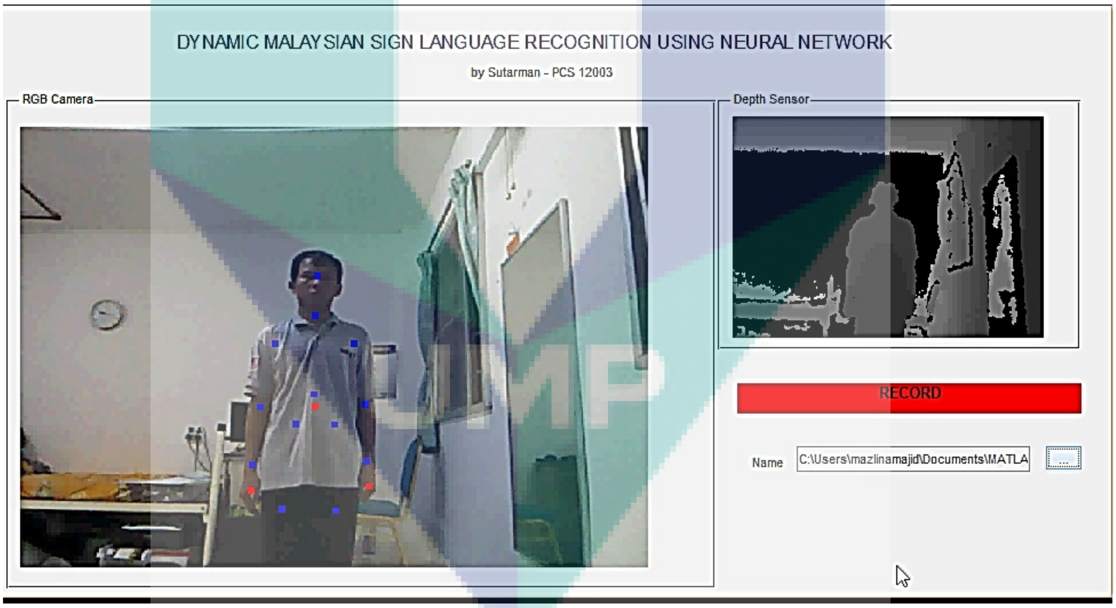


Figure 3.4 The design of User Interface to capture data collection

RGB camera : to display the image in RGB mode.
Depht Sensor: to display the image in Depth sensor mode.
Record : to display indicators start (red) and stop (white) of data capturing in the sign language.
File Name : shows the path and file name to be saved.

3.4 FEATURE EXTRACTION

Figure 3.5 shows, the stages of feature extraction is conducted in this study, such as Normalization of the data, the Segmentation of frames, Sign descriptor using spherical coordinates.

Feature Extraction

- Normalization of the data
- Segmentation of frame
- Sign descriptor using Spherical coordinates

Figure 3.5 The stages of feature extraction

The problems within the sign language recognition system are different physical characteristics of the users such as body size (children and adults) and the position of the camera sensor. Hence, the process of normalization is needed to cater the users of different sizes, whilst the camera sensor positions are detected. In this study, each frame of the joints list tracked of eight joints as shown in Figure 3.6 which are the positions of the joints and the notation that will be used.

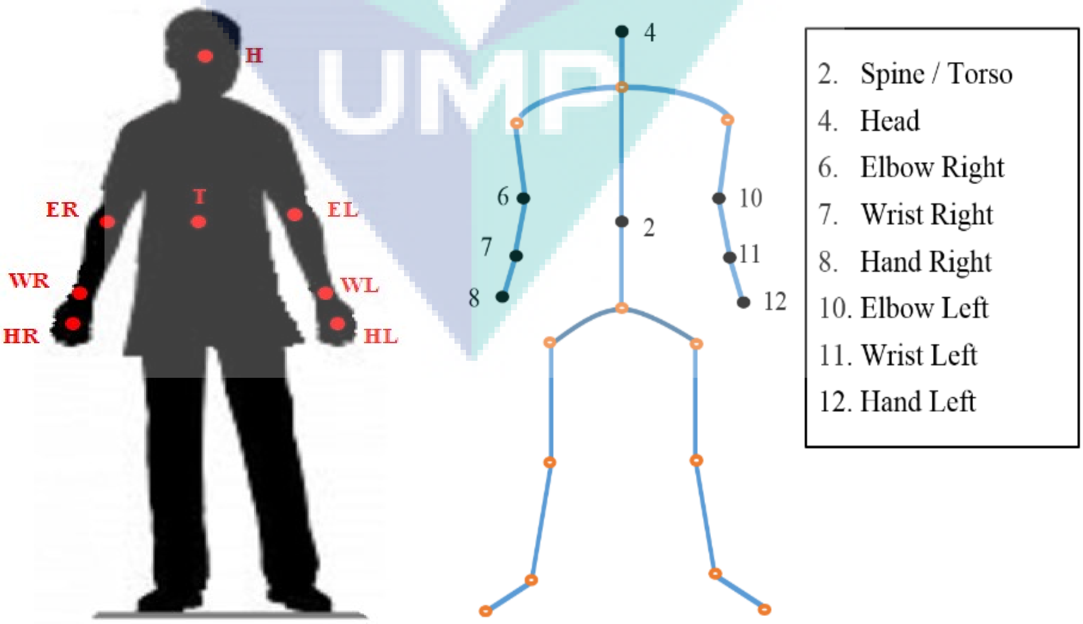


Figure 3.6 Kinect Eight joints positions

3.4.1 Normalization of data

To identify deaf users with different sizes and positions is necessary to do normalization the data so that the process of recognition can work well.

Invariant to user's positions (Torso)

Normalization is taken into account for the positions of different users in a room for the data stored in that position. As show in Figure 3.7 a little variation in depth can cause a considerable variation of the X and Y values. Distances between one joint and another can drastically vary depending on the position of users.

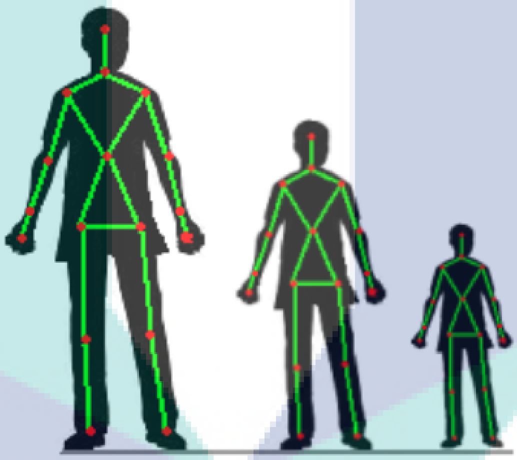


Figure 3.7 Normalization required for the position of the user.

Cartesian coordinate X, Y, and Z can be stored directly. It can use the OpenNI/NITE. In this study, it consist of the normalization of all coordinate together with respect to the Torso position.

The position that remains always constant throughout the frame signs is considered correct and will be used to make the position-invariant systems. Instead of using cartesian coordinate X, Y, and Z, the spherical coordinates of Torso are stored as the origins.

Invariant to user’s size (Head)

In a sign language translator, the system must be able to translate the receipt of the user either tall or short, so that the translator can produce output with the right word in every case. The problem of user's size is shown in Figure 3.8. The distance from one joint to another change significantly depending on the size of the users. After the user's position normalization process, each joint is expressed by distance d relative to the Spine joint and two angles θ and φ that describes the orientation of distance.

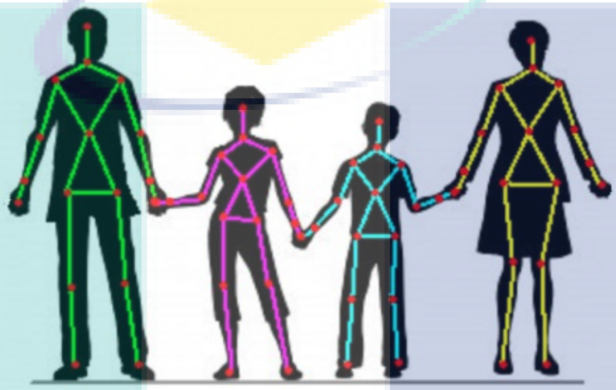


Figure 3.8 Normalization required for the user sizes

3.4.2 Segmentation/grouping of Frame

Images or videos of signs is entered in the process of recognition and is stored in the form of the frame. The descriptors do not share the same length which means that, for instance, a sample sign A can have 100 frames and a sample sign B can have 120. Sometimes, the same sign done by the same user can be slightly different in length.

The classifier must be able to compare data that can have a different length such as sign A and sign B. It is necessary to the process of segmentation or grouping for each sign of the same on different users, using the average function.

3.4.3 Sign Descriptor

The descriptor must be able to show the signs in a way that this descriptor will be unique and quite different from the other descriptors of the dictionary. As shown in Figure

3.9. the descriptor of sign contains many rows as a frame. In every frame, the spherical coordinates for each of the six joints are stored in the corresponding layers (d, θ or φ).

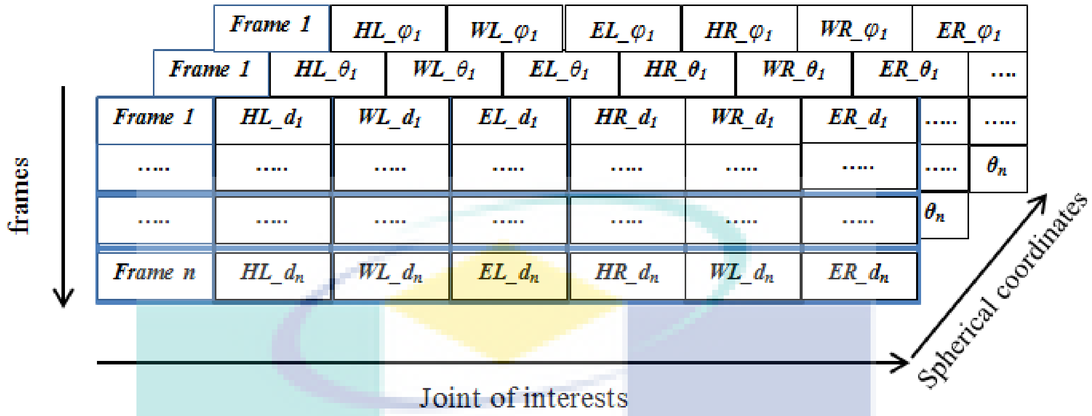


Figure 3.9 The descriptor of Sign based on the spherical coordinates for every joint

3.5 CONCLUSIONS

In this chapter, the overview of the stages in the process of recognition starting from the data collection, feature extraction, feature selection and classification is presented. The feature extraction include normalization of data, segmentation of frames each sign and sign descriptor. This study used three algorithm in feature selection: CfsSubsetEval, CSE, CorrelationAttributeEval in order to performance analysis on result accuracy. Each are compared with different number of joints (6,7,8).

CHAPTER 4

PROPOSED METHOD

4.1 INTRODUCTION

Chapter 4 will discuss the proposed method, in the following sequence: CFsSubsetEval of feature selection that contains an algorithms and flowchart of the feature selection, then classification process using ANN with detailed information about the architecture and the parameter of Neural Network. The conclusions that can be made as a result are discussed at the end of the chapter.

4.2 FEATURE SELECTIONS

Basically in the search subset of the dataset is to achieve ideal results, to look one by one, starting from an empty set of the set contains the entire attribute. However, it cannot be implemented, especially for a fairly large number of attributes. It will take a very long and its unpractical.

The algorithm of search for a subset used for this study is greedy algorithms (BestFirst) for CSE and CfsSubsetEval and CorrAttributEval used Ranker. This study will compare the attributes of selection CSE and CfsSubsetEval and CorrAttributEval. In this study the process of recognition before the classification, is done on feature selection as shown in Figure 4.1, it is the function of selecting a subset of relevant/important features by removing most irrelevant and redundant features.

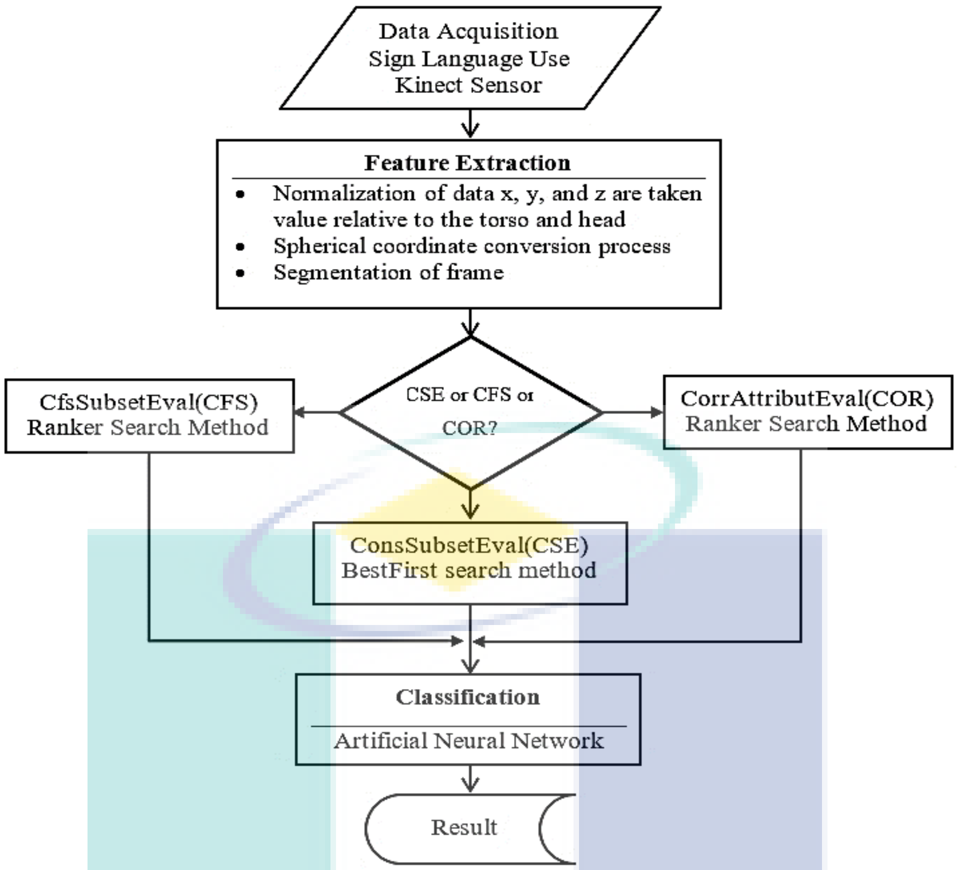


Figure 4.1 The flowchart process of Feature Selection

4.2.1 The algorithm of Feature Selection

There are two operational components in Feature selection process: evaluation of feature subsets; and searching for feature subsets as show in Figure 4.2. In this study, CfsSubsetEval evaluation of feature subsets and the Best-First search method of feature subsets are proposed.

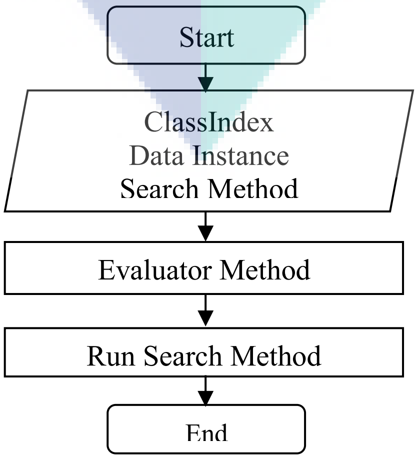


Figure 4.2 The flowchart process of Feature Selection Algorithm

The process of CfsSubsetEval as shown in Figure 4.3 is done using WEKA software. WEKA software is an open source software that also known as free software. It can be downloaded for free. This study uses source code plug-in WEKA software, and developed using Delphi 7.0 software tool.

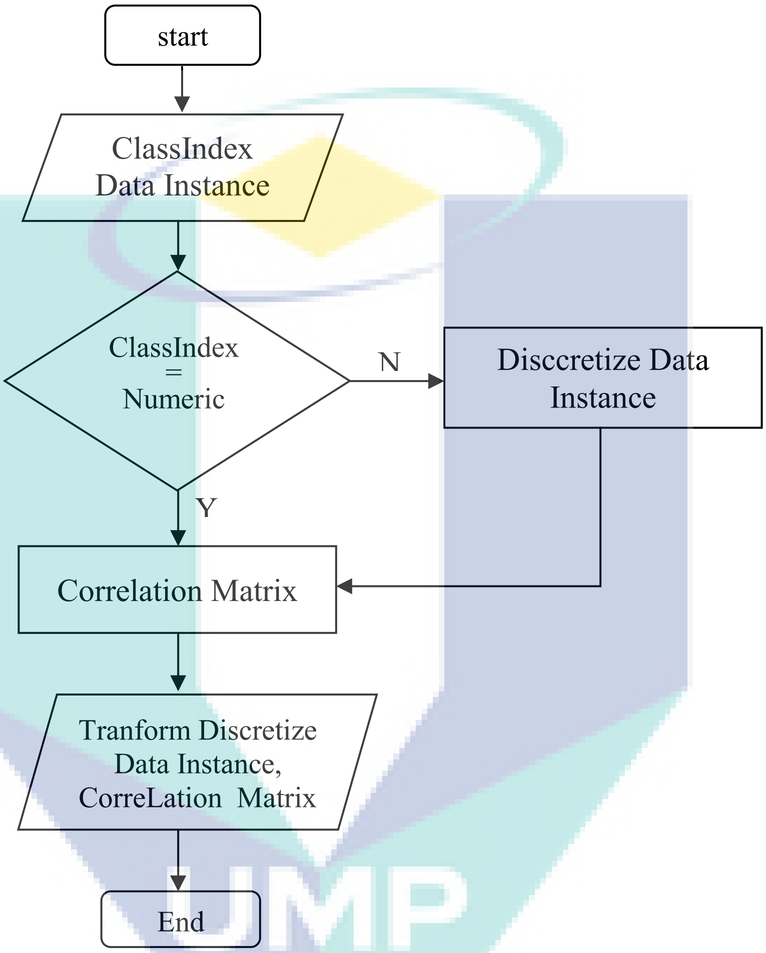


Figure 4.3 The Flowchart of Build Evaluator CsfSubSetEval

In cfssubseteval, there is the disadvantage that if the number of attributes are more than 100, the computer can not produce anything (computer hanging). To solve this problem, the study proposes of algorithm for reading the data attribute should be more than 100, as shown in Figure 4.4.

The following is new algorithm proposed in this study:

The discretize Algorithm:

1. *Do the data mapping;*
   (Procedure TConsSubsetEval.BuildEvaluator (AInstances: TInstances);)
   The command used to build the entropy
   {FDisTransform: = TDiscretize.Create;
   FDisTransform.SetUseBetterEncoding (True);
   FDisTransform.SetInputFormat (FTrainInstances);
   FTrainInstances: = FDisTransform.useFilter;}.
2. *Determine the range / limit first-last (function TDiscretize.UseFilter: TInstances;)*
3. *Determining classindex / number of attributes (function TDiscretize.UseFilter: TInstances;)*
4. *Make output format to determine the relation of data (FOutputFormat.RelationName: = GetInputFormat.RelationName;), determining the capacity of the data (FOutputFormat.Attributes.Capacity: = GetInputFormat.NumAttributes;)*
5. *Calculate the cut point*
6. *Looping from the low to the highest data*
7. *Using a copy to preserve order*
8. *Finding the first instance*
9. *Calculating the cut point //entropy*
10. *The evaluators attribute initialization (Procedure AttributeSelection.Select Attributes (Instances: TInstances); fieldWidth: = Round (Ln (FTrainInstances.numAttributes) + 1.0);)*
11. *Do the data of search*
    (Procedure TAttributeSelection.SelectAttributes (AInstances: TInstances); attributeSet: = FSearchMethod.RunSearch (FEvaluator, FTrainInstances);)
12. *To evaluate the function evaluatesubset subset*
    (function TConsSubsetEval.EvaluateSubset (subset: TBitSet): Double;)
13. *Calculates the number of of attributes*
14. *Create and manage two arrays corresponding to the number of attributes (SetLength (instArray, Count);) enter attribute to an array*
15. *Creating a new hash table*
    (If (FHashTable.Count> 0) then FHashTable.Clear;)
16. *Input data to determine the class hash table and index sort.*
17. *Calculate the amount Correlate*
    (function TCfsSubsetEval.CfsCount: Double;) check the existing data in the hash table.

Figure 4.4 The discretize algorithm

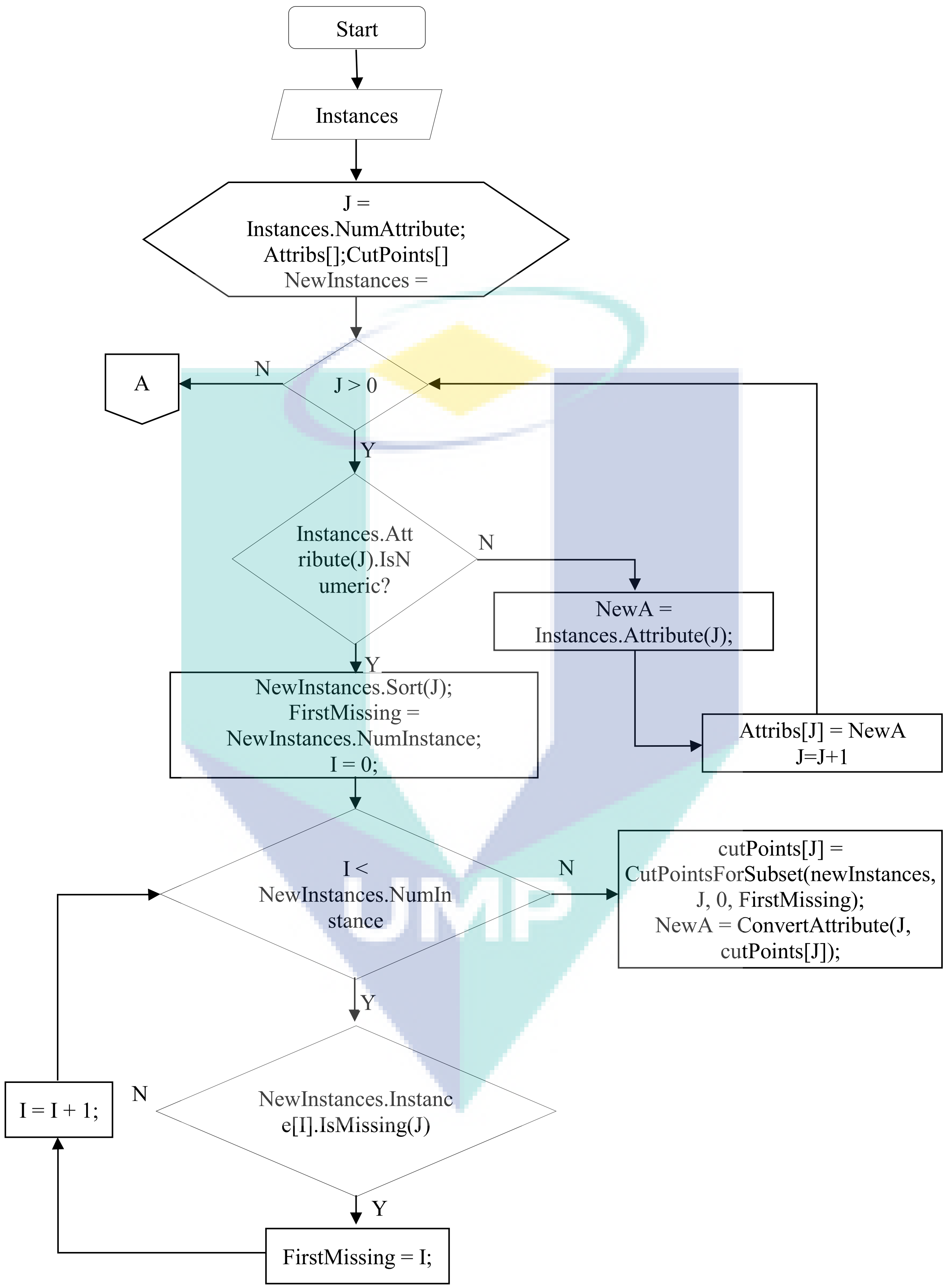


Figure 4.5 The flowchart algorithm

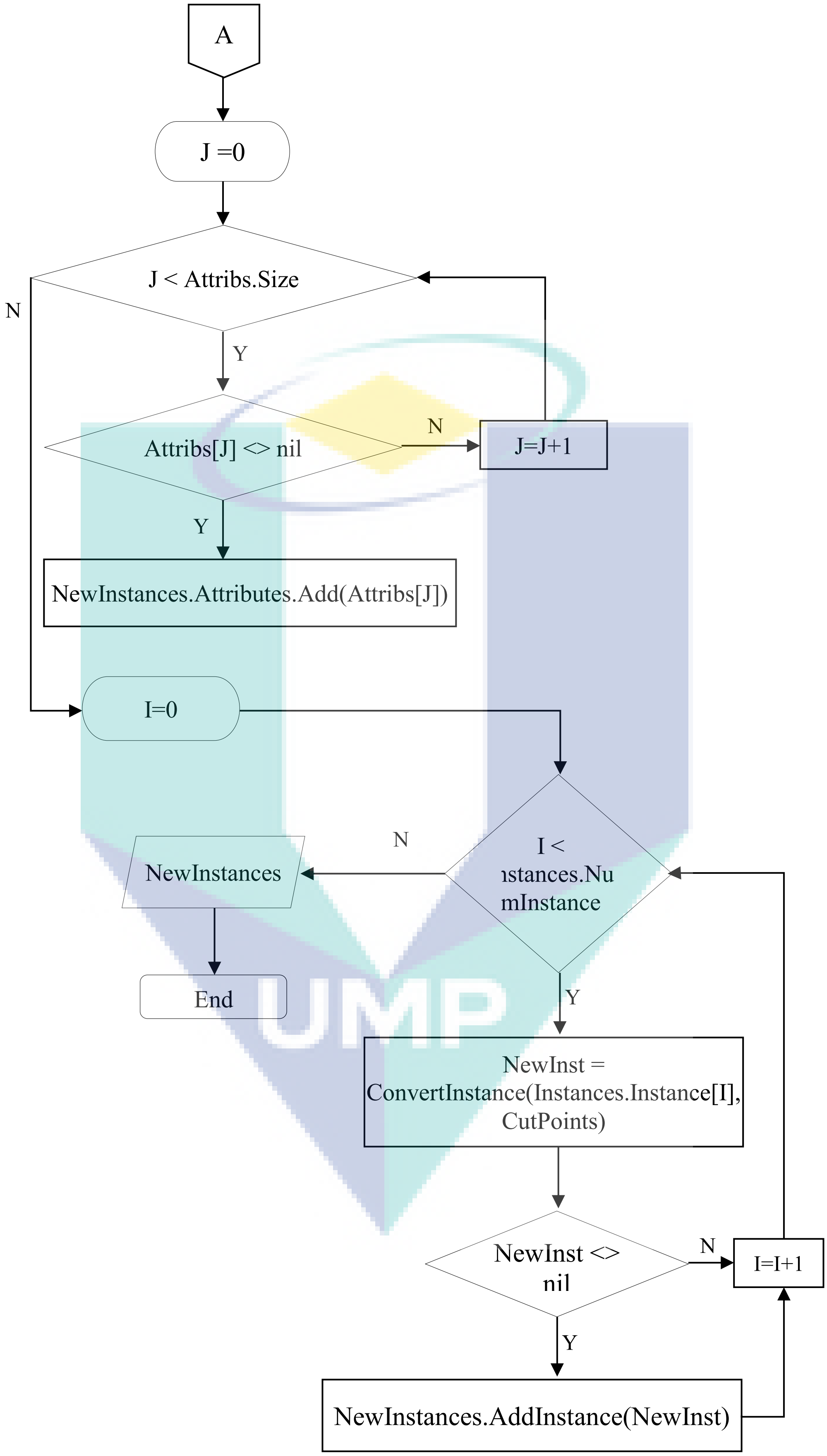


Figure 4.5. Continued

4.3 CLASSIFICATION USING ANN

4.3.1 Neural Network Modeling

The Back-propagation Neural Network (BPNN) is a supervised learning neural network model highly applied in different applications around the globe. Many researchers use this method because it is more practical or simple and has a relatively good performance.

In this study, a neural network model using error back-propagation is used. This network is trained by the conventional back-propagation procedure with momentum and adaptive learning rate. Typical values for the learning rate parameter are numbers between 0 and 1. The hidden and output neurons are activated by the bipolar sigmoidal activation function.

a. Network Architecture Selection

Once the data representation and selection are complete, the next step is to model and train the ANN. The neural network with this architecture was trained by back propagation with momentum. The network consists of three layers: input, hidden and output layers. The number of hidden layer nodes was selected experimentally.

The Neural Network architecture has 672 layers consisting of an input layer, a hidden layer and 15 output layers, as shown in Figure 4.6. The number of cells input layer 672 is composed of 8 (the number of joint), six (3D coordinates - x, y, z; spherical coodinates - r, θ, φ) and 14 (the number of groups of frames each sign). There are 225 samples that were used for training the neural network and 150 samples for testing. A trial weight set consists of 375 sets of randomized weight samples was considered.

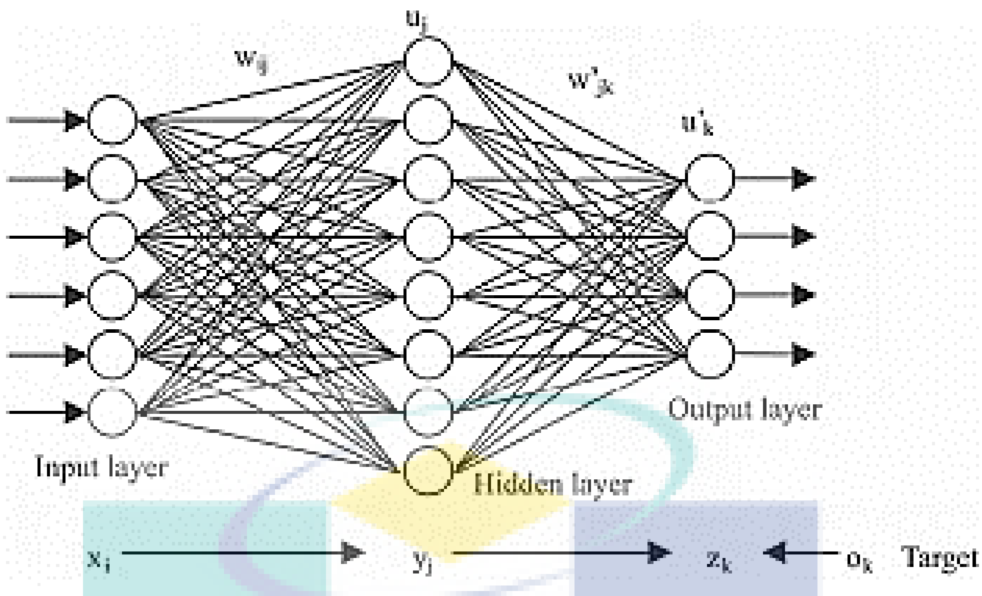


Figure 4.6. The Neural Network Architecture

b. Selection of Network Parameter

There are a number of parameters that are associated with this training. Some of the parameters of the network such as learning rate (μ), momentum constant (α) and the number of epochs were determined experimentally. The Neural Network is trained and tested with different kinds of learning rate, momentum coefficients and hidden layer (as shown in Table 4.3). In this research using node of hidden layer random from 25 to 500 with multiple 25, different combinations of learning rate (α): {0.04, 0.05, 0.06, 0.07} and learning rate (μ): { 0.06, 0.07, 0.08} were simulated to observe their effect on network convergence and recognition.

The combination was taken based on the results of experiments as shown in Table 4.1, the default value of range the learning rate and momentum was 0 - 1. Typical values for the learning rate parameter are numbers between 0 and 1 (Bartkowiak, 2004). To generate the best accuracy experiments were carried out with a variety of learning rate and momentum. Firstly chosen middle value between 0 – 1 was 0.5, then the result were compared with a combination between 0.1 to 0.9 as shown in Table 4.1 because ofthe result, no better, combination used value 0.01 – 0.09 and the best combination for learning rate (α): {0.04, 0.05, 0.06, 0.07}and learning rate (μ): {0.06, 0.07, 0.08}.

Table 4.1 The result of experiments differences in learning rate and momentum coefficients with value 0.1 – 0.9

Learning Rate	Momentum Coefficients		
	0.25	0.50	0.70
	Accuracy %	Accuracy %	Accuracy %
0.25	65.00	79.44	32.22
0.50	70.00	61.67	19.44
0.70	54.44	50.56	11.67

Table 4.2 The result of experiments differences in learning rate and momentum coefficients with value 0.01 – 0.09.

Learning Rate	Momentum Coefficients						
	0.025	0.040	0.050	0.060	0.070	0.080	0.100
	Accuracy %	Accuracy %	Accuracy %	Accuracy %	Accuracy %	Accuracy %	Accuracy %
0.025	81.111	81.111	83.889	83.333	82.222	82.778	70.556
0.040	81.000	81.000	83.333	85.670	85.560	85.670	81.000
0.050	82.778	83.889	83.890	86.111	87.222	86.111	83.889
0.060	83.333	83.333	83.890	86.111	88.890	86.667	83.333
0.070	82.222	83.889	82.222	88.333	86.667	85.560	82.222
0.080	82.222	82.222	81.111	83.333	82.222	82.778	70.556
0.100	81.111	81.667	80.000	81.111	82.222	80.000	75.000

Table 4.3 The parameter of Neural Network

Parameter	Value
Input layer	672 cells
Hidden layer	25, 50, 75, 100 --- 500
Output layer	15 class
Learning rate	0.04; 0.05; 0.06; 0.07
Momentum coefficients	0.06; 0.07; 0.08

The parameters will be applied to differences number of joints: six of joints, seven of joints and eight of joints.

4.4 CONCLUSIONS

In this chapter, the methodology for dynamic Malaysian Sign language recognition is proposed. A complete overview of the stages in the process of recognition starting from the data collection, feature extraction, feature selection and classification. The feature extraction include normalization of data, segmentation of frames each sign and sign descriptor.

This study proposed three algorithm in feature selection: CfsSubsetEval, CSE and CorrelationAttributeEval. Each also was compared with different number of joints (6,7,8). The experimental results are shown and discussed in the following chapter.

CHAPTER 5

EXPERIMENTATION AND RESULT

5.1 INTRODUCTION

Chapter 5 will discuss the experimentation and results, briefly in the following sequence: image acquisition using kinect, feature extraction, feature selection, classification using ANN. The conclusions that can be made as a result of the experiment are discussed at the end of the chapter. Stages in the experiment are:

a. Image Acquisition and skeletal feature of Sign Language Recognition (SLR) using Kinect.
b. The feature extraction process
    - Normalization of the feature data;
    - Coordinate conversion process with spherical coordinate;
    - Segmentation of the frame to get the same number of dimensions using statistical function (mean).
c. Implementation of the method classification using Artificial Neural Network
d. Implementation of the Correlation-based Feature Selection Subset Evaluation CfsSubsetEval and CSE and CorrAttributeEval with Artificial Neural Network, last compared performance.

5.2 RECOGNITION RATE

The performance of the recognition is evaluated based on its ability to correctly classify samples to their corresponding classes. The Measurements used to accomplish this job is called the recognition rate. The recognition rate is defined as the ratio to the number of correctly classified samples to the total number of samples as shown in Eq. (5.1).

$$Recognition\ Rate = \frac{Number\ of\ correctly\ classified\ samples}{Total\ number\ of\ samples}\ X\ 100\ \% \quad (5.1)$$

5.3 EXPERIMENT RESULT

Experiments started in feature extraction process with making the relative distance from the data of X, Y, and Z to the torso and head. Data have differences value ranges of frames because they are from different people and different time. Therefore it must be made within the same dimension, by dividing each image frame into the same segmentation. Mean statistical function was used in the process of frame segmentation of every sign. To determine how many groups experimenting was done with a number of groups: 12, 13, 14, 15 and the evaluation made 30 times the experiment. The experiment results showed that the best group number is 14, as shown in Table 5.1.

Table 5.1 The results from the experiment Grouping frame

No of experiments	Number of groups					
	10	11	12	13	14	15
1	73.00	71.48	72.64	75.00	75.00	78.00
2	15.00	34.56	36.49	75.34	75.34	79.00
3	16.67	35.57	38.85	75.34	75.34	79.67
4	18.00	38.59	40.54	76.01	76.01	81.67
5	21.67	41.95	43.58	76.01	76.01	75.00
6	25.33	51.68	52.70	76.01	76.01	78.00
7	35.33	63.42	68.24	77.36	77.36	79.00
8	52.00	67.45	69.93	78.38	78.38	79.00
9	53.33	67.79	71.28	79.05	78.38	79.67
10	53.67	69.13	72.97	79.39	79.00	81.67
11	55.00	72.15	73.65	78.00	79.67	75.00
12	60.33	72.48	73.99	79.00	81.67	78.00
13	60.67	72.48	74.32	79.67	82.00	79.00
14	63.33	73.15	74.32	81.67	83.00	79.67
15	70.00	73.83	75.00	70.00	81.67	81.67
16	72.00	74.50	75.34	72.00	82.00	79.00
17	74.00	74.83	75.34	78.00	83.00	79.67
18	74.00	74.83	76.01	79.00	84.00	81.67
19	75.00	75.84	76.01	79.67	85.00	75.00
20	75.00	76.17	76.01	81.67	85.56	78.00
21	75.00	76.51	77.36	75.00	84.00	79.00

Table 5.1 Continued

No of experiments	Number of groups					
	10	11	12	13	14	15
21	75.00	76.51	77.36	75.00	84.00	79.00
22	75.33	76.85	78.38	78.00	85.00	79.67
23	76.00	77.18	78.38	79.00	85.56	81.67
24	77.00	77.52	78.38	79.67	84.00	79.00
25	77.00	77.85	78.72	81.67	85.00	79.67
26	77.33	77.85	79.05	82.00	85.56	81.67
27	78.00	78.52	79.39	83.00	84.00	75.00
28	79.00	79.19	83.00	84.00	85.00	78.00
29	79.67	79.19	84.00	85.00	85.56	79.00
30	81.67	79.53	85.00	85.56	85.56	79.67
Average	60.64	68.74	70.63	78.65	81.45	78.99

Four different approaches were evaluated:

Experiment 1: Artificial Neural Network; the descriptor contains the spherical coordinates (X, Y, and Z), the classifier used Artificial Neural Network, as shown in Figure 5.2.

Experiment 2: CSE with Artificial Neural Network, as shown in Figure 5.4 the descriptor contains the spherical coordinates (X, Y, and Z).

Experiment 3: CorrAttributeEval with Artificial Neural Network, as shown in Figure 5.9; the descriptor contains the spherical coordinates (X, Y, and Z).

Experiment 4: CfsSubsetEval with Artificial Neural Network, as shown in Figure 5.11; the descriptor contains the spherical coordinates (X, Y, and Z).

Evaluation was conducted with a variation of node in the hidden layer (25, 50, 75, 100, 125 … 500), different kinds of learning rate, momentum coefficients, hidden layer, number of joints as shown in Table 5.2. Figure 5.1 shows the number of joint that were used the experiments.

Table 5.2 Neural network parameters used in the experiments

Parameter	Value
Hidden layer	25, 50, 75, 100 --- 500
Learning rate	0.04; 0.05; 0.06; 0.07
Momentum coefficients	0.06; 0.07; 0.08
Number of Joint	6 joint; 7 joint and 8 joint

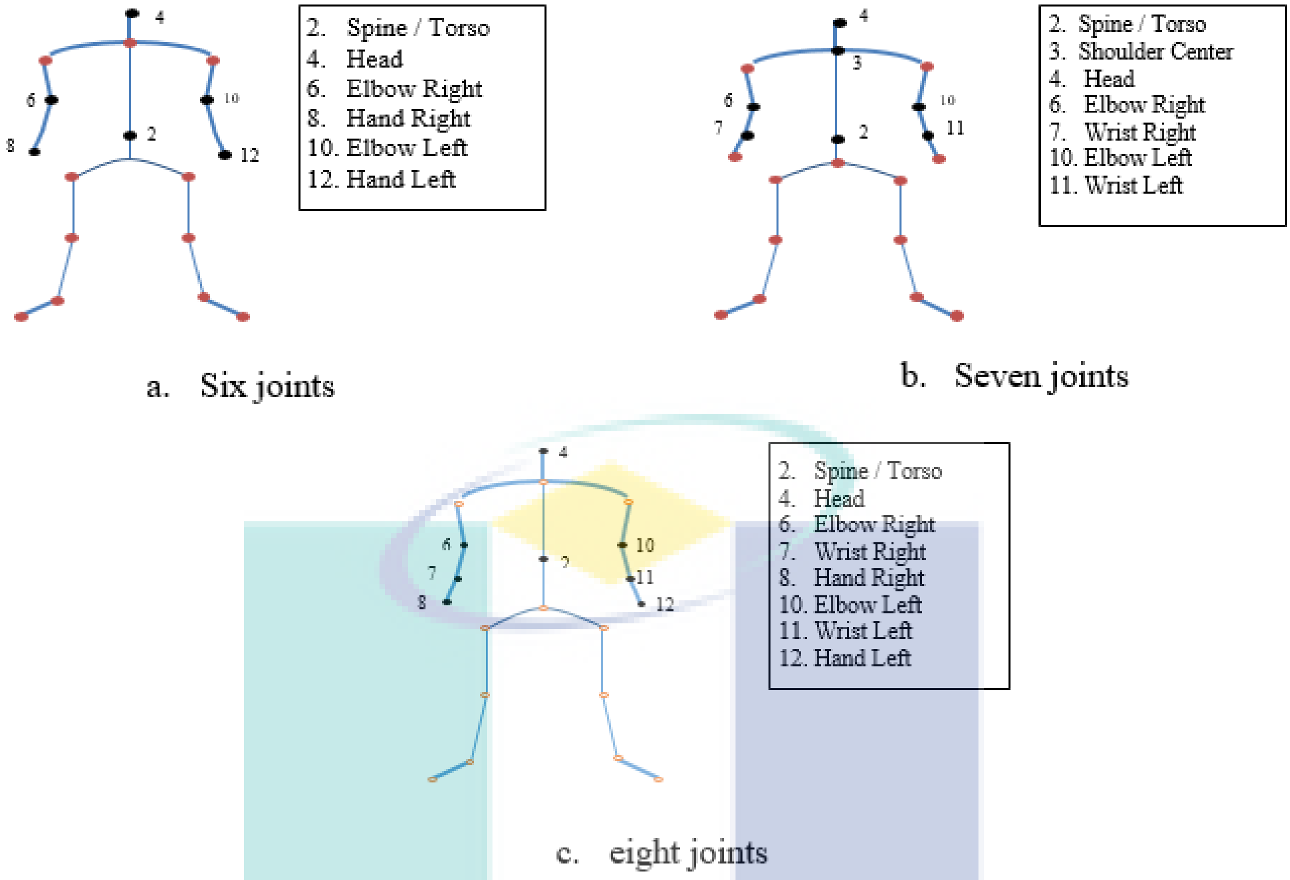


Figure 5.1 Number of joint that were used the experiments

5.3.1 Result of Experiment 1

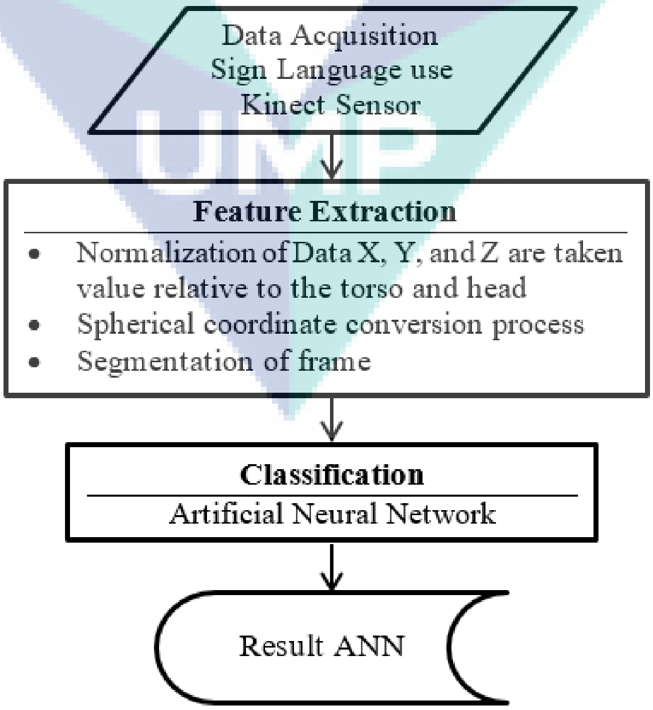


Figure 5.2. Flowchart system of experiment 1

The aims of experiment 1 was to apply the classification method using ANN. As shown in Figure 5.2, the flowchart experiment began with reading a sing date obtained using kinect sensor. Then the features were extracted by normalizing and converting to be spherical coordinates, the frame segmentation process for each sign. After extraction feature was done, the next stage was classification using ANN. Evaluation was conducted with a variation of a node in the hidden layer, different kinds of learning rate, momentum coefficients, hidden layer, number of joints. The following are the results of the experiment 1 by a category number of joint that has been tested.

Table 5.3 shows the number of joints 6 that use spherical coordinate with classifier ANN, the result of accuracy in the experiments is 85.56 %, variation of node in hidden layer is 350, learning rate is at 0.06, momentum coefficients is at 0.06, epoch 100 and time 4.502 sec. the completed results are shown in Appendix B..

Table 5.3 The results of experiment 1 (6 joint without feature selection)

Number Node Of Hidden Layer	Learning Rate (LR)	Momentum Coefficients (MC)	accuracy (%)	epoch	time (sec)
25	0.07	0.06	81.11	1144	2.203
50	0.06	0.06	85.00	269	5.021
75	0.06	0.06	83.89	125	2.788
100	0.05	0.06	85.00	158	4.193
125	0.05	0.06	85.00	244	7.243
150	0.05	0.08	83.89	130	4.586
175	0.05	0.08	84.44	72	2.761
200	0.07	0.07	83.33	96	4.111
225	0.05	0.06	82.78	224	11.154
250	0.06	0.07	83.89	140	7.161
275	0.04	0.07	83.89	197	13.378
300	0.06	0.07	85.56	139	11.190
325	0.04	0.08	82.22	131	9.688
350	0.06	0.06	85.56	100	4.502
375	0.05	0.08	83.89	271	24.606
400	0.04	0.06	81.67	208	18.689
425	0.06	0.07	82.22	240	22.658
450	0.06	0.08	78.33	94	9.464
475	0.05	0.08	83.89	146	15.589
500	0.04	0.08	82.78	277	30.389

Table 5.4 shows the number of joints 7 that use spherical coordinate with classifier ANN, the result is variation of node in hidden layer is 75, and accuracy is 87.11 %, learning rate is at 0.05, momentum coefficients is at 0.06, epoch is 256 and time is 5.086 sec. the completed results are shown in Appendix B.

Table 5.4 The results of experiment 1 (7 joint without feature selection)

Number Node Of Hidden Layer	Learning Rate (LR)	Momentum coefficients (mc)	Accuracy (%)	Epoch	Time (sec)
25	0.06	0.07	84.44	246	4.600
50	0.07	0.08	86.67	113	2.168
75	0.05	0.06	87.11	259	5.086
100	0.07	0.08	83.89	95	2.511
125	0.04	0.08	86.11	199	6.209
150	0.04	0.08	83.89	136	4.648
175	0.05	0.06	83.89	259	12.074
200	0.04	0.07	84.44	265	13.335
225	0.05	0.06	82.22	148	7.488
250	0.06	0.07	83.33	387	8.003
275	0.06	0.08	83.33	198	13.394
300	0.04	0.07	82.78	81	6.287
325	0.04	0.08	82.22	154	12.028
350	0.04	0.07	83.33	130	12.294
375	0.07	0.06	83.89	216	19.422
400	0.06	0.08	83.89	190	17.790
425	0.05	0.08	83.89	606	60.902
450	0.05	0.07	85.00	417	52.079
475	0.05	0.06	84.44	194	21.746
500	0.05	0.08	85.00	562	65.449

As shown in Table 5.5, the number of joints 8 that use spherical coordinate with classifier ANN, the results of accuracy in the experiments is 90.56 %, variation of node in hidden layer is 175, learning rate is at 0.06, momentum coefficients is at 0.07, epoch 238 and time 11.034 sec. the completed results are shown in Appendix B..

Table 5.5 The results of experiment 1 (8 joint without feature selection)

Number Node Of Hidden Layer	Learning Rate (LR)	Momentum coefficients (mc)	Accuracy (%)	Epoch	Time (sec)
25	0.07	0.08	84.44	509	8.112
50	0.04	0.08	85.00	183	3.292
75	0.07	0.08	83.33	259	6.177
100	0.07	0.07	85.56	252	9.174
125	0.04	0.06	82.78	90	2.823
150	0.07	0.08	83.33	60	2.133
175	0.06	0.07	90.56	238	11.034
200	0.07	0.06	84.44	242	11.017
225	0.04	0.07	81.11	131	7.408
250	0.07	0.08	85.00	208	82.220
275	0.07	0.06	82.22	189	13.903
300	0.05	0.07	82.22	142	10.802
325	0.05	0.08	82.22	143	28.088
350	0.04	0.06	82.22	148	12.777
375	0.06	0.08	80.00	124	11.535
400	0.05	0.07	80.56	234	23.092
425	0.05	0.08	81.67	190	19.957
450	0.07	0.06	80.56	182	22.605
475	0.06	0.08	78.33	163	19.034
500	0.07	0.06	78.89	167	20.242

Table 5.6 shows the results of experiments without feature selection and different number of joints. The results of accuracy rate on the number of joints 6 is 85.56 %, the number of joints 7 is 87.11 % and the number of joints 8 is 90.56 %.

Table 5.6 The comparison of experiment results 1 (ANN with different number of joint)

Number of Joint	Accuracy (%)	Epoch	Time (sec)	Number node of hidden layer	Learning Rate (lr)	Momentum Coefficients (mc)
6 Joint	85.56	100	4.502	350	0.06	0.06
7 Joint	87.11	259	5.086	75	0.05	0.06
8 Joint	90.56	238	11.034	175	0.06	0.07

The comparative diagram of the experiment results 1 (ANN with number of joint 6, 7 and 8) are shown in Figure 5.3.

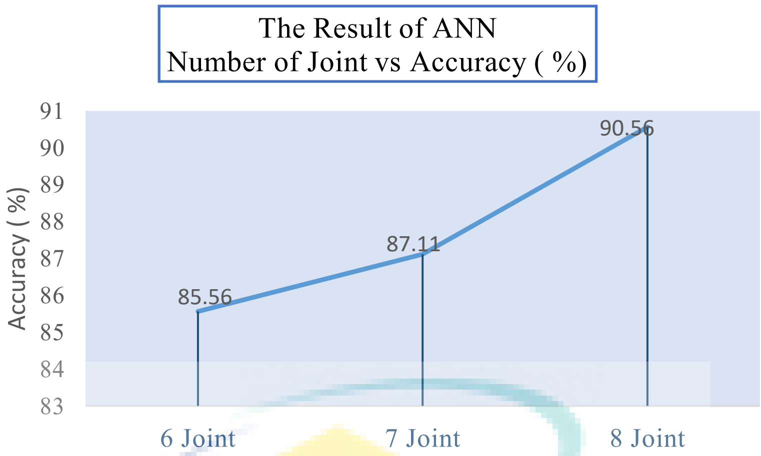


Figure 5.3 Comparative diagram of ANN with different number of joint (6, 7 and 8 joints)

5.3.2 Result of Experiment 2

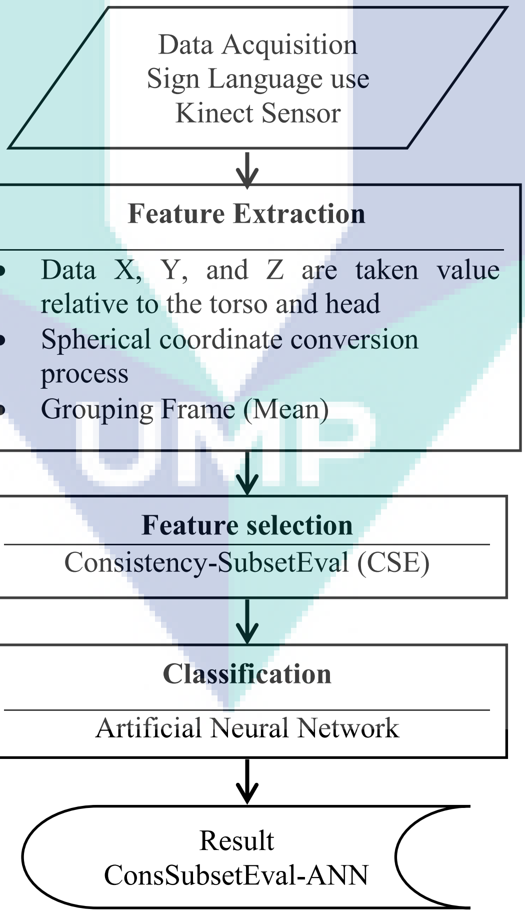


Figure 5.4 Flowchart system of the experiment 2

In this study CSE and CfsSubEval, was used while search method to find the best set of features used BestFirst. The stage of experiment is similar to an experiment 1, but after the extracted data feature selection process was done, it was the function to filter attributes related to the data of sign.

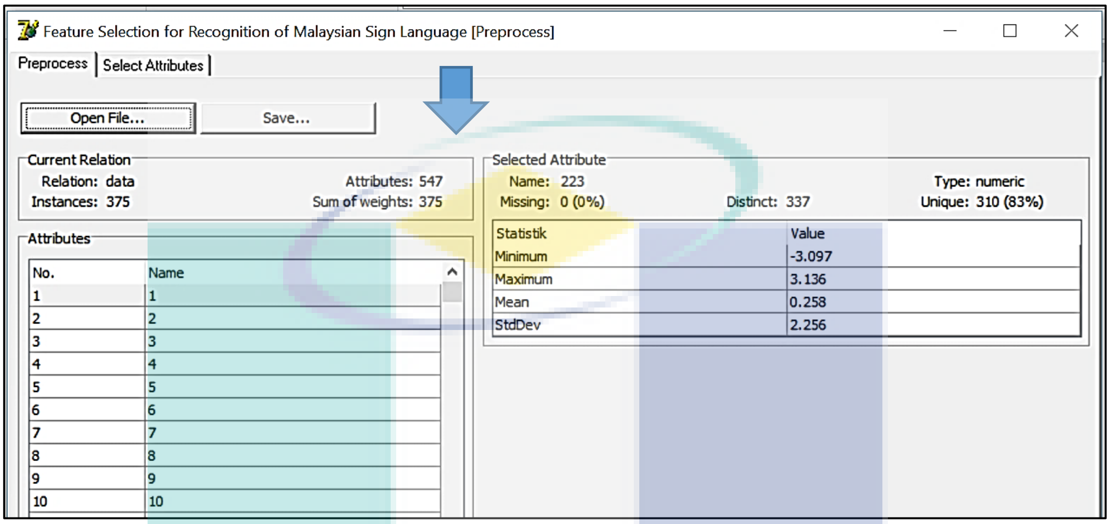


Figure 5.5. The of sign before running on Feature Selection (Number of joints 6)

Figure 5.6 shows the number of attributes before filtering is 547 attributes with sum of weights data 375. Figure 5.6 show the running process of CFE with number of joints 6.

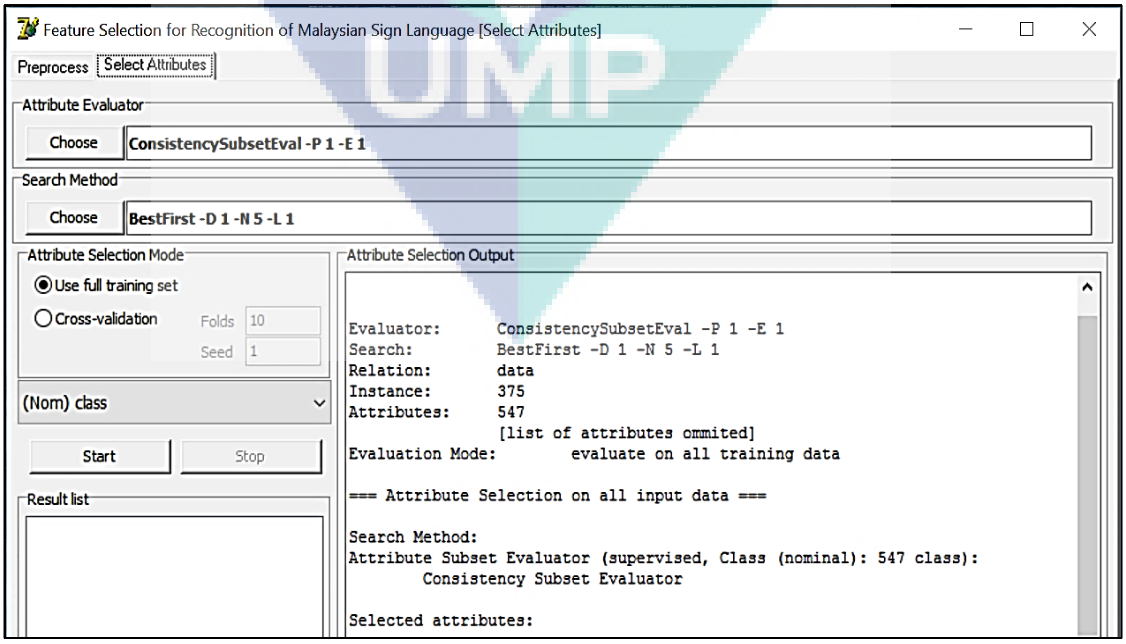


Figure 5.6 The running of CSE with number of joints 6

Figure 5.7 shows the result of filtering process, there are 13 attributes with sum of weights data 375, before filtering is 547 attributes with sum of weights data 375.

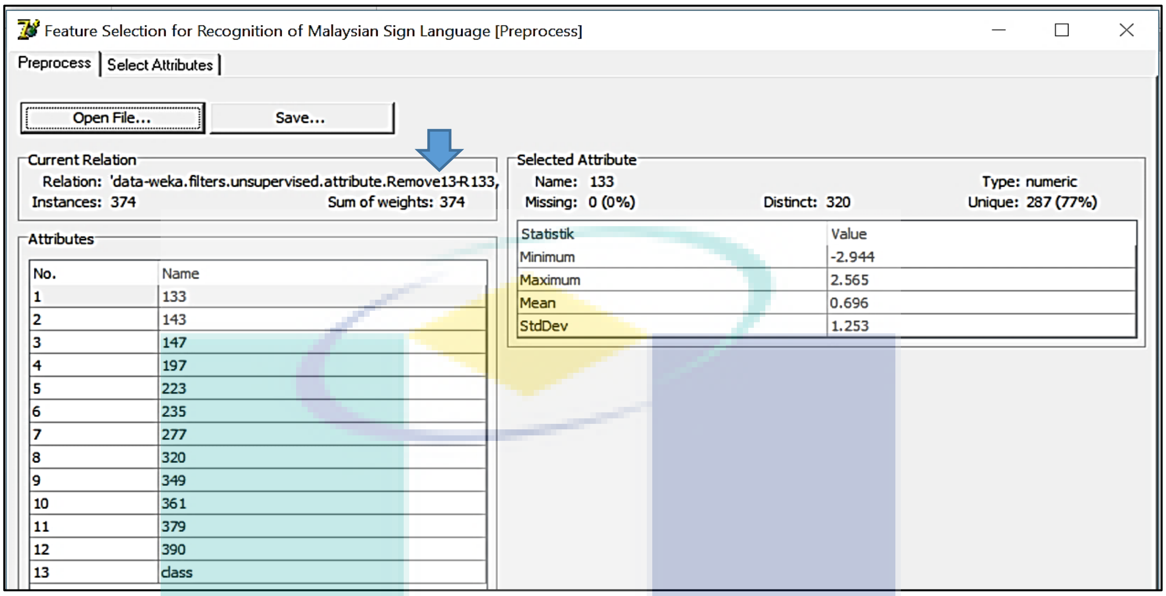


Figure 5.7 The result of CSE with number of joints 6

Table 5.7 shows the number of joints 6 that use CSE and classifier ANN, the result is a variation of node in hidden layer is 275, accuracy is 89.56 %, learning rate is at 0.05, momentum coefficients is at 0.06, epoch is 434 and time is 6.154 sec. the completed results are shown in Appendix B.

Table 5.7 The results of experiment 2 (6 joint with CSE)

Number node of hidden layer	Learning rate (lr)	Momentum coefficients (mc)	Accuracy (%)	Epoch	Time
25	0.07	0.06	65.00	890	1.063
50	0.07	0.08	75.00	1211	2.347
75	0.07	0.06	83.33	160	2.990
100	0.06	0.07	85.00	341	4.415
125	0.06	0.07	88.33	3640	6.761
150	0.06	0.08	88.33	1170	8.271
175	0.07	0.06	87.89	868	15.228
200	0.06	0.06	88.89	3014	57.982
225	0.06	0.08	87.22	2009	39.533
250	0.05	0.06	89.56	1297	31.450
275	0.05	0.06	89.56	434	6.154
300	0.07	0.08	86.11	1278	7.082

Table 5.7 Continued

Number node of hidden layer	Learning rate (lr)	Momentum coefficients (mc)	Accuracy (%)	Epoch	Time
325	0.07	0.06	87.22	776	19.025
350	0.05	0.07	88.56	1517	12.456
375	0.05	0.06	86.11	544	44.523
400	0.07	0.06	89.45	969	10.255
425	0.07	0.08	86.67	965	32.201
450	0.04	0.06	85.56	1207	6.864
475	0.05	0.07	84.44	3000	14.724
500	0.06	0.06	85.00	1160	8.947

The result of filtering, there are 13 attributes with sum of weight data 375, before filtering is 547 attributes with sum of weight data 375. As shown in Table 5.8, the number of joints 7 that use CSE and classifier ANN, the result of experiments is accuracy is 90.56 %, variation of node in hidden layer at 200, learning rate is at 0.06, momentum coefficients is at 0.07, epoch 1403 and time 5.666 sec. the completed results are shown in Appendix B.

Table 5.8 The Results of Experiment 2 (7 joint with CSE)

Number node of hidden layer	Learning rate (LR)	Momentum Coefficients (MC)	Accuracy (%)	Epoch	Time (sec)
25	0.06	0.07	66.56	158	0.988
50	0.06	0.08	80.89	265	3.577
75	0.06	0.08	83.33	153	2.024
100	0.07	0.07	83.56	570	4.299
125	0.07	0.06	82.44	1277	21.769
150	0.07	0.06	87.56	558	7.426
175	0.04	0.08	86.33	377	7.890
200	0.06	0.07	90.56	1403	5.666
225	0.06	0.08	87.56	486	9.815
250	0.06	0.06	87.78	350	5.448
275	0.05	0.06	88.33	695	13.272
300	0.06	0.06	86.67	338	8.206
325	0.04	0.06	83.33	160	14.801
350	0.04	0.07	86.11	267	6.722
375	0.05	0.06	87.22	830	27.899
400	0.07	0.08	85.56	147	3.430
425	0.07	0.08	86.67	789	27.813
450	0.06	0.07	86.11	220	6.544
475	0.06	0.08	85.56	418	12.550
500	0.05	0.06	86.67	190	6.556

Table 5.9 shows the number of joints 8 that used CSE and classifier ANN, the result is variation of node in hidden layer at 125, and accuracy is 93.67 %, learning rate is at 0.07, momentum coefficients is at 0.08, epoch 691 and time 8.706 sec. the completed results are shown in Appendix B.

Table 5.9 The Results of Experiment 2 (8 joint with CSE)

Number Node Of Hidden Layer	Learning Rate (LR)	Momentum coefficients (MC)	Accuracy (%)	Epoch	Time (sec)
25	0.07	0.07	75.56	101	1.403
50	0.07	0.06	86.67	243	2.773
75	0.06	0.06	83.33	190	2.408
100	0.05	0.08	86.11	436	5.883
125	0.07	0.08	93.67	691	8.706
150	0.07	0.07	86.89	673	4.571
175	0.06	0.06	86.67	742	8.546
200	0.07	0.07	87.89	906	12.275
225	0.06	0.08	87.78	495	7.380
250	0.04	0.08	86.11	184	4.978
275	0.05	0.08	86.11	2144	51.079
300	0.05	0.06	85.56	1438	39.013
325	0.05	0.06	86.67	151	31.000
350	0.07	0.06	85.00	355	9.137
375	0.07	0.07	85.00	176	4.742
400	0.07	0.06	84.44	101	2.027
425	0.04	0.08	85.56	792	25.569
450	0.07	0.08	85.56	99	3.572
475	0.07	0.06	84.56	276	9.465
500	0.06	0.06	83.44	825	10.317

Table 5.10 shows the results of experiments by CSE with different number of joints. The results of accuracy rate on the number of joints 6 is 89.56 %, the number of joints 7 is 90.56 % and the number of joints 8 is 93.56 %.

Table 5.10 The comparison result experiment 2 CSE with different number of joints

Number of joint	Accuracy (%)	Epoch	Time (sec)	Number node of hidden layer	Learning rate (lr)	Momentum coefficients (mc)
6 Joint	89.56	434	6.154	275	0.05	0.06
7 Joint	90.56	1403	5.666	200	0.06	0.07
8 Joint	93.67	691	8.706	125	0.07	0.08

The comparative diagram of the result experiment 2 (CSE + ANN with number of joint 6, 7 and 8) are shown in Figure 5.8.

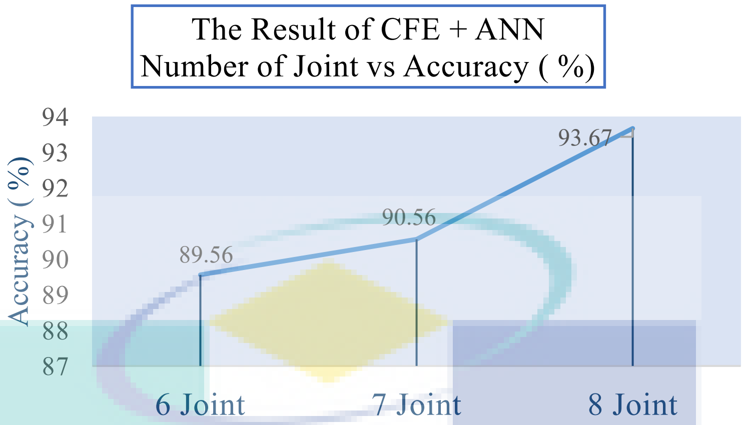


Figure 5.8 Comparative diagram of CSE with different number of joints (6,7 and 8 joint).

5.3.3 Result of Experiment 3

The next experiment applied CorrAttributeEval. In this method, search method was used to find the best set of features using Ranker.

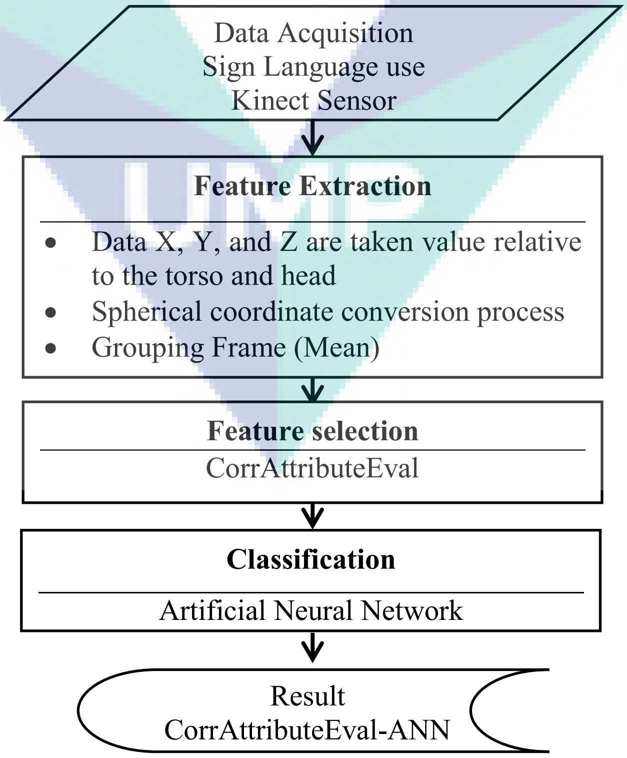


Figure 5.9 Flowchart system of the experiment 3

The number of attributes before filtering is 547 attributes, with sum of weight data is 375. The result of filtering are 547 attributes with sum of weight data is 375, the completed results are shown in Appendix B.

The next step after feature selection is the process of classification using ANN. Table 5.11 shows the number of joints 6 that use CorrAttributeEval and classifier ANN, the result of the variation of node in hidden layer is 225 and accuracy is 86.67 %, learning rate is 0.06, momentum coefficients is 0.08, epoch is 239 and time is 10.659 sec. the completed results are shown in Appendix B.

Table 5.11 The results of experiment 3 (6 joint with CorrAttributeEval)

Number node of hidden layer	**Learning rate (lr)**	**Momentum coefficients (mc)**	**Accuracy (%)**	**Epoch**	**Time**
25	0.06	0.06	80.00	81	1.213
50	0.06	0.07	83.33	269	5.212
75	0.05	0.06	85.56	104	2.451
100	0.06	0.08	86.11	93	2.495
125	0.06	0.06	83.89	650	18.643
150	0.06	0.06	85.56	200	6.679
175	0.05	0.07	83.89	97	6.501
200	0.07	0.07	85.56	107	4.434
225	0.06	0.08	86.67	239	10.659
250	0.05	0.06	83.33	163	8.299
275	0.04	0.08	83.33	390	25.662
300	0.07	0.08	84.44	129	9.344
325	0.06	0.06	83.33	157	11.608
350	0.05	0.08	77.78	117	9.587
375	0.06	0.07	83.89	272	28.773
400	0.04	0.08	82.22	260	23.197
425	0.04	0.06	81.67	468	44.289
450	0.05	0.06	80.00	125	14.321
475	0.07	0.07	81.67	148	16.069
500	0.04	0.06	83.89	114	12.387

The number of attributes before filtering process are 547 attributes, with sum of weight data is 375. And the result of filtering process are 547 attributes with sum of weight data is 375, the completed results are shown in Appendix B.

The next step is the process of classification used ANN. Table 5.12 shows the number of joints 7 that use CorrAttributeEval and classifier ANN, the results of the variation of node in hidden layer is 75, and accuracy is 88.33 %, learning rate is at 0.05, momentum coefficients is at 0.07, epoch is 113 and time is 2.168 sec. the completed results are shown in Appendix B.

Table 5.12 The results of experiment 3 (7 joint with CorrAttributeEval)

Number node of hidden layer	Learning rate (lr)	Momentum coefficients (mc)	Accuracy (%)	Epoch	Time
25	0.04	0.08	78.33	114	1.701
50	0.05	0.08	81.67	71	1.708
75	0.05	0.07	88.33	113	2.168
100	0.05	0.06	86.11	79	2.415
125	0.05	0.08	83.89	182	5.820
150	0.06	0.06	83.33	145	5.029
175	0.05	0.07	83.89	97	6.501
200	0.07	0.07	82.22	75	3.393
225	0.07	0.07	83.33	184	12.350
250	0.06	0.06	80.00	172	9.872
275	0.04	0.06	85.00	224	15.616
300	0.06	0.08	84.44	425	30.527
325	0.07	0.06	83.89	278	21.688
350	0.04	0.08	83.89	469	38.423
375	0.05	0.07	82.78	170	15.129
400	0.04	0.06	82.78	141	13.229
425	0.07	0.06	82.22	395	39.572
450	0.04	0.06	82.22	172	17.987
475	0.07	0.08	83.89	157	18.236
500	0.04	0.08	81.67	219	25.272

The next experiment was applied to the number of joints 8. The number of attributes of the data with number of joints 8, before filtering process is 547 attributes with sum of weight data is 375. The result of filtering process, there are 547 attributes with sum of weight data is 375, before filtering process is 547 attributes with sum of weight data is 375, the completed results are shown in Appendix B.

The next step after the data filtering using feature selection is the process of classification using ANN. Table 5.13 shows the number of joints 8 that use CorrAttributeEval and classifier ANN, the result is variation of node in hidden layer is 150, and accuracy is 91.56 %, learning rate is at 0.06, momentum coefficients is at 0.06, epoch is 98 and time is 2.515 sec. the completed results are shown in Appendix B.

Table 5.13 The results of experiment 3 (8 joint with CorrAttributeEval)

Number node of hidden layer	Learning rate (lr)	Momentum coefficients (mc)	Accuracy (%)	Epoch	Time
25	0.05	0.08	83.33	295	5.600
50	0.06	0.06	84.44	221	4.333
75	0.06	0.08	85.56	329	8.992
100	0.07	0.07	83.33	300	9.476
125	0.06	0.08	82.78	169	5.257
150	0.06	0.06	91.56	98	2.515
175	0.07	0.06	83.33	142	5.828
200	0.07	0.08	83.89	102	4.726
225	0.06	0.08	85.00	238	11.350
250	0.07	0.06	84.44	208	13.947
275	0.05	0.07	82.22	91	6.423
300	0.07	0.06	81.67	543	40.860
325	0.04	0.07	84.44	203	21.743
350	0.05	0.08	82.22	470	45.612
375	0.04	0.08	82.78	188	17.488
400	0.05	0.07	80.56	105	10.417
425	0.06	0.08	83.89	215	22.848
450	0.05	0.06	81.11	264	39.560
475	0.07	0.07	81.11	151	18.033
500	0.06	0.06	83.33	466	56.722

Table 5.14 shows the results of experiments by CorrAttributeEval with different number of joints. The results of accuracy rate on the number of joints 6 is 86.67 %, the number of joints 7 is 88.33 % and the number of joints 8 is 91.56 %.

Table 5.14 The comparison result experiment 3 (CorrAttributeEval with different number of joints)

Number of Joint	Accuracy (%)	Epoch	Time (sec)	Number node of hidden layer	Learning Rate (lr)	Momentum Coefficients (mc)
6 Joint	86.67	239	10.659	225	0.06	0.08
7 Joint	88.33	113	2.168	75	0.05	0.07
8 Joint	91.56	89	2.515	150	0.06	0.06

The comparative diagram of the result experiment 3 (CorrAttributeEval + ANN with number of joint 6, 7 and 8) are shown in Figure 5.10.

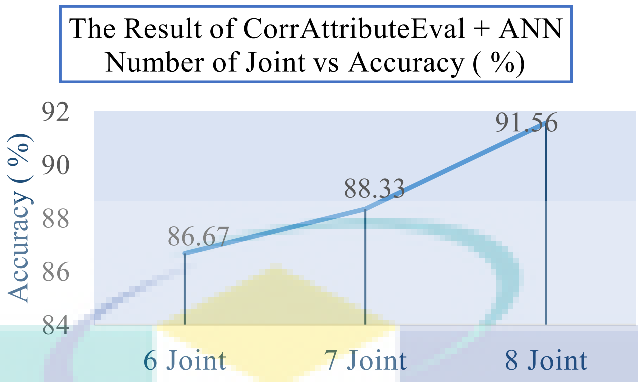


Figure 5.10. Comparative diagram of CorrAttributeEval+ANN with different number of joints

5.3.4 Result of Experiment 4

The next experiment applied CfsSubsetEval. In this, while search method to find the best set of features used BestFirst.

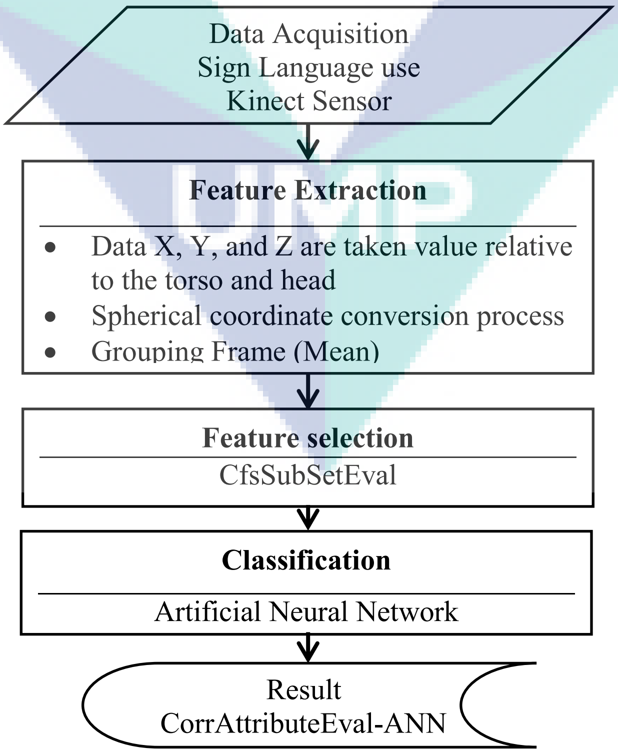


Figure 5.11 Flowchart system of experiment 4

The number of attributes before filtering process is 547 attributes with sum of weight data is 375. The result of filtering process are 45 attributes, with sum of weight data is 375, the completed results are shown in Appendix B.

Table 5.15 shows the number of joints 6 that use CfsSubsetEval and classifier ANN, the result of the variation of node in hidden layer is 100, and accuracy is 91.85 %, learning rate is at 0.06, momentum coefficients is at 0.07, epoch is 541 and time is 5.425 sec. the completed results are shown in Appendix B.

Table 5.15 The results of experiment 4 (6 joint with CfsSubsetEval)

Number node of hidden layer	Learning rate (lr)	Momentum coefficients (mc)	Accuracy (%)	Epoch	Time
25	0.07	0.06	65.00	99	1.063
50	0.07	0.08	75.00	151	1.747
75	0.07	0.06	83.33	163	1.994
100	0.06	0.07	91.85	541	5.425
125	0.06	0.07	88.33	364	6.761
150	0.06	0.08	88.33	517	8.271
175	0.07	0.06	87.89	868	15.228
200	0.06	0.06	88.89	3214	59.982
225	0.06	0.08	87.22	2019	39.733
250	0.05	0.08	87.78	1497	31.450
275	0.05	0.06	90.56	534	13.154
300	0.07	0.08	86.11	278	7.082
325	0.07	0.06	87.22	776	19.025
350	0.05	0.07	90.56	1547	42.456
375	0.05	0.06	86.11	144	44.523
400	0.07	0.06	89.44	969	28.255
425	0.07	0.08	86.67	985	32.401
450	0.04	0.06	85.56	207	6.864
475	0.05	0.07	84.44	5000	179.724
500	0.06	0.06	85.00	260	8.947

The next experiment was applied on the number of joints 7. The number of attributes of the data with number of joints 7, before filtering process is 547 attributes with sum of weight data is 375. The result of filtering process, there are 40 attributes with sum of weight data is 375, before filtering process is 547 attributes with sum of weight data is 375, the completed results are shown in Appendix B.

Table 5.16 shows the number of joints 7 that use CfsSubsetEval and classifier ANN, the result of the variation of node in hidden layer is 125, and accuracy is 93.67%, learning rate is at 0.07, momentum coefficients is at 0.06, epoch is 1445 and time is 6.665 sec. the completed results are shown in Appendix B.

Table 5.16 The results of experiment 4 (7 joint with CfsSubsetEval)

Number node of hidden layer	Learning rate (lr)	Momentum coefficients (mc)	Accuracy (%)	Epoch	Time
25	0.06	0.07	66.56	158	0.988
50	0.06	0.08	80.89	265	3.577
75	0.06	0.08	83.33	153	2.024
100	0.07	0.07	83.56	570	4.299
125	0.07	0.06	93.67	1445	6.665
150	0.07	0.06	87.56	558	7.426
175	0.04	0.08	86.33	377	7.890
200	0.06	0.07	91.11	2403	45.666
225	0.06	0.08	87.56	486	9.815
250	0.06	0.06	87.78	350	5.448
275	0.05	0.06	88.33	695	13.272
300	0.06	0.06	86.67	338	8.206
325	0.04	0.06	83.33	160	14.801
350	0.04	0.07	86.11	267	6.722
375	0.05	0.06	87.22	830	27.899
400	0.07	0.08	85.56	147	3.430
425	0.07	0.08	86.67	789	27.813
450	0.06	0.07	86.11	220	6.544
475	0.06	0.08	85.56	418	12.550
500	0.05	0.06	86.67	190	6.556

The next experiment was applied on the number of joints 8. The number of attributes of the data with number of joints 8, before filtering process is 547 attributes with sum of weight data is 375. The results of filtering process, there are 53 attributes with sum of weight data is 375, before filtering process is 547 attributes with sum of weight data is 375, the completed results are shown in Appendix B.

Table 5.17 shows the number of joints 8 that use CfsSubsetEval and classifier ANN, the results of the variation of node in hidden layer is 75, and accuracy is 96.56%, learning rate is at 0.06, momentum coefficients is at 0.07, epoch 499 and time is 3.699 sec. the completed results are shown in Appendix B.

Table 5.17 The results of experiment 4 (8 joint with CfsSubsetEval)

Number node of hidden layer	Learning rate (lr)	Momentum coefficients (mc)	Accuracy (%)	Epoch	Time
25	0.07	0.07	75.56	101	1.403
50	0.07	0.06	86.67	243	2.773
75	0.06	0.07	96.56	499	3.699
100	0.05	0.08	86.11	436	5.883
125	0.07	0.08	88.89	591	5.506
150	0.07	0.07	86.89	673	4.571
175	0.06	0.06	86.67	742	8.546
200	0.07	0.07	87.89	906	12.275
225	0.06	0.08	87.78	495	7.380
250	0.04	0.08	86.11	184	4.978
275	0.05	0.08	86.11	2144	51.079
300	0.05	0.06	85.56	1438	39.013
325	0.05	0.06	86.67	151	3.450
350	0.07	0.06	85.00	355	9.137
375	0.07	0.07	85.00	176	4.742
400	0.07	0.06	84.44	101	2.027
425	0.04	0.08	85.56	792	25.569
450	0.07	0.08	85.56	99	3.572
475	0.07	0.06	84.56	276	9.465
500	0.06	0.06	83.44	825	10.317

Table 5.18 shows the results of experiments 4 by CfsSubsetEval with different number of joints. The results of accuracy rate on the number of joints 6 is 91.85 %, the number of joints 7 is 93.67 % and the number of joints 8 is 96.56 %.

Table 5.18 The comparison result experiment 4 (CfsSubsetEval)

Number of Joint	Accuracy (%)	Epoch	Time (sec)	Number node of hidden layer	Learning Rate (lr)	Momentum Coefficients (mc)
6 Joint	91.85	541	5.425	100	0.06	0.07
7 Joint	93.67	1445	6.665	125	0.07	0.06
8 Joint	96.56	499	3.699	75	0.06	0.07

The comparative diagram of the result in experiment 4 (CfsSubsetEval + ANN with number of joint 6, 7 and 8) are shown in Figure 5.12.

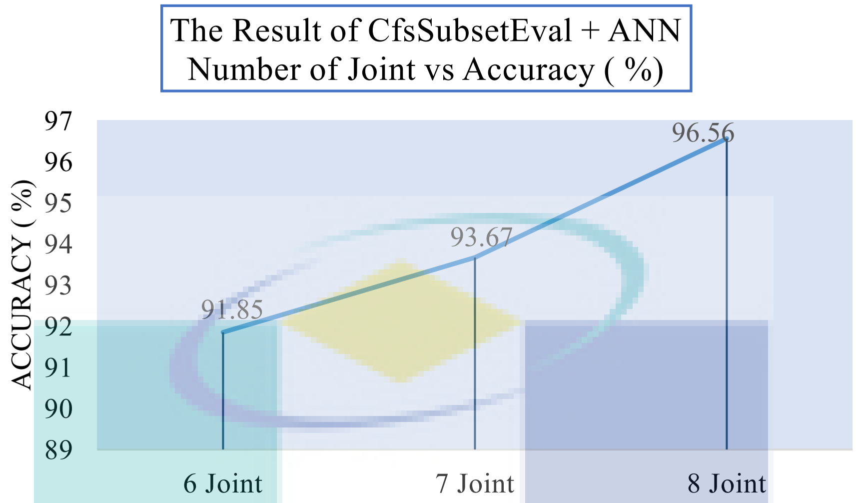


Figure 5.12. Comparative diagram of CfsSubsetEval with different number of joint

The following displayed the results of experiments by different number of joints and classification methods. Table 5.19 shows the results of experiments by six of joints position and different method of feature selection and classification. The results of accuracy rate on recognition using ANN is 85.56 %, CorrAttributEval is 86.67 %, CSE is 89.56 %, and CfsSubsetEval is 91.85 %. Therefore, the use of feature selection could improve the value of accuracy of recognition process.

Table 5.19 The result experiment with six of joints position

Number of Joint	Accuracy (%)	Epoch	Time (sec)	Number node of hidden layer	Learning Rate (lr)	Momentum Coefficients (mc)
CfsSubsetEeval + ANN	91.85	541	5.425	100	0.06	0.07
CFE + ANN	89.56	434	6.154	275	0.05	0.06
CorrAttributEval + ANN	86.67	239	10.659	225	0.06	0.08
ANN	85.56	100	4.502	100	0.05	0.08

The results of experiments with seven of joints position and different method of feature selection and classification are shown in Table 5.20. The results of accuracy rate on recognition using ANN is 87.11 %, CorrAttributEval is 88.33 %, CSE is 90.56 %, and

CfsSubsetEval is 93.67 %. Therefore, the use of feature selection could improve the value of accuracy of recognition process.

Table 5.20 The result experiment with seven of joints position

Number of Joint	Accuracy (%)	Epoch	Time (sec)	Number node of hidden layer	Learning Rate (lr)	Momentum Coefficients (mc)
CfsSubsetEeval + ANN	93.67	1445	6.665	125	0.07	0.06
CFE + ANN	90.56	1403	5.666	200	0.06	0.07
CorrAttributEval + ANN	88.33	113	2.168	75	0.05	0.07
ANN	87.11	259	5.086	75	0.05	0.06

Table 5.21 shows the results of experiments with eight of joints and different method of feature selection and classification. The results of accuracy rate on recognition is 90.56 %, CorrAttributEval is 91.56 %, CSE is 93.67 %, and CfsSubsetEval is 96.56 %. Therefore, the use of feature selection could improve the value of accuracy of recognition process.

Table 5.21 The result experiment with eight of joints position

Number of Joint	Accuracy (%)	Epoch	Time (sec)	Number node of hidden layer	Learning Rate (lr)	Momentum Coefficients (mc)
CfsSubsetEeval + ANN	96.56	499	3.699	75	0.06	0.07
CFE + ANN	93.67	691	8.706	125	0.07	0.08
CorrAttributEval + ANN	91.56	98	2.515	150	0.06	0.06
ANN	90.56	238	11.034	175	0.06	0.07

Figure 5.13 is a comparison chart of on the whole experimental results based on a recognition method of malaysian sign language (ANN, CFS, CFE and CfsSubsetEval) and different number of joint (6, 7 and 8 joint).

Figure 5.13. Comparative diagram of result the experiment

5.4 CONCLUSIONS

The results of experiments with a varying number of joint (6 joint, 7 joint, 8 joint) showed the feature selection algorithms can contribute to improve the accuracy of sign language recognition. The Correlation-based Feature Subset Evaluation (CfsSubsetEval) algorithm produces the best accuracy rate. Performance of the Correlation-based Feature Subset Evaluation (CfsSubsetEval) with ANN could improve the accuracy of dynamic malaysian sign language recognition.

CHAPTER 6

CONCLUSIONS AND FUTURE WORKS

6.1 CONTRIBUTIONS OF THE PRESENT WORK

The present work involves the following contribution:

1. Image acquisition and skeletal feature using RGB and depth sensor to obtain joints positions. This study uses 8 of the 20 joints.

   Several Malaysian Sign Languages have been observed in the dictionary. Six of the 20 joints that contribute in identifying the movement were determined are the left hand, right hand, left wrist, right wrist, left elbow and right elbow. The joints of head and torso were also taken into consideration for the process of normalization of the variations in size and distance from the position of the user. Thus, the number of joints used are eight out of the 20 joints. Using of six joints and seven joints can not handle sign language that has more complex characteristics, such as the movement of the wrist and fingers. Therefore, by increasing number of joints, aids improvement significantly for Sign Language Translator.

   In this study, the results showed that by using templates to 8 the number of joints feature recognition rate better than 6 and 7 joints. This indicates that the more the joint will be able to handle the number of sign language with complex characteristics and better accuracy.

   Simple and effective features extraction are essential to achieve an accurate recognition system. The skeleton extraction is essential for general shape representation and will affect the system performance and algorithm's complexity.

The joints of the tracked users in space can be located, and their movements can be followed. The skeletal tracking recognizes users in standing or sitting position, and facing the sensor of the camera.

For the sample data of dynamic Malaysian Sign Language, image acquisition was obtained using kinect sensor cameras and the data was recorded from five different people. The subjects were asked to perform the same sign repeatedly for five times. Samples of data in the form of X, Y, and Z were obtained from Kinect. There are 375 data samples of Malaysian Sign Language in total.

2. Proposed Feature Selection method using Correlation-based Feature Subset Evaluation (CfsSubsetEval).
   Correlation-based Feature Subset Evaluation (CfsSubsetEval) algorithm produces the best accuracy rate compared with Consistency-based Subset Evaluation (CFE) and Correlation-based Attribute Evaluation (CorrAttributEval) by 96.56 % accuracy rate.

3. Combining the Correlation-based Feature Subset Evaluation (CfsSubsetEval) with Artificial Neural Network for improving the accuracy of Dynamic Malaysian Sign Language Recognition.

In general performance from a combined Feature Selection evaluation with Artificial Neural Network for Dynamic Malaysian Sign Language Recognition can contribute to improving the accuracy of sign language recognition.

6.2 FUTURE WORKS

Many points can be suggested to improve this study. The improvement can be in the feature extraction, feature selection or in the classification stage. Below are some suggested future works:

1. The number of sign can be added, it might be able to improve the accuracy.
2. Development of some of the factors which can lead to a better accuracy rate can be considered such as the data capturing process, the position of the kinect and speed of participants in demonstrating.

3. Using different classification methods such as Hidden Markov Model (HMM, Dynamic Time Wraping (DTW) and Support Vector Machine (SVM).

REFERENCES

Adithya, V. R, V.P. and Gopalakrishnan, U. (2013). Artificial neural network based method for Indian sign language recognition. *IEEE Conference on Information and Communication Technologies (ICT 2013)* pp.1080–1085. Accessed from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6558259

Admasu, Y.F. and Raimond, K. (2010). Ethiopian sign language recognition using Artificial Neural Network. *Intelligent Systems Design and Applications (ISDA)* pp.995–1000. Conference Proceedings.

Agarwal, A. and Thakur, M.K. (2013). Sign language recognition using Microsoft Kinect. *Sixth International Conference on Contemporary Computing (IC3)* pp.181–185.

Aggarwal, M. and Amrita. (2013). Performance Analysis Of Different Feature Selection Methods In Intrusion Detection. *International Journal Of Scientific & Technology Research*, *2*(6), 225–231.

Akmeliawati, R. Ooi, M.P.-L. and Ye Chow Kuang (2007). Real-time Malaysian sign language translation using colour segmentation and neural network. *Instrumentation and Measurement Technology Conference* pp.1–6. Conference Proceedings.

Al-Rousan, M. Assaleh, K. and Tala'a, A. (2009). Video-based signer-independent Arabic sign language recognition using hidden Markov models. *Applied Soft Computing*, *9*(3), 990–999. Journal Article.

Asriani, F. and Susilawati, H. (2010). Static Hand Gesture Recognition of Indonesian Sign Language System Based on Backpropagation Neural Networks. *Makara Seri Teknologi*, *14*(2), 150–154.

Athavale, S. and Deshmukh, M. (2014). Dynamic Hand Gesture Recognition for Human Computer interaction ; A Comparative Study. *International Journal of Engineering Research and General Science*, *2*(2), 38–55.

Auephanwiriyakul, S. Phitakwinai, S. Suttapak, W. Chanda, P. and Theera-Umpon, N. (2013). Thai sign language translation using scale invariant feature transform and hidden markov models. *Pattern Recognition Letters*, *34*(11), 1291–1298.

Bailador, G. Tröster, G. and Roggen, D. (2007). Real time gesture recognition using Continuous Time Recurrent Neural Networks. *International Conference on Body Area Networks.* pp.454–876. Conference Proceedings.

Bartkowiak, A. (2004). *Neural Networks And Pattern Recognition*.

Bellman, R. and Kalaba, R. (1959). On adaptive control processes. *IEEE Transactions on Automatic Control*, *4*(2), 1–9.

Biswas, K.K. and Basu, S.K. (2011). Gesture recognition using Microsoft Kinect. *The 5th International Conference on Automation Robotics and Applications* pp.100–103.

Braem, P.B. (1995). *Einführung in die Gebärdensprache und ihre Erforschung*. S. Prillwitz & T. Vollhaber, Eds. Popp-Druck, Langenhagen Printed in Germany.

Brashear, H. Starner, T. Lukowicz, P. and Junker, H. (2003). Using multiple sensors for mobile sign language recognition. *International Symposium on Wearable Computers* pp.45–52.

Capilla, D.M. (2012). *Sign Language Translator using Microsoft Kinect XBOX 360TM*. (Thesis). University of Tennessee, Knoxville - USA.

Celebi, S. Aydin, A.S. Temiz, T.T. and Arici, T. (2013). Gesture Recognition Using Skeleton Data with Weighted Dynamic Time Warping. *8th International Conference on Computer Vision Theory and Applications (VISAPP 2013)* p.Paper #79. Accessed from http://www.visapp.visigrapp.org/Home.aspx?y=2013

Chen, Q. El-Sawah, A. Joslin, C. and Georganas, N.D. (2005). A dynamic gesture interface for virtual environments based on hidden Markov models. *IEEE International Workshop on Haptic Audio Visual Environments and Their Applications*, 1–2.

Chen, Q. Georganas, N.D. and Petriu, E.M. (2008). Hand gesture recognition using Haar-like features and a stochastic context-free grammar. *IEEE Transactions on Instrumentation and Measurement*, *57*(8), 1562–1571.

Chen, X. (2014). *Real-time Action Recognition for RGB- D and Motion Capture Data*. Aalto University School of Science.

Chou, T.-S. Yen, K.K. Luo, J. Pissinou, N. and Makki, K. (2007). Correlation-Based Feature Selection For Intrusion Detection Design. *Military Communications Conference* pp.1–7.

Corporation, M. (2013). *Kinect for Windows - Human Interface Guidelines v1.7*. Microsoft Corporation. Accessed from http://www.microsoft.com/en-us/kinectforwindows/develop/learn.aspx

Corporation, M. (2015). Developing with Kinect for Windows. (online). Retrieved from http://www.microsoft.com/en-us/kinectforwindows on June 25, 2015.

Das, S. (2001). Filters, Wrappers and a Boosting-Based Hybrid for Feature Selection. *18th International Conference on Machine Learning* pp.74–81.

Dash, M. and Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis*, *1*(3), 131–156.

Dash, M. and Liu, H. (2003). Consistency-based search in feature selection. *Artificial Intelligence*, *151*(1–2), 155–176.

Dinakaran, S. and Thangaiah, P.R.J. (2013). Role of Attribute Selection in Classification Algorithms. *International Journal of Scientific & Engineering Research*, *4*(6), 67–

71.

Dogic, S. and Karli, G. (2014). Sign Language Recognition using Neural Networks. *TEM Journal*, *3*(4), 296–301.

Dymarski, P. (2011). *Hidden Markov Models, Theory and Applications*. Accessed from http://www.intechopen.com/books/show/title/hidden-markov-models-theory-and-applications

Gao, W. Wu, J. Wang, C. and Jiyong Gao Jiangqin, Wu Chunli, Wang, M.W. (2000). A continuous Chinese sign language recognition system. *Proceedings. Fourth IEEE International Conference on Automatic Face and Gesture Recognition* pp.428–433. Conference Proceedings.

Garg, P. Aggarwal, N. and Sofat, S. (2009). Vision based hand gesture recognition. *World Academy of Science, Engineering and Technology 49*, 972–977.

Gaus, Y.F.A. and Wong, F. (2012). Hidden Markov Model-Based Gesture Recognition with Overlapping Hand-Head/Hand-Hand Estimated Using Kalman Filter. *2012 Third International Conference on Intelligent Systems Modelling and Simulation*, 262–267.

Gaus, Y.F.A. Wong, F. Chin, R. Porle, R.R. and Chekima, A. (2013). Analysis and Performance Comparison of the Feature Vectors in Recognition of Malaysian Sign Language. *International Journal of Computer and Electrical Engineering*, *5*(1), 65–68.

Gaus, Y.F.B.A. Tze, F.W.H. and Kin, K.T.T. (2011). Feature extraction from 2D gesture trajectory in Malaysian Sign Language recognition. *4th International Conference on Mechatronics: Integrated Engineering for Industrial and Societal Development, ICOM'11* pp.17–19.

Geer, D. (2004). Will gesture-recognition technology point the way? *Computer*, *37*(10), 20–23.

Gilorkar, N.K. and Ingle, M.M. (2014). A Review on Feature Extraction for Indian and American Sign Language. *International Journal of Computer Science and Information Technologies*, *5*(1), 314–318.

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. GEN, Addison Wesley, Reading, Massachusetts.

Haindl, M. Somol, P. Ververidis, D. and Kotropoulos, C. (2006). Feature selection based on mutual correlation. *Progress in Pattern ...* pp.569–577. Accessed from http://link.springer.com/chapter/10.1007/11892755_59

Hall, M. (1999). *Correlation-based feature selection for machine learning*. Waikato University.

Hall, M. a (2000). Feature Selection for Discrete and Numeric Class Machine Learning. *Seventeenth International conference on Machine Learning* pp.1–16.

Hall, M. and Smith, L. a (1999). Feature Selection for Machine Learning : Comparing a Correlation-based Filter Approach to the Wrapper. *The Twelfth International Florida Artificial Intelligence Research Society Conference* pp.235–239.

Hasan, M.M. and Mishra, P.K. (2010). HSV Brightness Factor Matching for Gesture Recognition System. *International Journal of Image Processing (IJIP)*, *4*(5). Journal Article.

Hazewinkel, M. (2001). Spherical coordinates. *Encyclopedia of Mathematics*. Springer.

Imagawa, I. Matsuo, H. Taniguchi, R. Arita, D. Lu, S.L.S. and Igi, S. (2000). Recognition of local features for camera-based sign language recognition system. *15th International Conference on Pattern Recognition. ICPR-2000* pp.849–853.

Iqbal, M. Purnomo, M.H. and Purnama, I.K.E. (2011). Indonesian sign language recognition system based on flex sensors and accelerometer using Dynamic Time Warping. *The 12th Seminar on Intelligent Technology and ITS Aplications*.

Isabelle Guyon and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*, 1157–1182.

Jangyodsuk, P. and Conly, C. (2014). Sign Language Recognition using Dynamic Time Warping and Hand Shape Distance Based on Histogram of Oriented Gradient Features. *Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments* pp.1–6.

Jantawan, B. and Tsai, C. (2014). A Comparison of Filter and Wrapper Approaches with Data Mining Techniques for Categorical Variables Selection. *International Journal of Innovative Research in Computer and Communication Engineering*, *2*(6), 4501–4508. Accessed from www.ijircce.com

Jhon, G.H. Kohavi, R. and Pfleger, K. (2015). Irrelevant Features and the Subset Selection Problem. *Machine Learning: Proceedings of the Eleventh International Conference* pp.121–129.

Karabasi, M. Bhatti, Z. and Shah, A. (2014). A model for real-time recognition and textual representation of malaysian sign language through image processing. *International Conference on Advanced Computer Science Applications and Technologies, ACSAT* pp.195–200.

Karami, A. Zanj, B. and Sarkaleh, A.K. (2011). Persian sign language (PSL) recognition using wavelet transform and neural networks. *Expert Systems with Applications*, *38*(3), 2661–2667.

Kelly, D. (2010). *Computational Models for the Automatic Learning and Recognition of Irish Sign Language*. National University of Ireland, Maynooth.

Khan, R.Z. and Ibraheem, N.A. (2012). aHand Gesture Recognition : A Literature. *International Journal of Artificial Intelligence & Applications (IJAIA)*, *3*(4), 161–174.

Khan, R.Z. and Ibraheem, N.A. (2012). bSurvey on Gesture Recognition for Hand Image Postures. *Computer and Information System*, *5*(3), 110–121.

Koller, D. and Sahami, M. (1996). Toward optimal feature selection. *Proceedings of International Conference on Machine Learning*.

Kumar, K. and Thakur, G.S.M. (2012). Advanced Applications of Neural Networks and Artificial Intelligence: A Review. *International Journal of Information Technology and Computer Science*, *4*(6), 57–68.

Kumar, S. (2004). *Neural Networks: A Classroom Approach*. 2nd ed. New Delhi: Tata McGraw Hill Education Private. Accessed from http://cs.jnu.edu.cn/international/soft/pptchapter08.pdf

Lang, S. Block, M. Rojas, R. and Simon Lang Rojas Raúl, B.M. (2011). *Sign Language Recognition Using Kinect*. (CHAP) L. K. Rutkowski Marcin & R. T. Scherer Ryszard Zadeh, Lotfi Zurada, Jacek, Eds. *Artificial Intelligence and Soft Computing*. Springer Berlin / Heidelberg. Retrieved from http://dx.doi.org/10.1007/978-3-642-29347-4_46

Lang, S. Rojas, R. and Block-Berlitz, M. (2012). *Sign Language Recognition Using Kinect*. Freie Universität Berlin. Retrieved from http://dx.doi.org/10.1007/978-3-642-29347-4_46

Lee, Y.-H. Tsai, C.-Y. and Lee Cheng-Yueh, Y.-H.T. (2009). Taiwan sign language (TSL) recognition based on 3D data and neural networks. *Expert Systems with Applications*, *36*(2, Part 1), 1123–1128. Journal Article.

Lei, H. and Govindaraju, V. (2005). A comparative study on the consistency of features in on-line signature verification. *Pattern Recognition Letters*, *26*(15), 2483–2489.

Li, H. and Greenspan, M. (2007). *Continuous Time-Varying Gesture Segmentation by Dynamic Time Warping of Compound Gesture Models*. (Thesis) *Dept. of Electrical & Computer Engineering*. Queen's University, Kingston, Canada.

Liang, W. Guixi, L. and Hongyan, D. (2015). Dynamic and combined gestures recognition based on multi-feature fusion in a complex environment. *The Journal of China Universities of Posts and Telecommunications*, *22*(2), 81–88.

Liu, H. and Setiono, R. (1996). A probabilistic approach to feature selection - A filter solution. *13th International Conference on Machine Learning* pp.319–327.

Liu, H. and Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, *17*(4), 491–502.

Lowe, D.G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Lungociu, C. (2011). Real time sign language recognition using artificial neural networks. *Studia Univ. Babes-Bolyai, Informatica, LVI*(4), 75–84.

Maarif, H.A.Q. Akmeliawati, R. and Bilal, S. (2012). Malaysian Sign Language Database for Research. *International Conference on Computer and Communication Engineering* pp.3–5.

Mamat, M. Majid, M.A. and Abdullah, R. (2014). MCMSL Translator : Malaysian Text Translator For Manually Coded Malay Sign Language. *IEEE Symposium on Computers & Informatics (ISCI 2014)*.

Maraqa, M. Abu-Zaiter, R. and Manar Maraqa, R.A.-Z. (2008). Recognition of Arabic Sign Language (ArSL) Using Recurrent Neural Networks. *2008 First International Conference on the Applications of Digital Information and Web Technologies (ICADIWT)*, *(ICADIWT 2*, 478–48. Journal Article.

Masitry, A.K. Majid, M.A. Toh, M.Z. Sutarman and Herawan, T. (2013). An Investigation on Learning Performance among Disabled People Using Educational Multimedia Software : A Case Study for Deaf People. *International Journal of Bio-Science and Bio-Technology*, *5*(6), 9–20.

Maung, T.H.H. (2009). Real-Time Hand Tracking and Gesture Recognition System Using Neural Networks., 466–470.

Mauridhi, M.I. Purnama, E. and Purnomo, H. (2011). Pengenalan Bahasa Isyarat Indonesia Berbasis Sensor Accelerometer dan Sensor Flex Menggunakan Dynamic Time Warping. *Teknik Elektro*. Thesis.

McKee, C. (2014). *The Signs of Language Revisited: An Anthology to Honor Ursula Bellugi and Edward Klima (review)*. Hove, United Kingdom: Psychology Press.

Mekala, P. Gao, Y. Fan, J. Davari, A. and Eng, C. (2011). Real-time Sign Language Recognition based on Neural Network Architecture. *Institute of Electrical and Electronics Engineers*, 197–201.

Mitra, S. and Acharya, T. (2007). Gesture Recognition : A Survey. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, *37*(3), 311–324. Journal Article.

Mohandes, M.A. (2013). Recognition of Two-Handed Arabic Signs Using the CyberGlove. *Arabian Journal for Science and Engineering*, *38*(3), 669–677.

Monirul Kabir, M. Monirul Islam, M. and Murase, K. (2010). A new wrapper feature selection approach using neural network. *Neurocomputing*, *73*(16–18), 3273–3283.

Munib, Q. Habeeb, M. Takruri, B. and Al-Malik, H.A. (2007). American sign language (ASL) recognition based on Hough transform and neural networks. *Expert Systems with Applications*, *32*(1), 24–37.

Nasser Dardas and Georganas, N.D. (2011). Real-Time Hand Gesture Detection and Recognition Using Bag-of-Features and Support Vector Machine Techniques. *IEEE Transactions on Instrumentation and Measurement*, *60*(11), 3592–3607.

Ng, C.W. and Ranganath, S. (2002). Real-time gesture recognition system and application. *Image and Vision Computing*, *20*(13–14), 993–1007. Journal Article.

Nguyen, T. Huynh, H. and Meunier, J. (2015). Static Hand Gesture Recognition Using Principal Component Analysis Combined with Artificial Neural Network. *Journal of Automation and Control Engineering*, *3*(1), 40–45.

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

Paulraj, M.P. Yaacob, S. Desa, H. and Majid, W.M.R.W.A. (2009). Gesture recognition system for Kod Tangan Bahasa Melayu (KTBM) using neural network. *2009 5th International Colloquium on Signal Processing & Its Applications* pp.19–22. Conference Proceedings.

Principe, J.C. Euliano, N.R. and Lefebvre, W.C. (2000). *Neural and Adaptive Systems: Fundamentals Through Simulations*. Wiley.

Quan, Y. and Jinye, P. (2008). Application of improved sign language recognition and synthesis technology in IB. *IEEE Conference on Industrial Electronics and Applications* pp.1629–1634.

R. G. Gordon Jr. (2005). Ethnologue: Languages of the World. (online). Retrieved from http://www.ethnologue.com/show_language.asp?code=psg on January 3, 2014.

Reyes, M. Domínguez, G. and Escalera, S. (2011). Feature weighting in dynamic time warping for gesture recognition in depth data. *Proceedings of the IEEE International Conference on Computer Vision* pp.1182–1188.

Rusu, R.B. (2015). OpenNI. Open-source SDK for 3D sensors. (online). Retrieved from http://openni.ru/openni-sdk/index.html on June 25, 2015.

Sakar, C.O. Kursun, O. and Gurgen, F. (2012). A feature selection method based on kernel canonical correlation analysis and the minimum Redundancy-Maximum Relevance filter method. *Expert Systems with Applications*, *39*(3), 3432–3437.

Sakhalkar, A. Sanghavi, D. Ajmera, K. Joshi, M. Patil, S. and Rawani, Y. (2014). Gesture Recognition Of American Sign Language : A Neural Network Based Approach. *International Journal Of Science, Engineering And Technology*, *2*(4), 76–83.

Sanchez-Nielsen, E. Hernandez-Tejera, M. and Anton-Canalís, L. (2003). Hand Gesture Recognitionfor Human- Machine Interaction. *Journal of WSCG*, *12*, 1–3. Journal Article.

Sandjaja, I.N. and Marcos, N. (2009). Sign Language Number Recognition. *Fifth International Joint Conference on INC, IMS and IDC*, 1503–1508. Conference Paper.

Santiago-Ramírez, E. González-Fraga, J.Á. and Lázaro-Martínez, S. (2012). Face recognition and tracking using unconstrained non-linear correlation filters. *Procedia Engineering*, *35*, 192–201.

Schalkoff, R.J. (1997). *Artificial Neural Networks Hardcover*. McGraw-Hill Companies.

Scully, P. and Jensen, R. (2011). *Investigating Rough Set Feature Selection for Gene Expression Analysis*. the Aberystwyth University.

Sebban, M. and Nock, R. (2013). A Hybrid Filter/Wrapper Approach of Feature Selection using Information Theory. *Journal of Chemical Information and Modeling*, *53*(9), 1689–1699.

Sharma, K. and Garg, N.K. (2014). Hand Gesture Recognition Using Back Propagation Neural Network and Support Vector Machine. *International Journal of Advances in Computer Science and Communication Engineering (IJACSCE)*, *2*(2), 40–46. Accessed from www.sciencepublication.org

Sharma, M. Pal, R. and Sahoo, A.K. (2014). Indian Sign Language Character Recognition using Neural Networks. *ARPN Journal of Engineering and Applied Sciences*, *9*(8), 40–45.

Shimada, N. Kimura, K. and Shirai, Y. (2001). Real-time 3D hand posture estimation based on 2D appearance retrieval using monocular camera. *Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*. pp.23–30.

Shotton, J. Fitzgibbon, A. Cook, M. Sharp, T. and Finocchio, M. (2011). Real-time human pose recognitiom in parts from single depth images. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, *2*, 7.

Siang, J.J. (2012). *Artificial neural networks and programming using MATLAB*. Yogyakarta: ANDI Publisher.

Stergiopoulou, E. and Papamarkos, N. (2009). Hand gesture recognition using a neural network shape fitting technique. *Engineering Applications of Artificial Intelligence*, *22*(8), 1141–1158. Journal Article.

Swee, T.T. Salleh, S. Ariff, A.K. Ting, C. and Seng, S.K. (2007). Malay Sign Language Gesture Recognition System. *International Conference on Intelligent and Advanced Systems* pp.982–985.

Tan, C. LIM, K.-S. and LAI, W.-K. (1993). Multi-Dimensional Features Reduction of Consistency Subset Evaluator on Unsupervised Expectation Maximization Classifier for Imaging Surveillance Application Centre for Advanced Informatics. *International Journal of Image Processing*, *2*(1), 18–26.

Tang, A.O. Lu, K.E. Wang, Y. Huang, J.I.E. and Li, H. (2015). A Real-Time Hand Posture Recognition System Using Deep Neural Networks. *ACM Transactions on Intelligent Systems and Technology*, *6*(2).

Tanibata, N. Shimada, N. and Shirai, Y. (2002). Extraction of Hand Features for Recognition of Sign Language Words. *Computer-Controlled Mechanical Systems*. Thesis.

Tavari, N. V and Deorankar, P.A. V (2014). Implementation of Neural Network based Hand Gesture Recognition System. *International Journal Of Engineering And Computer Science*, *3*(6), 6395–6401.

Temel, T. (2010). Neuromodeling and natural optimization of nonlinear devices and circuits. *System and Circuit Design for Biologically-Inspired Intelligent Learning* 1st ed., p.326. Hershey PA: IGI Global.

Tin Hninn, H.M. (2009). Real-Time Hand Tracking and Gesture Recognition SystemUsing Neural Networks . Conference Paper, World Academy of Science, Engineering and Technology.

Tiwari, R. (2010). Correlation-based Attribute Selection using Genetic Algorithm. *International Journal*, *4*(8), 28–34.

Vogler, C. and Metaxas, D. (1999). Parallel hidden Markov models for American sign language recognition. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 116–122 vol.1.

Vogler Dimitris, C.M. (2004). *Handshapes and Movements: Multiple-Channel American Sign Language Recognition Gesture-Based Communication in Human-Computer Interaction*. A. V. Camurri Gualtiero, Ed. Book, Springer Berlin Heidelberg.

Wang, X. Jiang, F. and Yao, H. (2008). DTW/ISODATA algorithm and multilayer architecture in sign language recognition with large vocabulary. *Proceedings - 2008 4th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIH-MSP 2008* pp.1399–1402.

Wang, Y. Yang, C. Wu, X. Xu, S. Li, H. and Wang Cheng Wu, Xiaoyu Xu, Shengmiao Li, Hui Xu, Shengmiao Li, Hui, Y.Y. (2012). Kinect Based Dynamic Hand Gesture Recognition Algorithm Research. *Intelligent Human-Machine Systems and Cybernetics (IHMSC)* pp.274–279. Conference Proceedings.

Weisstein, E.W. (2015). Spherical Coordinates. (online). Retrieved from http://mathworld.wolfram.com/SphericalCoordinates.html on May 23, 2015.

Wong, F. Sainarayanan, G. Abdullah, W.M. Chekima, A. Jupirin, F.E. and Gaus, Y.F.A. (2013). Software-based Malaysian sign language recognition. *Advances in Intelligent Systems and Computing*, *182 AISC*, 297–306.

Xingyan, L. (2003). *Gesture Recognition Based On Fuzzy C-Means Clustering Algorithm*. (Thesis) *Department Of Computer Science* . The University Of Tennessee Knoxville.

Xu, J. and Yang, J. (2006). Sign Language Classification Using Search Algorithms.

Yu, L. and Liu, H. (2003). Feature Selection for High-Dimensional Data : A Fast Correlation-Based Filter Solution. *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*.

Zhang, L. Hsieh, J.-C. and Wang, J. (2012). A Kinect-based golf swing classification system using HMM and Neuro-Fuzzy. *Computer Science and Information Processing (CSIP)* pp.1163–1166. Conference Proceedings.

Zhang, T. (2011). Adaptive forward-backward greedy algorithm for learning sparse representations. *IEEE Transactions on Information Theory*, *57*(7), 4689–4708.

APPENDICES

APPENDIX A

LIST OF PUBLICATIONS

1. Oral Presentation

- Sutarman, Mazlina Abdul Majid, Jasni Mohamad Zain.*Vision-Based Sign Language Recognition Systems : A Review*, International Conference on Computer Science and Information Technology, 195-201, 2013/6 Juni, 2013 University Technology of Yogyakarta, Indonesia. Proceeding conference.
- Sutarman, Mazlina Abdul Majid, Jasni Mohamad Zain. *Recognition of Malaysian Sign Language Using Skeleton Data with Neural Network*, International Conference on Science in Information Technology (ICSITech 2015), 27-28 Oct. 2015, Print ISBN: 978-1-4799-8384-1, IEEE Explore, 231-236, DOI: 10.1109/ICSITech.2015.7407809, Yogyakarta, Indonesia. Indexed by Scopus.

2. Journal

- Sutarman, Mazlina Abdul Majid, Jasni Mohamad Zain. *A Review On The Development Of Indonesian Sign Language Recognition System*. International Journal of Computer Science, Vol 9, 2013/09/11, 1496-1505, ISSN Print: 1549-3636, DOI : 10.3844/jcssp.2013.1496.1505, Scientific Journal Rankings (SJR): 0.3. Indexed by Scopus
- Sutarman, Mazlina Abdul Majid, Jasni Mohamad Zain, *Recognition of Dynamic Malaysian Sign Language using Combined Consistency-based Subset Evaluation and Artificial Neural Network*. In submission for Journal of Malaysia Journal Computer Science, Indexed by SSCI, Scopus, Lib.Lit, LISA, LISTA (JCR 2009: IF 0.533, ranked 44/65, Tier 3)

- Ananthi Krishnasami Masitry, Mazlina Abdul Majid1, M. Zulfahmi Toh, Sutarman and Tutut Herawan *An Investigation on Learning Performance among Disabled People Using Educational Multimedia Software: A Case Study for Deaf People.* International Journal of Bio-Science and Bio-Technology, Vol. 5 No. 6 December 2013, ISSN: 2233-7849 pp.9-20 Doi:10.14257/ijbsbt.2013.5.6.02. Index: Scopus

APPENDIX B

EXPERIMENT RESULTS

SIX (6) JOINT POSITIONS - RESULT OF NEURAL NETWORK

Table B.1 *Result of Neural Network with initialisasi variation 25 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	74.44	106	1.513	76.11	85	1.369	69.44	93	1.373
0.05	78.33	77	1.591	58.88	97	1.466	76.11	87	1.248
0.06	74.44	221	3.289	78.88	390	6.374	77.77	104	1.539
0.07	81.11	1144	2.203	71.11	72	1.076	80.00	287	4.477

Table B.2 *Result of Neural Network with initialisasi variation 50 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	84.44	118	2.137	61.11	124	1.509	81.67	70	1.217
0.05	80.56	113	2.143	85.00	92	1.669	78.33	94	3.037
0.06	85.00	269	5.021	77.78	70	1.236	78.33	58	1.084
0.07	75.00	67	1.212	81.67	117	2.184	75.00	106	1.950

Table B.3 *Result of Neural Network with initialisasi variation 75 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	83.89	234	4.852	75.52	52	1.246	83.33	136	2.964
0.05	79.44	201	4.681	83.33	289	8.668	79.44	165	3.594
0.06	83.89	125	2.788	80.00	132	3.286	76.78	123	2.837
0.07	81.11	118	2.570	79.44	99	2.097	83.33	123	2.792

Table B.4 *Result of Neural Network with initialisasi variation 100 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	85.00	392	9.968	77.78	169	2.430	83.33	128	3.166
0.05	85.00	158	4.193	78.89	133	3.374	82.22	274	24.837
0.06	77.22	118	3.069	82.22	203	7.495	82.22	134	5.082
0.07	85.00	284	7.082	83.89	301	7.580	68.89	63	1.653

Table B.5 *Result of Neural Network with initialisasi variation 125 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	78.89	74	2.184	84.44	372	11.548	80.00	86	2.527
0.05	85.00	244	7.243	76.67	55	2.067	82.78	181	5.620
0.06	80.56	4.644	68.890	68.89	52	1.739	78.89	83	2.491
0.07	51.11	30	0.909	79.44	76	2.474	78.89	166	5.102

Table B.6 *Result of Neural Network with initialisasi variation 150 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	80.56	217	7.192	72.22	74	2.800	81.11	181	6.022
0.05	82.78	194	6.531	81.11	157	5.249	83.89	130	4.586
0.06	78.33	67	2.244	82.22	219	7.550	73.89	95	3.188
0.07	80.00	119	4.310	81.67	119	4.331	82.22	68	2.387

Table B.7 *Result of Neural Network with initialisasi variation 175 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	81.67	92	3.479	82.78	179	7.296	76.67	45	1.685
0.05	78.89	164	6.459	83.33	189	12.038	71.67	65	2.727
0.06	77.78	62	2.281	77.78	77	3.011	80.56	56	2.088
0.07	77.22	85	3.173	82.78	159	6.258	84.44	72	2.761

Table B.8 *Result of Neural Network with initialisasi variation 200 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	78.33	194	8.471	55.56	45	2.154	60.00	45	1.903
0.05	60.00	56	2.324	77.22	86	3.679	77.78	84	6.202
0.06	80.56	125	5.001	62.22	52	2.246	81.67	137	5.376
0.07	75.56	77	3.238	83.33	96	4.111	72.78	90	1.792

Table B.9 *Result of Neural Network with initialisasi variation 225 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	80.00	141	6.381	82.22	129	6.416	76.11	78	3.682
0.05	82.78	224	11.154	80.56	81	3.793	80.00	169	7.648
0.06	67.78	74	3.380	79.44	65	3.212	75.56	69	3.102
0.07	77.78	60	2.802	81.11	177	10.764	80.56	152	7.316

Table B.10 *Result of Neural Network with initialisasi variation 250 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	81.67	145	7.956	73.33	69	3.660	83.33	139	6.958
0.05	78.89	181	9.641	79.44	131	6.645	83.33	155	7.546
0.06	81.11	96	4.709	83.89	140	7.161	78.33	69	3.321
0.07	76.11	121	6.104	82.22	186	10.330	83.89	226	11.934

Table B.11 *Result of Neural Network with initialisasi variation 275 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	78.89	129	8.471	83.89	197	13.378	80.56	457	30.124
0.05	82.78	151	16.668	81.11	311	20.693	77.78	126	8.221
0.06	69.44	57	3.886	76.67	99	6.507	78.33	101	6.680
0.07	78.89	94	6.314	70.56	108	7.847	77.78	76	5.195

Table B.12 *Result of Neural Network with initialisasi variation 300 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.22	137	9.484	80.56	167	12.034	76.11	65	4.478
0.05	65.00	47	3.661	84.44	266	19.512	76.67	76	5.317
0.06	80.00	101	7.106	85.56	139	11.190	82.22	179	12.447
0.07	68.33	75	5.277	81.11	124	9.494	68.33	57	4.165

Table B.13 *Result of Neural Network with initialisasi variation 325 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	76.11	55	4.867	80.00	120	9.287	82.22	131	9.688
0.05	78.89	76	6.542	78.33	123	9.098	80.56	101	14.004
0.06	81.67	114	8.563	80.56	243	17.862	75.00	100	7.346
0.07	79.44	109	8.201	76.11	106	7.800	76.11	76	5.944

Table B.14 *Result of Neural Network with initialisasi variation 350 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	80.56	230	18.127	77.22	137	11.639	80.00	102	8.018
0.05	78.44	115	10.681	77.20	114	9.335	76.67	88	8.292
0.06	85.56	100	4.502	72.22	94	10.795	63.89	41	3.314
0.07	80.00	198	15.749	74.44	101	7.910	55.56	32	2.683

Table B.15 *Result of Neural Network with initialisasi variation 375 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	44.44	34	2.886	81.67	221	20.579	70.56	62	5.226
0.05	76.67	116	10.264	63.33	56	4.802	83.89	271	24.606
0.06	58.89	55	4.755	75.56	102	11.606	78.33	118	9.877
0.07	66.11	48	4.114	69.44	104	8.705	53.89	41	3.697

Table B.16 *Result of Neural Network with initialisasi variation 400 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	81.67	208	18.689	79.44	105	10.855	77.22	93	8.237
0.05	76.67	78	78.555	70.56	76	6.876	79.44	311	29.081
0.06	67.78	55	4.966	80.56	180	17.918	39.44	37	3.302
0.07	81.11	223	20.136	56.11	48	4.290	71.67	90	8.690

Table B.17 *Result of Neural Network with initialisasi variation 425 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	76.67	148	13.946	78.89	137	15.708	70.00	48	4.649
0.05	81.67	184	18.611	80.56	173	16.695	78.89	98	9.641
0.06	73.33	96	9.226	82.22	240	22.658	54.44	61	5.776
0.07	71.11	88	8.483	70.56	67	6.412	65.00	68	6.864

Table B.18 *Result of Neural Network with initialisasi variation 450 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	51.67	43	4.306	64.44	59	6.182	53.89	45	4.540
0.05	76.67	122	12.792	77.78	94	10.066	68.89	68	6.884
0.06	75.56	150	15.054	78.33	180	18.063	78.33	94	9.464
0.07	61.11	65	6.567	77.78	313	21.135	72.78	82	8.595

Table B.19 *Result of Neural Network with initialisasi variation 475 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	78.89	159	16.645	68.89	51	5.578	75.56	92	9.656
0.05	79.44	232	24.788	82.22	1431	157.228	83.89	146	15.589
0.06	82.22	144	15.594	83.89	302	37.327	75.00	131	13.938
0.07	78.89	288	30.822	82.22	210	22.979	80.56	358	39.327

Table B.20 *Result of Neural Network with initialisasi variation 500 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	48.89	43	4.711	66.67	51	5.831	82.78	277	30.389
0.05	82.22	202	22.589	75.56	120	14.075	76.11	120	13.273
0.06	77.22	143	15.781	43.89	41	6.532	56.67	58	6.398
0.07	82.22	193	21.199	76.67	148	16.973	33.89	32	3.728

SIX (6) JOINT POSITIONS - RESULT OF CORRELATION + NEURAL NETWORK

Table B.21 *Result of Correlation+ Neural Network with initialisasi variation 25 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	77.78	107	1.498	74.44	82	1.265	76.67	246	3.588
0.05	76.67	79	2.622	75.56	92	1.311	73.89	111	1.591
0.06	80.00	81	1.213	76.11	124	2.328	50.56	44	0.693
0.07	73.89	71	1.028	67.22	68	1.025	71.67	189	3.260

Table B.22 *Result of Correlation+ Neural Network with initialisasi variation 50 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.22	148	2.511	78.89	68	1.327	77.22	82	1.467
0.05	78.33	116	2.316	82.22	217	6.069	81.67	91	2.416
0.06	80.00	112	2.206	83.33	269	5.212	78.89	117	2.194
0.07	66.11	55	1.011	78.89	84	1.532	82.22	217	4.041

Table B.23 *Result of Correlation+ Neural Network with initialisasi variation 75 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	80.00	145	3.120	75.89	79	1.928	77.78	69	1.545
0.05	85.56	104	2.451	78.33	105	2.262	82.22	74	1.651
0.06	80.56	91	2.138	81.67	403	9.032	76.11	56	1.268
0.07	82.78	150	3.259	85.00	286	6.026	76.67	156	3.557

Table B.24 *Result of Correlation + Neural Network with initialisasi variation 100 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	78.33	100	2.480	76.67	70	1.890	84.44	381	9.516
0.05	52.22	31	2.261	82.78	140	3.583	81.67	99	2.422
0.06	79.44	67	1.697	82.78	162	4.061	86.11	93	2.495
0.07	72.22	180	4.454	83.33	72	2.197	73.33	87	2.246

Table B.25 *Result of Correlation + Neural Network with initialisasi variation 125 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	77.78	68	1.981	81.67	486	15.610	80.00	101	3.027
0.05	81.67	112	4.270	81.67	106	3.345	80.56	115	3.905
0.06	83.89	650	18.643	76.11	83	4.159	82.22	98	2.891
0.07	74.44	52	1.530	79.44	59	1.700	56.67	48	1.482

Table B.26 *Result of Correlation + Neural Network with initialisasi variation 150 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	81.11	92	3.088	80.00	127	4.652	82.78	175	5.850
0.05	77.22	89	3.039	82.22	135	5.230	72.78	66	2.152
0.06	85.56	200	6.679	81.67	111	3.868	80.00	132	4.288
0.07	82.22	185	6.210	77.78	103	3.852	82.78	110	3.822

Table B.27 *Result of Correlation + Neural Network with initialisasi variation 175 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	83.33	175	6.459	82.22	100	4.050	68.33	50	1.888
0.05	80.56	62	2.948	83.89	97	6.501	73.33	49	2.249
0.06	69.44	58	2.163	80.00	107	4.119	78.89	138	5.501
0.07	83.89	233	8.772	79.44	84	3.652	78.89	176	6.801

Table B.28 *Result of Correlation + Neural Network with initialisasi variation 200 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	81.67	154	6.210	78.33	93	4.275	81.67	239	9.968
0.05	81.67	90	3.635	79.44	89	5.501	81.67	93	5.177
0.06	78.33	69	2.844	51.67	44	1.856	68.89	63	2.525
0.07	81.67	105	4.408	85.56	107	4.434	77.22	79	3.417

Table B.29 *Result of Correlation + Neural Network with initialisasi variation 225 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	81.67	171	7.566	80.00	258	13.211	83.33	310	14.165
0.05	81.67	156	7.301	77.22	81	3.719	76.11	107	4.832
0.06	77.78	166	7.506	69.44	62	3.091	86.67	239	10.659
0.07	52.22	34	1.636	78.33	122	8.975	76.67	108	5.195

Table B.30 *Result of Correlation + Neural Network with initialisasi variation 250 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	80.00	88	4.508	81.67	120	6.968	78.89	69	3.650
0.05	83.33	163	8.299	76.11	86	4.356	71.67	89	4.331
0.06	78.33	104	5.274	80.00	180	8.954	72.22	112	5.386
0.07	82.78	326	16.409	79.44	264	14.676	75.00	80	4.119

Table B.31 *Result of Correlation + Neural Network with initialisasi variation 275 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	68.33	54	3.588	71.11	71	4.840	83.33	390	25.662
0.05	79.44	78	7.569	35.00	28	1.829	73.89	61	4.011
0.06	55.00	43	2.994	70.56	99	6.365	80.00	75	4.850
0.07	68.33	73	4.963	78.89	131	9.781	74.44	76	5.210

Table B.32 *Result of Correlation + Neural Network with initialisasi variation 300 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	81.11	121	8.440	80.00	139	9.998	80.56	208	14.367
0.05	77.78	113	9.991	77.22	124	8.598	83.33	384	27.347
0.06	76.67	66	4.613	66.67	69	4.799	82.78	226	15.540
0.07	72.22	88	6.167	57.78	41	2.922	84.44	129	9.344

Table B.33 *Result of Correlation + Neural Network with initialisasi variation 325 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	76.11	107	11.388	71.11	59	4.560	80.56	116	8.549
0.05	72.78	61	4.706	80.56	101	7.889	78.89	202	49.475
0.06	83.33	157	11.608	78.33	108	8.049	82.78	175	12.850
0.07	47.78	36	2.681	77.78	115	8.596	47.22	30	2.371

Table B.34 *Result of Correlation + Neural Network with initialisasi variation 350 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	74.44	90	7.176	74.44	74	6.955	77.22	61	4.790
0.05	57.78	43	3.666	75.00	83	6.851	77.78	117	9.587
0.06	54.44	38	3.054	56.78	48	4.392	74.44	69	5.545
0.07	74.44	112	8.922	66.67	66	5.273	75.56	149	12.543

Table B.35 *Result of Correlation + Neural Network with initialisasi variation 375 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	73.33	59	4.992	77.78	250	24.310	78.33	125	10.467
0.05	75.56	98	8.798	68.89	67	5.720	66.11	49	4.265
0.06	80.00	141	11.921	83.89	272	28.773	80.00	261	22.009
0.07	75.00	143	12.163	72.22	84	7.051	77.78	138	13.728

Table B.36 *Result of Correlation + Neural Network with initialisasi variation 400 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	81.11	341	30.217	74.44	93	9.050	82.22	260	23.197
0.05	80.00	175	18.106	71.67	61	5.862	67.78	54	5.270
0.06	76.11	113	10.123	67.22	65	6.147	80.56	172	15.326
0.07	81.11	416	37.644	75.56	183	16.208	65.56	102	10.437

Table B.37 *Result of Correlation + Neural Network with initialisasi variation 425 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	81.67	468	44.289	75.00	80	9.062	81.11	111	10.670
0.05	76.11	90	8.892	81.11	280	27.108	81.11	130	12.701
0.06	53.89	47	4.534	78.33	165	15.641	81.11	196	18.705
0.07	78.89	144	13.932	73.33	131	12.480	47.22	42	4.306

Table B.38 *Result of Correlation + Neural Network with initialisasi variation 450 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	78.89	167	16.598	76.11	74	7.787	60.56	47	4.711
0.05	80.00	125	14.321	77.78	100	11.189	68.89	67	6.784
0.06	66.67	104	10.448	75.56	124	12.336	76.67	137	13.846
0.07	78.33	179	18.887	77.22	169	17.425	47.22	38	4.212

Table B.39 *Result of Correlation + Neural Network with initialisasi variation 475 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	77.78	89	9.516	76.11	104	11.383	78.89	400	41.870
0.05	76.67	169	18.455	76.67	93	10.543	68.89	57	6.098
0.06	70.56	89	9.544	80.56	209	23.500	77.22	115	12.111
0.07	77.22	157	16.798	81.67	148	16.069	80.56	158	19.812

Table B.40 *Result of Correlation + Neural Network with initialisasi variation 500 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	83.89	114	12.387	78.89	227	26.078	61.67	50	5.522
0.05	75.00	146	16.256	75.56	85	9.373	81.11	328	36.225
0.06	66.11	77	8.520	81.11	222	30.175	77.22	124	13.625
0.07	76.67	192	21.103	76.67	160	18.065	70.56	81	9.345

SIX (6) JOINT POSITIONS - RESULT OF CONSISTENCY + NEURAL NETWORK

Table B.41 *Result of Consistency + Neural Network with Initialisasi variation 25 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	57.78	118	1.981	62.22	164	2.205	61.11	116	1.248
0.05	53.33	109	1.108	60.56	111	1.544	50.00	106	1.139
0.06	62.22	108	1.168	58.33	109	2.423	61.66	116	1.362
0.07	65.00	890	1.063	56.67	100	1.173	54.44	107	1.342

Table B.42 *Result of Consistency + Neural Network with initialisasi variation 50 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	67.78	147	1.654	65.56	139	1.662	68.89	127	1.389
0.05	65.56	127	2.279	67.78	144	2.309	73.33	136	1.607
0.06	58.33	97	1.151	62.22	118	1.325	62.22	98	1.149
0.07	71.67	191	2.123	66.67	120	2.026	75.00	1211	2.347

Table B.43 *Result of Consistency + Neural Network with initialisasi variation 75 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	75.00	186	1.684	73.33	171	2.294	82.78	295	3.635
0.05	78.89	228	3.377	70.56	125	1.904	76.78	208	2.558
0.06	81.11	320	3.893	73.89	156	2.026	77.22	236	3.309
0.07	83.33	160	2.990	76.11	182	2.242	77.78	165	2.121

Table B.44 *Result of Consistency + Neural Network with initialisasi variation 100 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	80.00	191	2.667	82.78	340	5.486	77.22	258	3.479
0.05	80.56	281	3.901	82.22	229	3.070	72.78	124	1.653
0.06	81.67	252	3.446	85.00	341	4.415	81.11	336	4.364
0.07	80.56	410	5.510	82.78	260	3.732	78.33	145	2.028

Table B.45 *Result of Consistency + Neural Network with initialisasi variation 125 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.78	90	2.823	76.67	332	5.161	77.78	197	2.948
0.05	76.67	275	4.952	87.78	333	5.817	83.89	506	7.709
0.06	82.78	325	4.696	88.33	3640	6.761	84.44	275	4.299
0.07	85.00	279	4.144	62.78	96	2.144	90.00	485	7.441

Table B.46 *Result of Consistency + Neural Network with initialisasi variation 150 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	76.11	149	2.340	81.11	457	8.065	84.44	536	8.626
0.05	81.67	488	7.861	77.78	151	2.406	77.22	267	4.214
0.06	83.89	436	6.977	75.56	120	2.059	88.33	1170	8.271
0.07	73.33	129	2.128	71.66	121	2.264	87.22	362	6.193

Table B.47 *Result of Consistency + Neural Network with initialisasi variation 175 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.22	376	6.396	83.89	397	7.611	83.89	470	8.097
0.05	83.89	466	9.360	83.33	376	11.234	68.33	129	2.780
0.06	83.89	501	8.677	85.56	465	8.112	77.22	117	2.286
0.07	87.89	868	15.228	78.89	125	2.208	87.22	400	7.207

Table B.48 *Result of Consistency + Neural Network with initialisasi variation 200 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	78.89	355	6.583	85.00	368	7.643	83.33	907	16.926
0.05	86.67	1209	23.150	80.56	395	7.592	86.67	1380	32.753
0.06	88.89	3014	57.982	81.11	686	12.964	80.56	347	6.325
0.07	86.67	428	8.127	84.44	515	11.883	83.33	327	6.489

Table B.49 *Result of Consistency + Neural Network with initialisasi variation 225 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	86.67	376	7.223	85.56	366	7.966	84.44	325	6.412
0.05	86.67	435	9.173	86.11	386	7.824	82.78	293	5.712
0.06	85.00	893	18.190	83.33	561	14.442	87.22	2009	39.533
0.07	78.89	123	2.483	87.78	429	12.552	86.67	381	8.128

Table B.50 *Result of Consistency + Neural Network with initialisasi variation 250 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	81.11	577	12.870	79.44	146	3.439	78.33	304	6.443
0.05	80.00	154	3.276	81.67	778	16.404	87.78	1297	31.450
0.06	82.22	943	19.315	81.67	826	18.301	85.00	277	5.913
0.07	83.33	162	3.517	85.00	529	14.174	85.56	255	5.709

Table B.51 *Result of Consistency + Neural Network with initialisasi variation 275 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	72.22	93	2.215	89.44	522	12.598	81.67	1180	27.549
0.05	89.56	434	6.154	77.22	111	2.529	85.00	1994	45.378
0.06	84.44	801	22.589	81.67	227	5.226	79.44	312	7.084
0.07	81.11	940	21.969	85.56	431	11.747	84.44	1465	35.209

Table B.52 *Result of Consistency + Neural Network with initialisasi variation 300 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	77.22	173	4.337	83.89	491	12.602	78.89	195	4.696
0.05	83.89	407	12.926	80.00	186	4.638	85.00	275	6.652
0.06	83.89	252	6.119	80.00	177	4.368	84.44	348	8.339
0.07	85.00	748	18.617	82.22	285	7.455	86.11	1278	7.082

Table B.53 *Result of Consistency + Neural Network with initialisasi variation 325 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	85.56	519	22.776	81.11	521	12.597	79.44	256	6.068
0.05	81.11	392	12.017	86.11	499	12.820	85.00	261	15.132
0.06	80.56	833	20.252	78.89	110	2.714	86.67	345	8.386
0.07	87.22	776	19.025	77.22	90	3.463	85.56	643	16.832

Table B.54 *Result of Consistency + Neural Network with initialisasi variation 350 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.22	171	4.384	77.22	145	4.066	76.11	188	4.789
0.05	81.66	146	3.974	87.56	1547	12.456	71.11	109	0.138
0.06	82.22	386	9.966	83.33	108	3.260	81.11	219	5.648
0.07	77.22	80	2.074	83.89	408	10.545	85.56	352	9.532

Table B.55 *Result of Consistency + Neural Network with initialisasi variation 375 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	85.00	412	11.232	84.44	370	10.619	79.44	221	6.053
0.05	86.11	144	44.523	83.89	171	4.739	82.22	484	15.083
0.06	85.56	1174	31.393	85.00	329	10.798	85.00	599	16.327
0.07	83.33	3470	94.938	69.44	79	2.153	79.44	97	2.777

Table B.56 *Result of Consistency + Neural Network with initialisasi variation 400 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	83.33	610	17.675	78.33	182	5.666	84.44	175	5.055
0.05	83.89	399	23.172	83.33	667	19.483	81.11	173	5.242
0.06	86.66	619	18.090	79.44	111	3.292	85.56	978	28.198
0.07	89.44	969	28.255	77.78	106	3.074	77.78	108	3.339

Table B.57 *Result of Consistency + Neural Network with initialisasi variation 425 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	83.33	765	23.213	77.22	152	5.415	81.11	270	8.174
0.05	78.89	272	8.611	83.33	436	13.341	82.78	526	16.878
0.06	83.33	1339	41.208	78.89	184	5.676	78.89	99	3.022
0.07	77.22	127	3.911	83.33	147	4.493	86.67	985	32.401

Table B.58 *Result of Consistency + Neural Network with initialisasi variation 450 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	85.56	207	6.864	82.78	322	10.707	57.22	70	2.059
0.05	84.44	720	27.690	85.00	406	16.302	83.89	448	14.442
0.06	82.78	803	25.484	85.00	1122	38.511	83.89	531	15.543
0.07	84.44	150	4.850	81.11	192	6.381	80.56	117	3.837

Table B.59 *Result of Consistency + Neural Network with initialisasi variation 475 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	80.56	198	6.443	83.33	463	16.132	83.33	809	26.552
0.05	82.78	238	8.807	84.44	5000	179.724	81.11	598	19.996
0.06	81.67	468	15.931	82.78	195	6.740	72.78	100	3.077
0.07	79.44	139	4.639	81.11	1156	39.312	83.33	513	18.439

Table B.60 *Result of Consistency + Neural Network with initialisasi variation 500 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	83.33	1778	60.544	77.22	210	7.722	81.11	217	7.426
0.05	79.44	153	5.335	82.78	215	7.467	85.00	726	25.123
0.06	85.00	260	8.947	82.22	217	8.601	83.89	191	6.494
0.07	82.78	274	9.453	79.44	123	4.399	84.44	231	8.252

SEVEN (7) JOINT POSITIONS - RESULT OF NEURAL NETWORK

Table B.61 *Result of Neural Network with initialisasi variation 25 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	81.11	203	2.964	77.22	70	1.157	71.67	84	1.279
0.05	65.56	53	1.017	82.78	181	2.589	79.44	135	2.075
0.06	72.78	80	1.231	84.44	246	4.600	79.44	180	2.789
0.07	78.89	124	1.941	66.11	78	1.243	65.00	53	0.825

Table B.62 *Result of Neural Network with initialisasi variation 50 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	78.89	94	1.701	77.22	90	1.792	77.22	69	1.264
0.05	77.22	70	1.391	77.22	71	1.373	83.33	143	2.591
0.06	81.11	156	2.838	64.44	58	1.080	83.33	74	1.595
0.07	57.78	53	1.002	71.11	63	1.140	86.67	113	2.168

Table B.63 *Result of Neural Network with initialisasi variation 75 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	86.11	221	5.086	85.00	122	2.964	75.79	104	2.355
0.05	87.11	259	5.086	86.11	358	9.934	81.11	72	1.583
0.06	81.11	170	3.762	86.11	263	5.834	83.33	130	3.024
0.07	82.78	178	3.988	78.33	102	2.239	67.22	48	1.123

Table B.64 *Result of Neural Network with initialisasi variation 100 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.22	89	2.511	76.11	114	3.100	82.78	82	2.075
0.05	75.56	91	3.898	82.22	71	2.295	78.89	68	1.748
0.06	64.44	73	1.907	76.11	118	3.607	77.22	72	2.229
0.07	82.22	116	3.000	83.89	136	3.845	83.89	95	2.511

Table B.65 *Result of Neural Network with initialisasi variation 125 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	74.44	75	2.247	73.89	56	1.845	86.11	199	6.209
0.05	73.33	65	2.112	76.11	135	4.254	81.11	100	3.391
0.06	80.00	70	2.111	77.78	156	5.039	73.89	87	2.649
0.07	69.44	46	1.520	75.56	118	3.599	76.11	87	2.792

Table B.66 *Result of Neural Network with initialisasi variation 150 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	72.78	58	2.013	83.33	104	3.857	83.89	136	4.648
0.05	79.44	62	2.259	76.67	70	2.410	81.11	60	2.105
0.06	81.67	273	9.510	76.11	65	2.340	80.56	99	3.925
0.07	79.44	175	6.157	71.67	100	3.456	81.67	98	3.588

Table B.67 *Result of Neural Network with initialisasi variation 175 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	76.67	56	2.277	82.78	274	13.201	79.44	86	3.370
0.05	83.89	259	12.074	83.33	189	12.038	79.44	76	3.384
0.06	80.56	95	3.772	78.89	92	3.770	81.67	159	6.049
0.07	80.00	213	8.546	78.33	115	4.921	80.56	169	6.942

Table B.68 *Result of Neural Network with initialisasi variation 200 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	79.44	220	9.204	84.44	265	13.335	65.56	59	2.636
0.05	81.67	237	10.608	81.11	178	7.522	67.22	42	1.807
0.06	81.67	153	6.866	80.56	99	4.616	81.11	133	5.622
0.07	80.00	127	5.501	83.33	135	6.492	75.56	89	4.181

Table B.69 *Result of Neural Network with initialisasi variation 225 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	67.22	65	3.026	80.56	76	4.081	81.11	149	7.067
0.05	82.22	148	7.488	77.22	94	4.574	81.67	156	7.589
0.06	73.33	54	2.670	73.89	86	4.719	77.78	69	3.322
0.07	77.78	98	4.909	77.78	125	7.092	80.00	36	1.825

Table B.70 *Result of Neural Network with initialisasi variation 250 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	69.44	61	3.666	73.33	63	3.734	80.00	237	13.588
0.05	76.11	62	4.160	78.89	98	5.615	82.78	222	12.745
0.06	75.00	101	5.743	83.33	387	8.003	81.11	69	3.894
0.07	65.56	63	3.723	69.44	97	6.469	69.44	61	3.650

Table B.71 *Result of Neural Network with initialisasi variation 275 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	80.00	279	20.670	54.44	38	2.741	65.00	45	3.136
0.05	76.11	118	9.543	82.22	145	9.905	80.56	207	14.170
0.06	76.67	74	5.134	78.89	106	7.410	83.33	198	13.394
0.07	79.44	196	13.761	78.33	189	16.126	77.22	124	8.799

Table B.72 *Result of Neural Network with initialisasi variation 300 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	77.78	565	25.382	82.78	81	6.287	80.56	158	11.326
0.05	78.33	69	5.244	76.11	58	4.195	70.56	66	7.889
0.06	79.44	111	8.029	80.56	71	5.102	72.22	76	5.467
0.07	73.33	146	10.673	81.67	180	15.070	77.22	114	8.673

Table B.73 *Result of Neural Network with initialisasi variation 325 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	79.44	99	8.658	71.67	62	6.620	82.22	154	12.028
0.05	73.89	69	5.650	76.11	71	5.646	81.11	181	26.602
0.06	75.00	96	7.500	71.11	75	5.960	67.22	68	5.299
0.07	79.44	111	8.703	78.33	141	10.889	71.11	58	4.773

Table B.74 *Result of Neural Network with initialisasi variation 350 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	81.67	341	28.454	83.33	130	12.294	75.00	101	8.377
0.05	82.78	148	13.822	80.00	238	20.034	81.67	124	16.123
0.06	71.67	71	6.073	77.78	114	10.343	78.33	172	14.625
0.07	75.00	114	9.497	76.11	135	11.107	68.89	70	6.100

Table B.75 *Result of Neural Network with initialisasi variation 375 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	81.11	105	9.360	81.11	162	21.051	82.78	172	15.242
0.05	76.56	104	9.877	80.56	123	11.434	75.56	85	9.469
0.06	73.33	77	6.876	81.11	123	11.653	80.00	92	8.980
0.07	83.89	216	19.422	77.78	125	10.998	68.89	68	6.380

Table B.76 *Result of Neural Network with initialisasi variation 400 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	80.56	179	16.957	77.78	91	9.788	77.22	135	12.698
0.05	70.00	56	5.758	78.89	97	9.574	77.78	199	19.776
0.06	70.00	65	6.138	77.78	85	8.698	83.89	190	17.790
0.07	77.22	139	13.330	62.22	47	4.633	78.33	159	15.584

Table B.77 *Result of Neural Network with initialisasi variation 425 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	80.56	364	35.943	66.67	56	6.280	49.44	35	3.479
0.05	80.00	103	10.483	79.44	269	30.599	83.89	606	60.902
0.06	82.78	218	21.942	81.11	1038	33.259	80.56	232	22.864
0.07	77.78	121	12.248	78.89	155	15.788	60.56	51	5.320

Table B.78 *Result of Neural Network with initialisasi variation 450 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	81.11	70	7.270	68.89	47	5.232	78.89	136	14.133
0.05	76.67	82	8.923	85.00	417	52.079	83.33	302	31.907
0.06	73.89	56	6.082	77.78	111	13.246	78.89	133	13.935
0.07	70.00	92	9.740	32.22	27	2.933	78.89	237	25.928

Table B.79 *Result of Neural Network with initialisasi variation 475 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	77.78	148	16.193	81.67	225	25.650	78.89	183	20.061
0.05	84.44	194	21.746	70.56	81	13.290	81.67	484	57.964
0.06	77.22	217	24.350	73.33	106	14.301	80.56	112	12.370
0.07	78.89	288	30.822	70.56	110	12.511	80.00	109	12.683

Table B.80 *Result of Neural Network with initialisasi variation 500 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.22	165	19.047	78.89	113	13.558	82.22	152	17.504
0.05	83.33	215	25.116	53.89	41	4.788	85.00	562	65.449
0.06	72.22	76	8.889	71.11	91	11.516	71.67	85	10.005
0.07	77.78	131	15.166	77.22	201	23.853	70.00	77	9.328

SEVEN (7) JOINT POSITIONS - RESULT OF CORRELATION + NEURAL NETWORK

Table B.81 *Result of Correlation + Neural Network with initialisasi variation 25 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	70.56	86	1.311	68.89	87	1.427	78.33	114	1.701
0.05	68.89	84	1.457	73.89	93	1.373	66.67	118	1.844
0.06	31.11	33	0.523	70.00	134	19.935	44.44	61	0.937
0.07	31.00	54	0.081	77.78	243	4.861	57.78	60	0.921

Table B.82 *Result of Correlation +Neural Network with initialisasi variation 50 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	64.44	60	1.060	81.11	135	2.684	80.00	174	3.229
0.05	78.89	163	3.379	66.11	53	0.983	81.67	71	1.708
0.06	80.00	78	1.448	54.44	43	0.772	78.89	74	1.474
0.07	74.44	68	1.243	73.89	73	1.401	50.78	144	2.746

Table B.83 *Result of Correlation +Neural Network with initialisasi variation 75 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	85.00	72	1.560	82.78	204	4.980	81.67	163	3.712
0.05	80.00	77	1.910	88.33	113	2.168	86.11	112	2.455
0.06	72.22	74	1.673	85.56	117	2.496	84.44	86	2.019
0.07	83.33	163	3.667	61.67	52	1.119	71.11	89	2.044

Table B.84 *Result of Correlation +Neural Network with initialisasi variation 100 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.22	80	2.168	86.11	120	3.317	83.33	100	2.589
0.05	86.11	79	2.415	83.33	114	3.620	80.56	185	5.171
0.06	81.11	139	3.632	76.67	108	4.159	82.78	71	2.030
0.07	78.89	130	3.340	82.22	155	3.937	77.22	63	1.684

Table B.85 *Result of Correlation +Neural Network with initialisasi variation 125 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	73.33	74	2.247	81.67	78	2.579	80.00	178	5.460
0.05	81.11	79	2.658	82.78	189	5.710	83.89	182	5.820
0.06	78.33	82	2.462	77.22	88	3.051	81.67	217	6.458
0.07	71.67	54	1.654	80.00	91	3.082	82.22	76	2.418

Table B.86 *Result of Correlation +Neural Network with initialisasi variation 150 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	79.44	125	4.305	78.89	281	10.726	83.33	150	5.288
0.05	78.89	148	5.585	83.33	188	6.650	81.67	123	4.138
0.06	83.33	145	5.029	73.33	92	3.198	82.78	160	7.656
0.07	72.22	85	2.995	75.56	100	3.456	83.33	155	5.600

Table B.87 *Result of Correlation +Neural Network with initialisasi variation 175 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	71.67	60	2.496	78.89	93	4.997	79.44	97	3.869
0.05	78.33	161	7.769	83.89	97	6.501	82.22	161	8.637
0.06	75.00	67	2.651	82.22	94	4.234	77.22	109	4.891
0.07	73.89	67	2.710	58.89	42	2.093	80.00	173	7.113

Table B.88 *Result of Correlation +Neural Network with initialisasi variation 200 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	77.78	517	22.308	73.33	81	4.118	81.11	289	12.511
0.05	80.56	138	6.552	76.67	83	4.077	73.89	82	3.492
0.06	82.22	204	8.668	75.56	141	6.346	82.22	132	5.688
0.07	75.00	61	2.724	82.22	75	3.393	81.11	154	7.004

Table B.89 *Result of Correlation +Neural Network with initialisasi variation 225 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.78	111	5.304	76.11	58	3.139	80.00	90	4.446
0.05	76.67	59	2.871	82.22	213	10.345	77.22	73	3.576
0.06	80.56	155	7.366	81.11	116	5.974	77.78	60	2.789
0.07	80.00	157	7.601	83.33	184	12.350	56.11	36	1.825

Table B.90 *Result of Correlation +Neural Network with initialisasi variation 250 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	61.67	54	3.260	77.78	124	7.452	75.00	53	3.089
0.05	68.33	58	3.904	72.78	59	3.499	64.44	55	3.235
0.06	80.00	172	9.872	45.00	38	2.168	77.22	59	3.376
0.07	78.89	79	4.665	66.11	72	4.672	56.67	44	2.699

Table B.91 *Result of Correlation +Neural Network with initialisasi variation 275 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	85.00	224	15.616	76.11	83	5.900	80.00	131	9.032
0.05	75.56	103	7.176	59.44	35	2.419	83.33	119	8.310
0.06	76.11	90	6.202	65.56	66	6.380	70.00	69	4.888
0.07	53.33	51	3.861	77.78	215	18.624	69.44	65	4.602

Table B.92 *Result of Correlation +Neural Network with initialisasi variation 300 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	78.89	223	16.395	76.67	103	7.804	57.22	42	3.042
0.05	75.56	102	7.776	81.67	149	10.790	80.00	279	31.761
0.06	66.11	50	3.628	81.11	226	16.262	84.44	425	30.527
0.07	62.22	48	3.547	63.89	58	4.867	83.33	188	14.180

Table B.93 *Result of Correlation +Neural Network with initialisasi variation 325 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	83.33	190	16.801	78.89	176	14.500	83.33	171	13.416
0.05	76.67	118	10.032	77.22	64	5.588	77.22	68	14.610
0.06	80.00	114	8.887	83.33	110	8.736	77.22	70	5.437
0.07	83.89	278	21.688	63.33	49	3.791	71.11	52	4.305

Table B.94 *Result of Correlation +Neural Network with initialisasi variation 350 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	81.67	341	28.454	74.44	86	7.640	83.89	469	38.423
0.05	80.00	180	15.659	74.44	95	8.296	76.67	61	11.942
0.06	76.67	103	8.667	76.11	98	10.857	70.00	67	5.748
0.07	72.78	111	9.261	76.11	101	8.346	76.67	121	10.499

Table B.95 *Result of Correlation +Neural Network with initialisasi variation 375 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	80.56	137	12.090	82.22	563	56.529	80.56	163	14.352
0.05	82.22	175	22.750	82.78	170	15.129	75.56	85	9.469
0.06	63.33	50	4.474	82.22	185	19.515	60.56	50	4.416
0.07	72.22	88	7.866	71.11	83	7.301	70.00	87	8.252

Table B.96 *Result of Correlation +Neural Network with initialisasi variation 400 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.78	141	13.229	79.44	134	14.846	77.79	111	10.389
0.05	73.89	89	9.522	76.67	130	16.532	40.56	33	3.301
0.06	71.67	68	6.427	67.78	61	6.793	76.67	128	11.955
0.07	63.89	79	7.505	75.56	120	11.185	46.67	27	2.730

Table B.97 *Result of Correlation +Neural Network with initialisasi variation 425 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	62.22	48	4.946	76.11	79	8.238	60.56	40	3.994
0.05	79.44	166	16.770	57.78	41	4.083	75.00	87	8.869
0.06	78.33	152	15.210	76.67	148	16.006	79.44	117	11.569
0.07	82.22	395	39.572	76.11	153	16.287	79.44	151	15.772

Table B.98 *Result of Correlation +Neural Network with initialisasi variation 450 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.22	172	17.987	80.00	781	90.236	78.89	138	14.336
0.05	79.44	183	19.953	80.56	151	22.820	70.00	53	5.622
0.06	73.33	146	15.300	68.33	60	7.710	75.00	94	9.837
0.07	81.11	184	19.467	76.67	138	15.038	69.44	94	10.311

Table B.99 *Result of Correlation +Neural Network with initialisasi variation 475 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	68.33	61	6.708	78.89	132	15.188	73.89	49	5.397
0.05	77.22	150	16.958	78.89	191	21.382	71.67	64	7.498
0.06	78.89	146	16.459	78.89	181	23.567	69.44	93	10.284
0.07	72.22	80	9.049	43.89	42	4.820	83.89	157	18.236

Table B.100 *Result of Correlation +Neural Network with initialisasi variation 500 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	76.11	96	11.014	76.67	114	13.697	81.67	219	25.272
0.05	77.22	170	20.109	75.00	86	9.902	71.67	64	7.498
0.06	75.56	164	19.040	63.89	79	11.196	81.11	322	36.882
0.07	75.56	159	18.408	56.11	49	5.866	75.00	111	13.557

SEVEN (7) JOINT POSITIONS - RESULT OF CONSISTENCY + NEURAL NETWORK

Table B.101 *Result of Consistency + Neural Network with initialisasi variation 25 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	57.78	118	1.981	53.33	107	1.174	52.78	109	1.061
0.05	57.22	116	1.451	64.44	109	0.967	57.78	120	1.186
0.06	58.33	134	1.585	65.56	108	0.998	58.89	106	1.069
0.07	55.00	101	0.992	51.11	99	0.993	61.67	106	1.092

Table B.102 *Result of Consistency + Neural Network with initialisasi variation 50 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	57.22	105	1.323	61.11	124	1.509	63.89	107	1.170
0.05	60.56	110	1.285	57.22	100	1.092	70.00	160	2.341
0.06	66.11	93	1.031	60.00	98	1.267	80.56	244	3.477
0.07	73.89	170	1.885	75.00	199	2.144	67.78	92	1.061

Table B.103 *Result of Consistency + Neural Network with initialisasi variation 75 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	63.33	137	1.638	73.33	172	2.351	75.56	204	2.496
0.05	66.11	101	1.249	66.11	132	1.783	72.22	144	1.728
0.06	73.89	116	1.456	61.11	102	1.406	83.33	130	3.024
0.07	81.11	253	3.093	81.11	287	3.464	73.33	153	1.966

Table B.104 *Result of Consistency + Neural Network with initialisasi variation 100 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	76.11	191	2.605	77.78	169	2.430	75.00	232	3.120
0.05	83.89	264	3.878	76.11	180	2.455	85.00	348	4.797
0.06	78.33	211	2.888	76.11	170	3.397	77.78	224	5.905
0.07	76.11	157	2.125	85.56	370	5.299	81.11	246	3.214

Table B.105 *Result of Consistency + Neural Network with initialisasi variation 125 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	83.33	361	5.132	67.22	128	2.049	82.22	208	3.073
0.05	75.00	203	3.183	82.78	298	4.551	75.56	181	2.673
0.06	76.11	175	2.578	82.78	267	6.665	79.44	199	2.969
0.07	89.44	1477	21.769	78.33	157	2.536	77.78	215	3.307

Table B.106 *Result of Consistency + Neural Network with initialisasi variation 150 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	70.00	133	2.122	77.78	200	3.525	86.67	429	6.864
0.05	73.89	173	2.880	83.33	456	7.656	77.22	187	3.348
0.06	83.89	201	3.222	83.33	381	6.162	83.89	467	8.676
0.07	87.78	578	9.426	82.22	340	5.827	83.33	1892	31.699

Table B.107 *Result of Consistency + Neural Network with initialisasi variation 175 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	81.67	295	5.054	78.33	292	6.692	86.11	577	9.890
0.05	78.33	251	4.711	83.33	376	11.234	68.89	134	3.096
0.06	80.56	327	5.683	84.44	403	8.997	82.78	259	5.085
0.07	76.67	276	4.841	83.33	452	9.298	82.78	127	2.325

Table B.108 *Result of Consistency + Neural Network with initialisasi variation 200 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	87.78	559	10.405	83.33	230	4.910	82.22	242	4.571
0.05	87.78	368	7.114	82.78	357	11.353	74.44	172	3.249
0.06	85.00	1214	22.573	90.56	1403	5.666	81.11	490	8.971
0.07	79.44	162	3.104	88.33	821	15.357	72.78	90	1.794

Table B.109 *Result of Consistency + Neural Network with initialisasi variation 225 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	83.89	954	18.471	83.33	359	7.992	69.44	120	2.418
0.05	83.33	356	7.394	80.00	709	14.229	83.33	347	6.750
0.06	87.22	636	12.692	80.00	576	12.703	87.22	446	8.815
0.07	83.89	249	5.128	85.56	289	7.228	86.67	381	8.128

Table B.110 *Result of Consistency + Neural Network with initialisasi variation 250 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.22	338	7.410	87.22	2435	54.479	80.56	2239	5.086
0.05	87.22	2915	74.032	72.22	125	2.605	81.67	257	5.356
0.06	87.78	300	6.448	83.33	387	8.003	82.77	402	8.596
0.07	87.22	828	17.976	77.22	110	2.492	85.00	514	11.466

Table B.111 *Result of Consistency + Neural Network with initialisasi variation 275 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	86.11	296	7.503	88.33	595	14.272	86.67	480	11.091
0.05	87.78	393	9.786	86.11	345	7.866	82.22	348	7.943
0.06	83.89	462	10.677	86.67	519	12.724	83.89	684	15.535
0.07	83.89	381	11.134	79.44	239	7.088	82.22	306	7.456

Table B.112 *Result of Consistency + Neural Network with initialisasi variation 300 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	80.56	324	7.846	65.56	92	2.347	80.56	223	5.336
0.05	76.11	157	4.157	81.67	310	8.235	85.00	928	26.082
0.06	86.67	338	8.206	82.22	588	14.756	76.11	97	2.370
0.07	83.89	419	10.292	78.33	153	4.056	85.00	163	4.150

Table B.113 *Result of Consistency + Neural Network with initialisasi variation 325 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	83.33	190	16.801	68.89	90	2.320	78.89	267	6.650
0.05	82.22	155	4.244	76.67	99	2.617	83.33	485	13.978
0.06	81.67	123	3.070	83.33	333	10.623	81.11	329	8.175
0.07	80.56	348	8.701	82.22	171	4.181	80.00	610	16.240

Table B.114 *Result of Consistency + Neural Network with initialisasi variation 350 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	78.89	449	11.450	86.11	237	7.522	82.22	350	8.658
0.05	82.78	111	2.973	73.33	130	3.395	82.22	239	6.549
0.06	82.78	1475	37.983	78.89	175	6.434	85.00	814	20.723
0.07	85.56	788	20.449	80.56	211	5.460	82.78	231	6.256

Table B.115 *Result of Consistency + Neural Network with initialisasi variation 375 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	83.33	691	18.767	78.89	190	5.745	82.78	2410	64.646
0.05	87.22	830	27.899	77.22	119	3.231	81.67	270	7.613
0.06	83.89	2621	71.532	82.22	222	9.001	79.44	153	4.032
0.07	84.44	234	6.458	79.44	108	2.917	85.56	116	3.572

Table B.116 *Result of Consistency + Neural Network with initialisasi variation 400 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	78.33	121	3.510	72.78	121	6.373	84.44	5000	144.332
0.05	73.89	98	3.200	82.22	639	18.767	82.78	233	7.057
0.06	78.89	175	5.120	83.33	1671	55.461	84.44	923	26.627
0.07	83.33	153	4.473	78.33	110	3.198	85.00	142	4.430

Table B.117 *Result of Consistency + Neural Network with initialisasi variation 425 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	86.66	953	28.922	83.33	3844	141.645	85.56	778	23.494
0.05	83.33	857	27.643	86.11	325	10.003	83.89	364	11.398
0.06	77.78	99	3.047	55.00	43	4.290	81.67	230	6.994
0.07	82.78	335	10.292	82.22	151	4.961	86.67	889	28.813

Table B.118 *Result of Consistency + Neural Network with initialisasi variation 450 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	86.11	314	9.704	73.89	109	3.544	83.33	569	17.675
0.05	81.67	122	3.853	83.33	244	9.273	76.67	133	4.297
0.06	84.44	450	14.253	86.11	210	8.944	76.67	160	4.903
0.07	80.00	240	8.119	80.00	203	6.786	81.11	268	9.017

Table B.119 *Result of Consistency + Neural Network with initialisasi variation 475 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.78	241	7.690	83.33	295	10.241	81.11	279	9.079
0.05	83.89	388	12.901	83.33	474	15.666	81.11	356	11.888
0.06	81.67	175	5.892	83.33	183	6.395	85.56	418	13.550
0.07	83.89	240	8.192	81.11	253	8.704	82.22	727	25.288

Table B.120 *Result of Consistency + Neural Network with initialisasi variation 500 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	85.56	485	16.551	77.78	130	4.632	86.11	351	11.809
0.05	86.67	192	6.670	85.56	913	32.148	83.33	575	19.783
0.06	81.67	194	6.827	85.56	225	9.536	82.22	1667	56.760
0.07	81.11	377	12.652	81.67	93	3.338	81.67	147	5.320

EIGHT (8) JOINT POSITIONS - RESULT OF NEURAL NETWORK

Table B.121 *Result of Neural Network with initialisasi variation 25 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	74.44	106	1.513	76.11	109	1.817	76.67	98	1.482
0.05	78.33	144	2.656	74.44	94	1.404	76.67	133	3.447
0.06	76.67	83	1.305	74.44	73	1.081	76.67	66	1.018
0.07	79.44	151	2.302	76.67	169	4.517	84.44	509	8.112

Table B.122 *Result of Neural Network with initialisasi variation 50 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	78.89	71	1.342	80.56	84	1.701	85.00	183	3.292
0.05	81.67	71	1.354	84.44	144	2.683	80.00	119	2.174
0.06	81.11	68	1.322	83.33	223	4.242	73.89	85	1.639
0.07	72.78	60	1.130	60.56	49	0.970	81.11	135	2.761

Table B.123 *Result of Neural Network with initialisasi variation 75 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	78.33	71	1.684	75.81	177	4.277	74.44	91	2.590
0.05	80.00	169	4.007	68.33	63	1.446	77.78	65	1.507
0.06	81.67	70	1.636	73.33	95	2.512	79.44	131	3.037
0.07	81.11	83	1.925	68.33	45	1.023	83.33	259	6.177

Table B.124 *Result of Neural Network with initialisasi variation 100 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.78	90	2.823	57.78	50	1.408	85.00	98	2.590
0.05	82.22	329	8.606	75.56	61	1.868	73.33	45	1.760
0.06	82.22	329	8.606	85.00	144	5.221	75.00	52	1.563
0.07	82.22	170	4.519	85.56	252	9.174	75.00	66	1.841

Table B.125 *Result of Neural Network with initialisasi variation 125 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.78	90	2.823	79.44	76	2.586	82.22	89	2.776
0.05	80.56	81	2.590	81.11	162	5.040	63.33	41	1.306
0.06	82.78	302	9.685	78.89	104	3.417	75.56	78	2.450
0.07	63.89	48	1.565	78.33	55	1.964	80.00	60	2.028

Table B.126 *Result of Neural Network with initialisasi variation 150 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	81.11	73	2.589	81.67	795	31.955	60.00	50	1.825
0.05	81.67	169	6.646	70.00	40	2.950	77.78	93	3.639
0.06	83.33	232	8.278	82.22	148	5.554	83.33	60	2.133
0.07	82.20	335	12.516	80.56	197	7.039	78.33	79	3.026

Table B.127 *Result of Neural Network with initialisasi variation 175 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.78	133	5.475	82.22	131	5.581	79.44	66	2.745
0.05	82.78	65	2.792	81.11	142	7.460	80.00	109	4.739
0.06	80.00	63	2.527	90.56	238	11.034	79.44	89	3.493
0.07	70.56	47	1.922	76.67	64	3.194	80.56	256	10.670

Table B.128 *Result of Neural Network with initialisasi variation 200 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	77.78	565	25.382	81.67	153	7.448	78.89	83	3.775
0.05	81.67	125	5.835	77.78	90	4.320	66.67	48	2.150
0.06	77.78	79	3.510	82.22	175	9.794	52.78	36	1.608
0.07	84.44	242	11.017	75.00	81	5.490	62.78	51	2.699

Table B.129 *Result of Neural Network with initialisasi variation 225 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	77.78	565	25.382	81.11	131	7.408	81.11	140	7.275
0.05	73.33	50	2.527	78.89	162	8.204	81.11	199	9.963
0.06	78.33	77	3.873	78.89	101	6.972	76.11	67	3.320
0.07	50.00	31	1.600	77.78	73	4.593	62.78	51	2.699

Table B.130 *Result of Neural Network with initialisasi variation 250 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	76.67	104	7.176	70.00	47	3.423	85.00	217	14.398
0.05	77.22	89	6.195	53.89	32	2.156	82.78	517	33.936
0.06	66.11	56	3.754	77.78	125	8.907	72.78	60	3.919
0.07	71.11	71	4.815	77.22	76	5.584	84.44	208	82.220

Table B.131 *Result of Neural Network with initialisasi variation 275 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	77.22	102	7.488	56.11	34	2.508	80.00	111	7.862
0.05	54.44	35	2.851	66.67	54	3.837	60.56	48	3.423
0.06	70.00	52	3.753	65.00	64	4.577	81.11	173	12.005
0.07	82.22	189	13.903	67.22	49	3.616	62.22	59	4.414

Table B.132 *Result of Neural Network with initialisasi variation 300 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	70.56	57	7.363	77.78	77	6.055	81.67	193	14.539
0.05	74.44	89	9.293	82.22	142	10.802	79.44	92	11.920
0.06	82.22	237	17.926	77.78	143	11.029	77.78	174	13.198
0.07	76.11	116	8.790	52.78	39	3.370	70.56	59	4.696

Table B.133 *Result of Neural Network with initialisasi variation 325 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	81.67	131	10.842	73.89	62	5.372	65.56	43	3.525
0.05	79.44	248	22.188	72.22	64	5.195	82.22	143	28.088
0.06	79.44	166	13.540	70.56	72	6.115	65.56	46	3.935
0.07	71.11	104	8.623	70.00	44	3.604	81.11	129	10.998

Table B.134 *Result of Neural Network with initialisasi variation 350 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.22	148	12.777	78.33	100	9.172	70.00	56	4.836
0.05	67.22	54	5.195	59.44	45	4.024	77.78	128	12.291
0.06	80.56	149	12.927	62.78	45	4.212	81.11	147	12.821
0.07	75.00	105	9.177	78.89	149	12.823	74.44	104	9.547

Table B.135 *Result of Neural Network with initialisasi variation 375 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	78.89	114	10.561	76.67	74	8.276	78.89	220	20.390
0.05	77.22	78	7.564	79.44	154	14.577	64.44	47	4.610
0.06	76.11	203	19.029	67.78	48	4.633	80.00	124	11.535
0.07	76.67	158	14.855	71.67	98	9.110	51.67	41	4.025

Table B.136 *Result of Neural Network with initialisasi variation 400 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	73.89	66	6.490	55.00	37	4.182	76.11	96	9.423
0.05	68.33	44	4.462	80.56	234	23.092	67.22	56	5.752
0.06	77.22	166	16.282	38.89	27	2.764	54.44	36	3.587
0.07	75.56	113	11.181	80.00	270	26.427	65.00	58	5.990

Table B.137 *Result of Neural Network with initialisasi variation 425 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	68.89	55	6.053	72.78	55	6.019	79.44	123	12.698
0.05	80.00	135	14.711	72.22	70	8.437	81.67	190	19.957
0.06	78.89	101	10.637	75.00	93	9.828	76.67	211	21.959
0.07	58.33	51	5.386	60.00	50	5.398	66.11	76	8.280

Table B.138 *Result of Neural Network with initialisasi variation 450 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	71.11	64	6.957	80.00	130	16.336	77.78	94	10.234
0.05	71.11	63	9.814	74.44	67	7.859	78.89	93	10.331
0.06	52.78	51	5.724	67.78	85	9.875	74.44	90	9.895
0.07	80.56	182	22.605	71.67	124	13.947	79.44	133	15.241

Table B.139 *Result of Neural Network with initialisasi variation 475 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	63.89	61	7.051	69.44	47	5.701	68.33	52	6.052
0.05	57.78	41	4.852	67.22	79	9.936	76.67	137	16.646
0.06	72.22	165	19.233	78.33	199	26.576	78.33	163	19.034
0.07	76.11	190	22.272	75.56	119	14.367	65.00	69	8.424

Table B.140 *Result of Neural Network with initialisasi variation 500 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	71.67	80	9.641	52.78	36	4.945	77.78	90	10.889
0.05	73.33	93	11.575	50.00	32	4.206	77.22	144	17.553
0.06	70.00	108	13.177	67.79	82	10.654	77.78	113	13.664
0.07	78.89	167	20.242	73.89	106	13.198	52.22	56	7.130

EIGHT (8) JOINT POSITIONS - RESULT OF CORRELATION + NEURAL NETWORK

Table B.141 *Result of Correlation + Neural Network initialisasi variation 25 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	81.67	180	2.636	75.00	74	1.238	73.89	171	2.559
0.05	70.00	65	1.086	70.00	73	1.155	83.33	295	5.600
0.06	81.67	165	2.562	71.11	66	1.030	78.33	94	1.598
0.07	82.22	107	1.609	57.22	54	1.298	52.22	52	0.842

Table B.142 *Result of Correlation + Neural Network initialisasi variation 50 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	78.33	72	1.279	83.89	115	2.352	78.89	82	1.544
0.05	70.00	60	1.380	80.00	70	1.357	76.11	107	1.988
0.06	84.44	221	4.333	80.00	68	1.244	81.67	189	3.828
0.07	76.67	103	1.927	77.22	150	2.759	82.22	95	1.903

Table B.143 *Result of Correlation + Neural Network initialisasi variation 75 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	83.89	332	8.050	75.84	215	5.591	81.67	73	1.685
0.05	85.56	329	8.992	82.22	225	5.400	82.22	118	3.212
0.06	43.89	26	0.632	82.22	134	3.810	78.89	95	2.369
0.07	74.44	65	1.509	78.33	61	1.678	71.67	111	2.668

Table B.144 *Result of Correlation + Neural Network initialisasi variation 100 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.78	70	1.919	79.44	205	5.890	80.00	118	3.166
0.05	78.33	168	4.819	80.56	67	1.804	81.11	83	2.597
0.06	68.00	38	0.996	79.44	66	3.109	83.33	300	9.476
0.07	81.11	161	4.236	81.67	129	3.851	75.56	58	1.623

Table B.145 *Result of Correlation + Neural Network initialisasi variation 125 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	76.67	77	2.449	80.00	126	4.232	80.56	218	6.880
0.05	65.00	36	1.157	73.33	56	1.974	80.00	49	1.500
0.06	81.11	176	2.645	58.33	42	1.466	82.78	169	5.257
0.07	77.78	69	2.193	80.56	330	5.153	80.56	178	5.975

Table B.146 *Result of Correlation + Neural Network initialisasi variation 150 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.78	107	3.853	81.11	166	6.620	73.33	83	2.948
0.05	80.56	100	3.853	81.67	119	5.919	78.33	100	4.183
0.06	91.56	98	2.215	79.44	102	3.791	76.67	86	3.057
0.07	75.00	48	1.782	80.56	239	8.972	65.00	40	1.545

Table B.147 *Result of Correlation + Neural Network initialisasi variation 175 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	63.33	53	2.402	72.22	42	1.913	80.00	83	3.448
0.05	83.33	143	6.053	79.44	121	8.050	78.33	107	5.073
0.06	83.33	226	9.081	80.00	100	5.618	78.89	57	2.262
0.07	83.33	142	5.828	82.78	104	4.989	79.44	120	5.038

Table B.148 *Result of Correlation + Neural Network initialisasi variation 200 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	81.11	198	8.704	83.33	346	17.551	80.00	116	5.226
0.05	83.89	142	7.360	75.00	75	3.377	81.11	304	13.567
0.06	79.44	54	2.434	79.44	118	5.939	62.22	38	1.691
0.07	80.56	122	5.542	78.33	89	5.058	83.89	102	4.726

Table B.149 *Result of Correlation + Neural Network initialisasi variation 225 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	80.00	268	13.369	79.44	119	6.763	81.67	69	3.557
0.05	77.22	144	7.192	78.89	162	8.204	80.56	172	8.434
0.06	81.67	230	11.117	77.78	134	7.640	85.00	238	11.350
0.07	75.00	71	3.599	81.11	61	3.385	76.67	63	3.245

Table B.150 *Result of Correlation + Neural Network initialisasi variation 250 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	81.11	87	6.364	79.44	114	7.857	82.22	86	5.694
0.05	82.22	149	10.197	81.67	78	5.219	76.67	106	6.959
0.06	82.22	71	4.706	74.44	48	3.261	81.67	260	16.937
0.07	84.44	208	13.947	82.22	190	14.171	70.56	70	4.820

Table B.151 *Result of Correlation + Neural Network initialisasi variation 275 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	80.56	67	4.773	73.89	92	7.246	80.00	122	8.487
0.05	36.11	32	2.427	82.22	91	6.423	61.67	36	2.588
0.06	67.22	48	3.426	67.78	43	3.072	81.67	106	7.464
0.07	82.22	172	12.817	69.44	52	4.058	79.44	119	8.954

Table B.152 *Result of Correlation + Neural Network initialisasi variation 300 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	80.56	625	51.875	75.56	58	4.590	80.56	119	8.970
0.05	79.44	127	10.240	80.00	244	19.386	52.78	31	5.112
0.06	61.67	56	4.237	68.89	68	5.413	76.11	121	9.070
0.07	81.67	543	40.860	72.78	76	5.834	72.22	90	7.129

Table B.153 *Result of Correlation + Neural Network initialisasi variation 325 node hidden layer*

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	74.44	77	7.862	84.44	203	21.743	78.89	132	10.702
0.05	77.78	73	6.546	66.11	45	3.700	79.44	85	8.655
0.06	74.44	94	7.653	72.78	78	6.381	78.89	119	9.937
0.07	73.89	113	9.419	65.00	57	4.649	75.56	149	12.543

Table B.154 ***Result of Correlation + Neural Network initialisasi variation 350 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	80.56	130	11.419	78.33	73	6.670	80.00	132	11.326
0.05	69.44	54	4.820	80.00	183	15.835	82.22	470	45.612
0.06	76.11	113	9.797	79.44	130	11.216	76.11	72	6.301
0.07	60.00	61	5.310	65.00	48	4.150	67.22	56	5.148

Table B.155 ***Result of Correlation + Neural Network initialisasi variation 375 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	36.67	20	1.903	76.11	70	7.039	82.78	188	17.488
0.05	67.78	53	5.330	76.11	108	10.184	71.67	59	5.750
0.06	79.44	120	11.244	54.44	47	4.867	76.11	86	8.015
0.07	67.78	66	6.201	76.67	198	18.985	76.11	166	16.614

Table B.156 ***Result of Correlation + Neural Network initialisasi variation 400 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	77.78	84	8.314	76.11	89	9.980	80.56	154	14.992
0.05	65.00	53	5.421	80.56	105	10.417	80.00	108	11.200
0.06	50.56	34	3.392	72.22	70	7.697	71.11	73	7.161
0.07	69.44	64	6.358	76.67	112	11.450	73.33	101	10.390

Table B.157 ***Result of Correlation + Neural Network initialisasi variation 425 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	81.11	171	17.769	78.89	116	13.135	82.22	118	12.199
0.05	80.56	140	14.804	80.56	183	21.084	64.44	49	5.160
0.06	45.56	38	3.998	75.56	76	7.956	83.89	215	22.848
0.07	53.89	42	4.446	77.78	90	9.625	62.78	55	6.021

Table B.158 ***Result of Correlation + Neural Network initialisasi variation 450 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	77.78	565	25.382	67.78	39	5.810	71.11	60	6.552
0.05	81.11	264	39.560	75.56	92	10.443	65.00	64	7.122
0.06	71.11	66	7.601	77.78	93	10.660	80.56	178	19.469
0.07	70.56	58	7.117	75.56	138	15.522	76.11	90	10.437

Table B.159 ***Result of Correlation + Neural Network initialisasi variation 475 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	75.56	117	13.557	75.56	117	14.070	75.56	101	11.700
0.05	77.22	124	14.602	67.22	65	7.571	78.33	158	19.401
0.06	76.11	97	11.314	65.56	53	7.262	69.44	58	6.754
0.07	66.11	77	8.987	81.11	151	18.033	79.44	140	17.254

Table B.160 ***Result of Correlation + Neural Network initialisasi variation 500 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	78.89	80	9.657	68.89	73	9.523	81.67	283	34.117
0.05	67.22	61	7.535	71.67	68	8.400	75.00	70	8.565
0.06	83.33	466	56.722	64.44	59	9.436	56.11	41	4.976
0.07	53.89	42	5.135	80.56	296	37.050	78.89	172	21.778

EIGHT (8) JOINT POSITIONS - RESULT OF CONSISTENCY + NEURAL NETWORK

Table B.161 ***Result of Consistency + Neural Network with initialisasi variation 25 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	67.22	110	1.045	59.44	112	1.245	65.56	113	1.124
0.05	58.33	109	1.137	68.89	107	1.108	60.56	107	1.123
0.06	52.78	104	1.052	68.89	104	1.096	56.67	105	1.062
0.07	67.22	102	1.050	75.56	101	1.403	65.00	98	1.014

Table B.162 ***Result of Consistency + Neural Network with initialisasi variation 50 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	69.44	138	1.482	83.89	115	2.163	71.11	102	1.123
0.05	81.11	151	3.712	76.11	185	2.059	81.67	159	1.828
0.06	83.89	203	2.238	69.44	118	1.318	73.89	104	1.345
0.07	86.67	243	2.773	81.67	286	3.134	76.11	133	1.607

Table B.163 ***Result of Consistency + Neural Network with initialisasi variation 75 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	76.67	167	2.277	80.00	191	2.596	82.22	214	2.698
0.05	67.22	94	1.393	78.33	166	2.116	80.00	210	2.713
0.06	70.00	102	1.478	80.56	199	2.699	84.44	363	4.704
0.07	83.33	190	2.408	82.78	185	2.280	81.11	246	3.214

Table B.164 ***Result of Consistency + Neural Network with initialisasi variation 100 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	81.67	452	6.520	81.11	361	5.344	78.33	209	2.871
0.05	81.11	304	4.447	82.22	500	7.240	86.11	436	5.883
0.06	84.44	211	2.920	81.11	388	7.691	80.56	339	5.935
0.07	82.78	376	5.221	83.33	212	3.110	81.67	237	3.416

Table B.165 ***Result of Consistency + Neural Network with initialisasi variation 125 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	83.33	306	4.508	82.78	213	3.433	71.11	102	1.544
0.05	83.33	215	3.636	86.11	389	5.572	76.67	119	1.799
0.06	81.11	176	2.645	86.11	387	6.180	75.00	94	1.418
0.07	83.89	119	1.839	80.56	330	5.153	88.89	491	7.706

Table B.166 ***Result of Consistency + Neural Network with initialisasi variation 150 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	86.11	415	6.708	81.11	588	10.681	84.44	370	6.256
0.05	81.11	557	10.047	85.56	308	5.476	85.00	2753	55.442
0.06	84.44	1184	19.678	85.56	252	4.181	83.33	424	7.188
0.07	81.11	143	2.416	86.11	273	4.461	81.67	137	2.371

Table B.167 ***Result of Consistency + Neural Network with initialisasi variation 175 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.78	414	7.441	80.56	269	5.529	82.78	355	6.350
0.05	81.67	542	10.983	82.78	374	10.464	86.11	1758	40.151
0.06	86.11	422	7.546	82.78	357	9.577	86.11	662	11.618
0.07	81.11	160	2.918	86.11	1044	20.606	80.56	143	2.699

Table B.168 ***Result of Consistency + Neural Network with initialisasi variation 200 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.22	385	7.145	83.89	189	4.110	77.22	213	4.119
0.05	85.00	299	6.209	81.67	323	6.435	84.44	355	6.783
0.06	81.11	184	3.575	81.67	255	7.744	80.00	124	2.362
0.07	80.56	155	3.072	87.78	807	16.274	81.11	154	3.136

Table B.169 ***Result of Consistency + Neural Network with initialisasi variation 225 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	86.67	295	5.928	72.78	112	2.523	81.67	351	7.254
0.05	81.67	227	4.898	87.22	283	5.755	78.33	149	3.061
0.06	81.67	341	7.074	86.11	427	8.260	87.78	295	7.080
0.07	83.33	213	5.037	85.56	346	7.942	79.44	110	2.402

Table B.170 ***Result of Consistency + Neural Network with initialisasi variation 250 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	80.56	118	2.824	79.44	260	6.022	86.11	184	3.978
0.05	80.00	265	6.603	80.56	118	2.585	83.33	233	5.026
0.06	80.00	155	3.445	81.11	201	4.727	82.22	233	5.047
0.07	85.00	138	3.003	78.33	598	14.928	80.56	172	3.978

Table B.171 ***Result of Consistency + Neural Network with initialisasi variation 275 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	80.00	269	6.661	68.33	92	2.303	82.78	145	3.416
0.05	81.67	157	5.094	80.00	235	5.540	86.11	2144	51.079
0.06	83.89	1222	29.040	78.33	129	3.126	80.56	233	5.484
0.07	80.56	100	2.437	69.44	52	4.058	81.11	1734	43.087

Table B.172 ***Result of Consistency + Neural Network with initialisasi variation 300 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	72.78	101	2.605	81.67	141	3.732	73.33	88	2.200
0.05	85.56	1438	39.013	78.33	297	7.407	79.44	773	47.577
0.06	85.00	189	4.941	78.33	269	6.755	82.22	494	12.263
0.07	80.00	134	3.390	80.56	339	9.676	85.00	398	10.514

Table B.173 ***Result of Consistency + Neural Network with initialisasi variation 325 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.78	1825	65.208	86.66	300	8.048	80.00	186	4.758
0.05	86.67	111	301.000	82.78	251	6.451	83.33	181	6.417
0.06	81.11	209	5.308	82.78	607	15.163	85.56	917	27.352
0.07	80.56	245	6.262	80.56	131	3.322	82.78	215	5.881

Table B.174 ***Result of Consistency + Neural Network with initialisasi variation 350 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.22	2146	56.800	83.33	1668	55.699	75.00	118	3.167
0.05	81.67	385	12.574	81.11	254	7.595	83.33	244	8.878
0.06	79.44	440	11.767	79.44	92	2.528	80.00	259	6.888
0.07	85.00	375	10.137	82.78	684	18.299	83.33	937	26.598

Table B.175 ***Result of Consistency + Neural Network with initialisasi variation 375 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.22	1353	38.064	76.67	98	3.028	82.78	416	11.403
0.05	82.22	228	6.642	80.56	431	12.379	83.89	696	20.114
0.06	79.44	427	12.371	81.11	351	10.327	78.33	222	6.232
0.07	83.33	155	4.383	85.00	172	4.742	81.67	257	7.753

Table B.176 ***Result of Consistency + Neural Network with initialisasi variation 400 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	80.56	376	11.310	78.89	268	9.780	80.00	171	5.117
0.05	80.56	793	25.456	80.00	163	5.003	84.44	303	9.915
0.06	80.00	761	23.059	80.00	184	6.817	80.56	91	2.779
0.07	84.44	99	3.027	81.11	109	3.276	84.44	157	5.054

Table B.177 ***Result of Consistency + Neural Network with initialisasi variation 425 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	81.67	364	11.528	76.67	108	3.682	85.56	792	25.569
0.05	82.22	265	8.564	81.67	395	18.992	84.44	576	18.510
0.06	82.78	268	8.545	81.11	1038	33.259	81.67	230	7.294
0.07	81.11	338	10.832	83.89	302	10.124	81.67	100	3.417

Table B.178 ***Result of Consistency + Neural Network with initialisasi variation 450 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	78.89	238	7.597	83.33	581	20.751	79.44	192	6.240
0.05	83.33	493	16.006	77.22	182	6.092	80.00	140	4.738
0.06	81.11	347	14.636	80.56	341	13.529	81.67	83	2.701
0.07	78.33	94	3.108	81.11	234	7.988	85.56	99	3.572

Table B.179 ***Result of Consistency + Neural Network with initialisasi variation 475 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	79.44	317	10.982	81.11	463	16.496	81.67	279	9.406
0.05	81.67	479	16.661	79.44	323	11.636	84.44	302	10.768
0.06	79.44	260	8.759	79.44	239	9.825	80.56	129	4.449
0.07	84.44	271	9.445	77.22	60	2.137	79.44	241	8.767

Table B.180 ***Result of Consistency + Neural Network with initialisasi variation 500 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	79.44	121	4.290	78.89	484	18.013	79.44	394	13.946
0.05	77.22	131	4.664	78.33	92	3.340	80.00	375	13.523
0.06	83.33	846	30.317	78.33	82	3.100	78.33	303	10.838
0.07	81.11	183	6.716	78.33	88	3.323	78.89	152	5.756

EIGHT (8) JOINT POSITIONS - RESULT OF CFSSUBSETEVAL + NEURAL NETWORK

Table B.181 ***Result of CfsSubsetEval Neural Network with initialisasi variation 25 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	67.22	110	1.045	59.44	112	1.245	65.56	113	1.124
0.05	58.33	109	1.137	68.89	107	1.108	60.56	107	1.123
0.06	52.78	104	1.052	68.89	104	1.096	56.67	105	1.062
0.07	67.22	102	1.050	75.56	101	1.403	68.89	101	1.017

Table B.182 ***Result of CfsSubsetEval Neural Network with initialisasi variation 50 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	79.44	138	1.482	83.89	115	2.163	71.11	102	1.123
0.05	81.11	151	3.712	76.11	185	2.059	81.67	159	1.828
0.06	83.89	203	2.238	69.44	118	1.318	73.89	104	1.345
0.07	86.67	243	2.773	81.67	286	3.134	76.11	133	1.607

Table B.183 ***Result of CfsSubsetEval Neural Network with initialisasi variation 75 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	76.67	167	2.277	80.00	191	2.596	82.22	214	2.698
0.05	67.22	94	1.393	78.33	166	2.116	80.00	210	2.713
0.06	70.00	102	1.478	96.56	499	3.699	84.44	363	4.704
0.07	83.33	190	2.408	82.78	185	2.280	81.11	246	3.214

Table B.184 ***Result of CfsSubsetEval Neural Network with initialisasi variation 100 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	81.67	452	6.520	81.11	361	5.344	78.33	209	2.871
0.05	81.11	304	4.447	82.22	500	7.240	86.11	436	5.883
0.06	84.44	211	2.920	81.11	388	7.691	80.56	339	5.935
0.07	82.78	376	5.221	83.33	212	3.110	81.67	237	3.416

Table B.185 ***Result of CfsSubsetEval Neural Network with initialisasi variation 125 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	83.33	346	4.508	82.78	213	3.433	71.11	102	1.544
0.05	83.33	215	3.636	86.11	389	5.572	76.67	119	1.799
0.06	81.11	176	2.645	86.11	387	6.180	75.00	94	1.418
0.07	83.89	119	1.839	80.56	330	5.153	88.89	591	5.506

Table B.186 ***Result of CfsSubsetEval Neural Network with initialisasi variation 150 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	86.11	415	6.708	81.11	588	10.681	84.44	370	6.256
0.05	81.11	557	10.047	85.56	308	5.476	85.00	2753	55.442
0.06	84.44	1184	19.678	85.56	252	4.181	83.33	428	7.188
0.07	81.11	143	2.416	86.89	673	4.571	81.67	137	2.371

Table B.187 ***Result of CfsSubsetEval Neural Network with initialisasi variation 175 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.78	414	7.441	80.56	269	5.529	82.78	355	6.350
0.05	81.67	542	10.983	82.78	374	10.464	86.11	1758	40.151
0.06	86.67	742	8.546	82.78	357	9.577	86.11	662	11.618
0.07	81.11	160	2.918	86.11	1044	20.606	80.56	143	2.699

Table B.188 ***Result of CfsSubsetEval Neural Network with initialisasi variation 200 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.22	385	7.145	83.89	189	4.110	77.22	213	4.119
0.05	85.00	299	6.209	81.67	323	6.435	84.44	355	6.783
0.06	81.11	184	3.575	81.67	255	7.744	80.00	124	2.362
0.07	80.56	155	3.072	87.89	906	12.275	81.11	154	3.136

Table B.189 ***Result of CfsSubsetEval Neural Network with initialisasi variation 225 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	86.67	295	5.928	72.78	112	2.523	81.67	351	7.254
0.05	81.67	227	4.898	87.22	283	5.755	78.33	149	3.061
0.06	81.67	341	7.074	87.78	495	7.380	86.11	427	8.260
0.07	83.33	213	5.037	85.56	346	7.942	79.44	110	2.402

Table B.190 ***Result of CfsSubsetEval Neural Network with initialisasi variation 250 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	80.56	118	2.824	79.44	260	6.022	86.11	184	4.978
0.05	80.00	265	6.603	80.56	118	2.585	83.33	233	5.026
0.06	80.00	155	3.445	81.11	201	4.727	82.22	233	5.047
0.07	85.00	138	3.003	78.33	598	14.928	80.56	172	3.978

Table B.191 ***Result of CfsSubsetEval Neural Network with initialisasi variation 275 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	80.00	269	6.661	68.33	92	2.303	82.78	145	3.416
0.05	81.67	157	5.094	80.00	235	5.540	86.11	2144	51.079
0.06	83.89	1222	29.040	78.33	129	3.126	80.56	233	5.484
0.07	80.56	130	2.337	69.44	52	4.058	81.11	1734	43.087

Table B.192 ***Result of CfsSubsetEval Neural Network with initialisasi variation 300 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	72.78	101	2.605	81.67	141	3.732	73.33	88	2.200
0.05	85.56	1438	39.013	78.33	295	7.405	79.44	773	47.577
0.06	85.00	189	4.941	78.33	269	6.755	82.22	494	12.263
0.07	80.00	134	3.390	80.56	339	9.676	85.00	398	10.514

Table B.193 ***Result of CfsSubsetEval Neural Network with initialisasi variation 325 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.78	1825	65.208	86.66	300	8.048	80.00	186	4.758
0.05	86.67	151	3.450	82.78	251	6.451	83.33	184	4.417
0.06	81.11	209	5.308	82.78	607	15.163	85.56	917	27.352
0.07	80.56	245	6.262	80.56	131	3.322	82.78	215	5.881

Table B.194 ***Result of + CfsSubsetEval   Neural Network with initialisasi variation 350 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.22	2146	56.800	83.33	1668	55.699	75.00	118	3.167
0.05	81.67	385	12.574	81.11	254	7.595	83.33	244	8.878
0.06	79.44	440	11.767	79.44	92	2.528	80.00	259	6.888
0.07	85.00	355	9.137	82.78	684	18.299	83.33	937	26.598

Table B.195 ***Result of CfsSubsetEval Neural Network with initialisasi variation 375 node hidden layer***

LEARNING RATE (LR)	MOMENTUM CONT								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.22	1353	38.064	76.67	98	3.028	82.78	416	11.403
0.05	82.22	228	6.642	80.56	431	12.379	83.89	696	20.114
0.06	79.44	427	12.371	81.11	351	10.327	78.33	222	6.232
0.07	83.33	155	4.383	85.00	176	4.742	81.67	257	7.753

Table B.196 ***Result of CfsSubsetEval Neural Network with initialisasi variation 400 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	80.56	376	11.310	78.89	268	9.780	80.00	171	5.117
0.05	80.56	793	25.456	80.00	163	5.003	84.44	303	9.915
0.06	80.00	761	23.059	80.00	184	6.817	80.56	91	2.779
0.07	84.44	101	2.027	81.11	109	3.276	84.44	157	5.054

Table B.197 ***Result of CfsSubsetEval Neural Network with initialisasi variation 425 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	81.67	364	11.528	76.67	108	3.682	85.56	792	25.569
0.05	82.22	265	8.564	81.67	395	18.992	84.33	574	12.520
0.06	82.78	268	8.545	81.11	1038	33.259	81.67	230	7.294
0.07	81.11	338	10.832	83.89	302	10.124	81.67	100	3.417

Table B.198 ***Result of CfsSubsetEval Neural Network with initialisasi variation 450 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	78.89	238	7.597	83.33	581	20.751	79.44	192	6.240
0.05	83.33	493	16.006	77.22	182	6.092	80.00	140	4.738
0.06	81.11	347	14.636	80.56	341	13.529	81.67	83	2.701
0.07	78.44	114	3.128	81.11	234	7.988	85.56	99	3.572

Table B.199 ***Result of CfsSubsetEval Neural Network with initialisasi variation 475 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	79.44	317	10.982	81.11	463	16.496	81.67	279	9.406
0.05	81.67	479	16.661	79.44	323	11.636	84.44	302	10.768
0.06	79.44	260	8.759	79.44	239	9.825	80.56	129	4.449
0.07	84.56	276	9.465	77.22	60	2.137	79.44	241	8.767

Table B.200 ***Result of CfsSubsetEval Neural Network with initialisasi variation 500 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	79.44	121	4.290	78.89	484	18.013	79.44	394	13.946
0.05	77.22	131	4.664	78.33	92	3.340	80.00	375	13.523
0.06	83.44	825	10.317	78.33	82	3.100	78.33	303	10.838
0.07	81.11	183	6.716	78.33	88	3.323	78.89	152	5.756

SEVEN (7) JOINT POSITIONS - RESULT OF CFSSUBSETEVAL + NEURAL NETWORK

Table B.201 ***Result of CfsSubsetEval with Neural Network with initialisasi variation 25 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	57.78	118	1.981	53.33	107	1.174	52.78	109	1.061
0.05	57.22	116	1.451	64.44	109	0.967	57.78	120	1.186
0.06	58.33	134	1.585	66.56	158	0.988	58.89	106	1.069
0.07	55.00	101	0.992	51.11	99	0.993	61.67	106	1.092

Table B.202 ***Result of CfsSubsetEval with Neural Network with initialisasi variation 50 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	57.22	105	1.323	61.11	124	1.509	63.89	107	1.170
0.05	60.56	110	1.285	57.22	100	1.092	70.00	160	2.341
0.06	66.11	93	1.031	60.00	98	1.267	80.89	265	3.577
0.07	73.89	170	1.885	75.00	199	2.144	67.78	92	1.061

Table B.203 ***Result of CfsSubsetEval with Neural Network with initialisasi variation 75 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	63.33	137	1.638	73.33	172	2.351	75.56	204	2.496
0.05	66.11	101	1.249	66.11	132	1.783	72.22	144	1.728
0.06	73.89	116	1.456	61.11	102	1.406	83.33	153	2.024
0.07	81.11	253	3.093	81.11	287	3.464	73.33	153	1.966

Table B.204 ***Result of CfsSubsetEval with Neural Network with initialisasi variation 100 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	76.11	191	2.605	77.78	169	2.430	75.00	232	3.120
0.05	83.89	264	3.878	76.11	180	2.455	85.00	348	4.797
0.06	78.33	211	2.888	76.11	170	3.397	77.78	224	5.905
0.07	76.11	157	2.125	83.56	570	4.299	81.11	246	3.214

Table B.205 ***Result of CfsSubsetEval with Neural Network with initialisasi variation 125 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	83.33	361	5.132	67.22	128	2.049	82.22	208	3.073
0.05	75.00	203	3.183	82.78	298	4.551	75.56	181	2.673
0.06	76.11	175	2.578	93.67	1445	6.665	79.44	199	2.969
0.07	82.44	1277	21.769	78.33	157	2.536	77.78	215	3.307

Table B.206 ***Result of CfsSubsetEval with Neural Network with initialisasi variation 150 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	70.00	133	2.122	77.78	200	3.525	86.67	429	6.864
0.05	73.89	173	2.880	83.33	456	7.656	77.22	187	3.348
0.06	83.89	201	3.222	83.33	381	6.162	83.89	467	8.676
0.07	87.56	558	7.426	82.22	340	5.827	83.33	1892	31.699

Table B.207 ***Result of CfsSubsetEval with Neural Network with initialisasi variation 175 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	81.67	295	5.054	78.33	292	6.692	86.33	377	7.890
0.05	78.33	251	4.711	83.33	376	11.234	68.89	134	3.096
0.06	80.56	327	5.683	84.44	403	8.997	82.78	259	5.085
0.07	76.67	276	4.841	83.33	452	9.298	82.78	127	2.325

Table B.208 ***Result of CfsSubsetEval with Neural Network with initialisasi variation 200 node hidden layer***

LEARNING RATE (LR)	MOMENTUM CONT								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	87.78	559	10.405	83.33	230	4.910	82.22	242	4.571
0.05	87.78	368	7.114	82.78	357	11.353	74.44	172	3.249
0.06	85.00	1214	22.573	91.11	2403	45.666	81.11	490	8.971
0.07	79.44	162	3.104	88.33	821	15.357	72.78	90	1.794

Table B.209 ***Result of CfsSubsetEval with Neural Network with initialisasi variation 225 node hidden layer***

LEARNING RATE (LR)	MOMENTUM CONT								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	83.89	954	18.471	83.33	359	7.992	69.44	120	2.418
0.05	83.33	356	7.394	80.00	709	14.229	83.33	347	6.750
0.06	87.22	636	12.692	80.00	576	12.703	87.56	486	9.815
0.07	83.89	249	5.128	85.56	289	7.228	86.67	381	8.128

Table B.210 ***Result of CfsSubsetEval with Neural Network with initialisasi variation 250 node hidden layer***

LEARNING RATE (LR)	MOMENTUM CONT								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.22	338	7.410	87.22	2435	54.479	80.56	2239	5.086
0.05	87.22	2915	74.032	72.22	125	2.605	81.67	257	5.356
0.06	87.78	350	5.448	83.33	387	8.003	82.77	402	8.596
0.07	87.22	828	17.976	77.22	110	2.492	85.00	514	11.466

Table B.211 ***Result of CfsSubsetEval with Neural Network with initialisasi variation 275 node hidden layer***

LEARNING RATE (LR)	MOMENTUM CONT								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	86.11	296	7.503	88.33	695	13.272	86.67	480	11.091
0.05	87.78	393	9.786	86.11	345	7.866	82.22	348	7.943
0.06	83.89	462	10.677	86.67	519	12.724	83.89	684	15.535
0.07	83.89	381	11.134	79.44	239	7.088	82.22	306	7.456

Table B.212 ***Result of CfsSubsetEval with Neural Network with initialisasi variation 300 node hidden layer***

LEARNING RATE (LR)	MOMENTUM CONT								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	80.56	324	7.846	65.56	92	2.347	80.56	223	5.336
0.05	76.11	157	4.157	81.67	310	8.235	85.00	928	26.082
0.06	86.67	338	8.206	82.22	588	14.756	76.11	97	2.370
0.07	83.89	419	10.292	78.33	153	4.056	85.00	163	4.150

Table B.213 ***Result of CfsSubsetEval with Neural Network with initialisasi variation 325 node hidden layer***

LEARNING RATE (LR)	MOMENTUM CONT								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	83.33	160	14.801	68.89	90	2.320	78.89	267	6.650
0.05	82.22	155	4.244	76.67	99	2.617	83.33	485	13.978
0.06	81.67	123	3.070	83.33	333	10.623	81.11	329	8.175
0.07	80.56	348	8.701	82.22	171	4.181	80.00	610	16.240

Table B.214 ***Result of CfsSubsetEval with Neural Network with initialisasi variation 350 node hidden layer***

LEARNING RATE (LR)	MOMENTUM CONT								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	78.89	449	11.450	86.11	267	6.722	82.22	350	8.658
0.05	82.78	111	2.973	73.33	130	3.395	82.22	239	6.549
0.06	82.78	1475	37.983	78.89	175	6.434	85.00	814	20.723
0.07	85.56	788	20.449	80.56	211	5.460	82.78	231	6.256

Table B.215 ***Result of CfsSubsetEval with Neural Network with initialisasi variation 375 node hidden layer***

LEARNING RATE (LR)	MOMENTUM CONT								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	83.33	691	18.767	78.89	190	5.745	82.78	2410	64.646
0.05	87.22	830	27.899	77.22	119	3.231	81.67	270	7.613
0.06	83.89	2621	71.532	82.22	222	9.001	79.44	153	4.032
0.07	84.44	234	6.458	79.44	108	2.917	85.56	116	3.572

Table B.216 ***Result of CfsSubsetEval with Neural Network with initialisasi variation 400 node hidden layer***

LEARNING RATE (LR)	MOMENTUM CONT								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	78.33	121	3.510	72.78	121	6.373	84.44	5000	144.332
0.05	73.89	98	3.200	82.22	639	18.767	82.78	233	7.057
0.06	78.89	175	5.120	83.33	1671	55.461	84.44	923	26.627
0.07	83.33	153	4.473	78.33	110	3.198	85.56	147	3.430

Table B.217 ***Result of CfsSubsetEval with Neural Network with initialisasi variation 425 node hidden layer***

LEARNING RATE (LR)	MOMENTUM CONT								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	86.66	953	28.922	83.33	3844	141.645	85.56	778	23.494
0.05	83.33	857	27.643	86.11	325	10.003	83.89	364	11.398
0.06	77.78	99	3.047	55.00	43	4.290	81.67	230	6.994
0.07	82.78	335	10.292	82.22	151	4.961	86.67	789	27.813

Table B.218 ***Result of CfsSubsetEval with Neural Network with initialisasi variation 450 node hidden layer***

LEARNING RATE (LR)	MOMENTUM CONT								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	86.11	314	9.704	73.89	109	3.544	83.33	569	17.675
0.05	81.67	122	3.853	83.33	244	9.273	76.67	133	4.297
0.06	84.44	450	14.253	86.11	220	6.544	76.67	160	4.903
0.07	80.00	240	8.119	80.00	203	6.786	81.11	268	9.017

Table B.219 ***Result of CfsSubsetEval with Neural Network with initialisasi variation 475 node hidden layer***

LEARNING RATE (LR)	MOMENTUM CONT								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.78	241	7.690	83.33	295	10.241	81.11	279	9.079
0.05	83.89	388	12.901	83.33	474	15.666	81.11	356	11.888
0.06	81.67	175	5.892	83.33	183	6.395	85.56	418	12.550
0.07	83.89	240	8.192	81.11	253	8.704	82.22	727	25.288

Table B.220 ***Result of CfsSubsetEval with Neural Network with initialisasi variation 500 node hidden layer***

LEARNING RATE (LR)	MOMENTUM CONT								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	85.56	485	16.551	77.78	130	4.632	86.11	351	11.809
0.05	86.67	190	6.556	85.56	913	32.148	83.33	575	19.783
0.06	81.67	194	6.827	85.56	225	9.536	82.22	1667	56.760
0.07	81.11	377	12.652	81.67	93	3.338	81.67	147	5.320

SIX (6) JOINT POSITIONS - RESULT OF CFSSUBSETEVAL NEURAL NETWORK

Table B.221 ***initialisasi variation 25 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	57.78	118	1.981	62.22	164	2.205	61.11	116	1.248
0.05	53.33	109	1.108	60.56	111	1.544	50.00	106	1.139
0.06	62.22	108	1.168	58.33	109	2.423	61.66	116	1.362
0.07	65.00	99	1.063	56.67	100	1.173	54.44	107	1.342

Table B.222 ***initialisasi variation 50 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	67.78	147	1.654	65.56	139	1.662	68.89	127	1.389
0.05	65.56	127	2.279	67.78	144	2.309	73.33	136	1.607
0.06	58.33	97	1.151	62.22	118	1.325	62.22	98	1.149
0.07	71.67	191	2.123	66.67	120	2.026	75.00	151	1.747

Table B.223 ***initialisasi variation 75 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	75.00	186	1.684	73.33	171	2.294	82.78	295	3.635
0.05	78.89	228	3.377	70.56	125	1.904	76.78	208	2.558
0.06	81.11	320	3.893	73.89	156	2.026	77.22	236	3.309
0.07	83.33	163	1.994	76.11	182	2.242	77.78	165	2.121

Table B.224 ***initialisasi variation 100 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	80.00	191	2.667	82.78	340	5.486	77.22	258	3.479
0.05	80.56	281	3.901	82.22	229	3.070	72.78	124	1.653
0.06	81.67	252	3.446	91.85	541	5.425	81.11	336	4.364
0.07	80.56	410	5.510	82.78	260	3.732	78.33	145	2.028

Table B.225 ***initialisasi variation 125 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.78	90	2.823	76.67	332	5.161	77.78	197	2.948
0.05	76.67	275	4.952	87.78	333	5.817	83.89	506	7.709
0.06	82.78	325	4.696	88.33	364	6.761	84.44	275	4.299
0.07	85.00	279	4.144	62.78	96	2.144	90.00	485	7.441

Table B.226 ***initialisasi variation 150 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	76.11	149	2.340	81.11	457	8.065	84.44	536	8.626
0.05	81.67	488	7.861	77.78	151	2.406	77.22	267	4.214
0.06	83.89	436	6.977	75.56	120	2.059	88.33	517	8.271
0.07	73.33	129	2.128	71.66	121	2.264	87.22	362	6.193

Table B.227 ***initialisasi variation 175 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.22	376	6.396	83.89	397	7.611	83.89	470	8.097
0.05	83.89	466	9.360	83.33	376	11.234	68.33	129	2.780
0.06	83.89	501	8.677	85.56	465	8.112	77.22	117	2.286
0.07	87.89	868	15.228	78.89	125	2.208	87.22	400	7.207

Table B.228 ***initialisasi variation 200 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	78.89	355	6.583	85.00	368	7.643	83.33	907	16.926
0.05	86.67	1209	23.150	80.56	395	7.592	86.67	1380	32.753
0.06	88.59	3214	59.982	81.11	686	12.964	80.56	347	6.325
0.07	86.67	428	8.127	84.44	515	11.883	83.33	327	6.489

Table B.229 ***initialisasi variation 225 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	86.67	376	7.223	85.56	366	7.966	84.44	325	6.412
0.05	86.67	435	9.173	86.11	386	7.824	82.78	293	5.712
0.06	85.00	893	18.190	83.33	561	14.442	87.22	2019	39.733
0.07	78.89	123	2.483	87.78	429	12.552	86.67	381	8.128

Table B.230 ***initialisasi variation 250 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	81.11	577	12.870	79.44	146	3.439	78.33	304	6.443
0.05	80.00	154	3.276	81.67	778	16.404	87.78	1497	31.450
0.06	82.22	943	19.315	81.67	826	18.301	85.00	277	5.913
0.07	83.33	162	3.517	85.00	529	14.174	85.56	255	5.709

Table B.231 ***Result of Spherical with Neural Network with initialisasi variation 275 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	72.22	93	2.215	89.44	522	12.598	81.67	1180	27.549
0.05	90.56	534	13.154	77.22	111	2.529	85.00	1994	45.378
0.06	84.44	801	22.589	81.67	227	5.226	79.44	312	7.084
0.07	81.11	940	21.969	85.56	431	11.747	84.44	1465	35.209

Table B.232 ***initialisasi variation 300 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	77.22	173	4.337	83.89	491	12.602	78.89	195	4.696
0.05	83.89	407	12.926	80.00	186	4.638	85.00	275	6.652
0.06	83.89	252	6.119	80.00	177	4.368	84.44	348	8.339
0.07	85.00	748	18.617	82.22	285	7.455	86.11	278	7.082

Table B.233 ***initialisasi variation 325 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	85.56	519	22.776	81.11	521	12.597	79.44	256	6.068
0.05	81.11	392	12.017	86.11	499	12.820	85.00	261	15.132
0.06	80.56	833	20.252	78.89	110	2.714	86.67	345	8.386
0.07	87.22	776	19.025	77.22	90	3.463	85.56	643	16.832

Table B.234 ***initialisasi variation 350 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	82.22	171	4.384	77.22	145	4.066	76.11	188	4.789
0.05	81.66	146	3.974	90.56	1547	42.456	71.11	109	0.138
0.06	82.22	386	9.966	83.33	108	3.260	81.11	219	5.648
0.07	77.22	80	2.074	83.89	408	10.545	85.56	352	9.532

Table B.235 ***initialisasi variation 375 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	85.00	412	11.232	84.44	370	10.619	79.44	221	6.053
0.05	86.11	144	44.523	83.89	171	4.739	82.22	484	15.083
0.06	85.56	1174	31.393	85.00	329	10.798	85.00	599	16.327
0.07	83.33	3470	94.938	69.44	79	2.153	79.44	97	2.777

Table B.236 ***initialisasi variation 400 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	83.33	610	17.675	78.33	182	5.666	84.44	175	5.055
0.05	83.89	399	23.172	83.33	667	19.483	81.11	173	5.242
0.06	86.66	619	18.090	79.44	111	3.292	85.56	978	28.198
0.07	89.44	969	28.255	77.78	106	3.074	77.78	108	3.339

Table B.237 ***initialisasi variation 425 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	83.33	765	23.213	77.22	152	5.415	81.11	270	8.174
0.05	78.89	272	8.611	83.33	436	13.341	82.78	526	16.878
0.06	83.33	1339	41.208	78.89	184	5.676	78.89	99	3.022
0.07	77.22	127	3.911	83.33	147	4.493	86.67	985	32.401

Table B.238 ***initialisasi variation 450 node hidden layer***

LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	85.56	207	6.864	82.78	322	10.707	57.22	70	2.059
0.05	84.44	720	27.690	85.00	406	16.302	83.89	448	14.442
0.06	82.78	803	25.484	85.00	1122	38.511	83.89	531	15.543
0.07	84.44	150	4.850	81.11	192	6.381	80.56	117	3.837

Table B.239 ***initialisasi variation 475 node hidden layer***

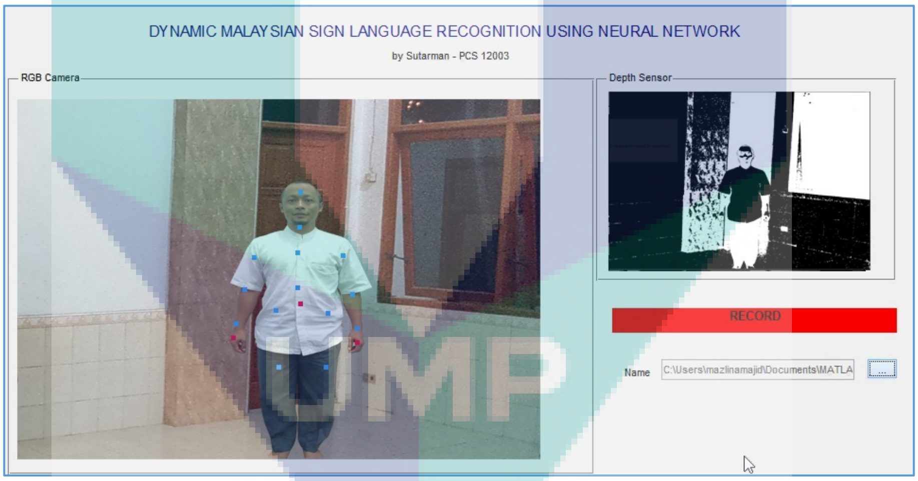
LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	80.56	198	6.443	83.33	463	16.132	83.33	809	26.552
0.05	82.78	238	8.807	84.44	5000	179.724	81.11	598	19.996
0.06	81.67	468	15.931	82.78	195	6.740	72.78	100	3.077
0.07	79.44	139	4.639	81.11	1156	39.312	83.33	513	18.439

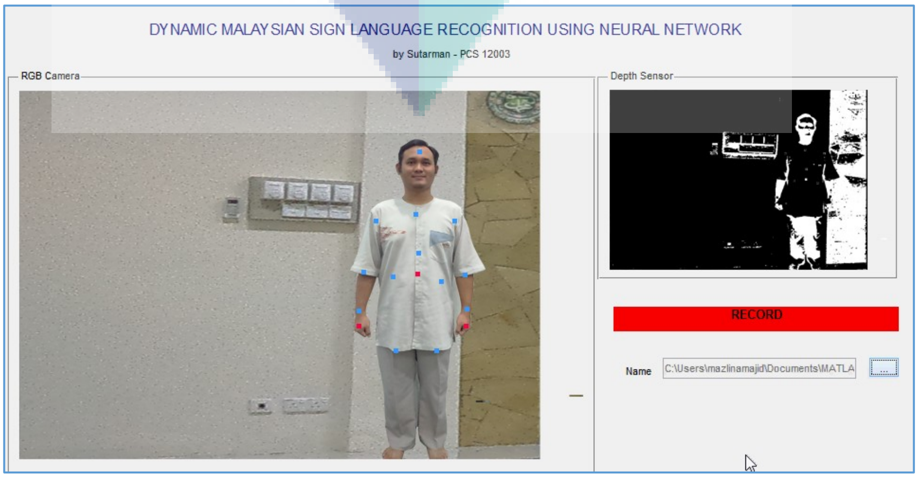
Table B.240 ***initialisasi variation 500 node hidden layer***

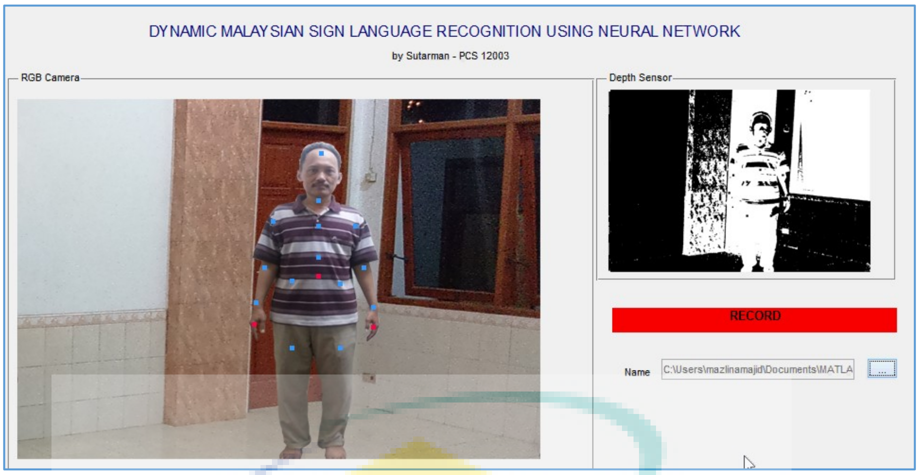
LEARNING RATE (LR)	MOMENTUM COEFFICIENTS								
	0.06			0.07			0.08		
	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time	Accuracy (%)	Epoch	Time
0.04	83.33	1778	60.544	77.22	210	7.722	81.11	217	7.426
0.05	79.44	153	5.335	82.78	215	7.467	85.00	726	25.123
0.06	85.00	260	8.947	82.22	217	8.601	83.89	191	6.494
0.07	82.78	274	9.453	79.44	123	4.399	84.44	231	8.252

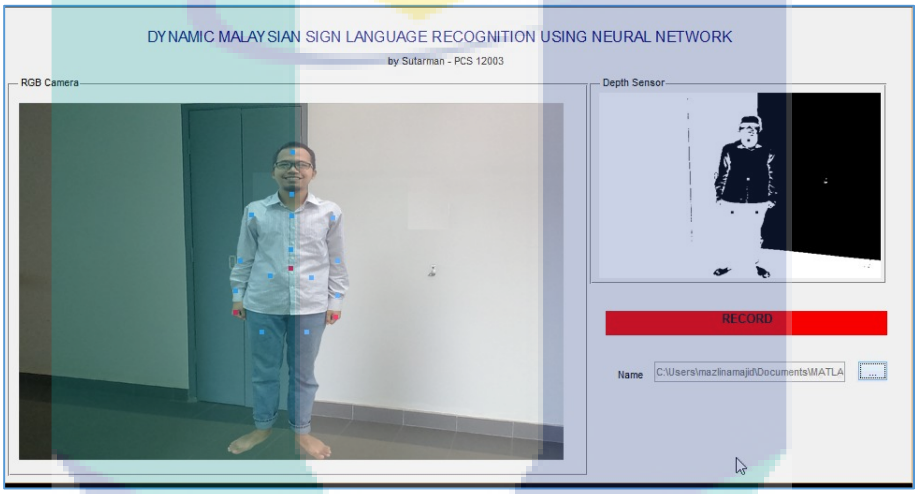
APPENDIX C

DIFFERENT PERSON USED TO CAPTURED DATA COLLECTION



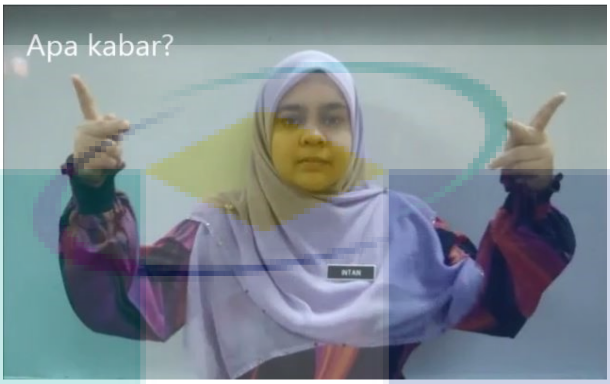



DYNAMIC MALAYSIAN SIGN LANGUAGE RECOGNITION USING NEURAL NETWORK
by Sutarman - PCS 12003
RGB Camera
Depth Sensor
RECORD
Name
C:\Users\mazlinamajid\Documents\MATLA


DYNAMIC MALAYSIAN SIGN LANGUAGE RECOGNITION USING NEURAL NETWORK
by Sutarman - PCS 12003
RGB Camera
Depth Sensor
RECORD
Name
C:\Users\mazlinamajid\Documents\MATLA

APPENDIX D

DICTIONARY OF USED SIGNS




belajar


cikgu


gembira


Kamu


makan
INTAN


mengapa
INTAN


pagi
INTAN


petang


saya


selamat


siapa


terima kasih