Indian Sign Language Recognition System for Simple Manual Signs

A Thesis submitted in fulfillment of the requirements for the award of the degree of

Doctor of Philosophy

Submitted by

Ankita Wadhawan

(Registration No. 951503010)

Under the supervision of

Dr. Parteek Kumar

Professor, TIET

Computer Science and Engineering Department

Thapar Institute of Engineering and Technology

Patiala-147004, India

February 2024

Contents

List of Figures

List of Tables

List of Abbreviations

HCI	Human Computer Interaction
ASL	American Sign Language
BANZSL	British Australian and New Zealand Sign Language
JSL	Japanese Sign Language
KSL	Korean Sign Language
TSL	Taiwanese Sign Language
ISL	Indian Sign Language
IPSL	Indo-Pakistani Sign Language
BSL	Brazilian Sign Language
NUI	Natural User Interface
ArSL	Arabic Sign Language
CSL	Chinese Sign Language
CNN	Convolutional Neural Network
GUI	Graphical User Interface
NLP	Natural Language Processing
SLR	Systematic Literature Review
HOG	Histogram of Oriented Gradient
KNN	K Nearest Neighbors
SVM	Support Vector Machines
PCA	Principle Component Analysis
LDA	Linear Discriminant Analysis
NN	Neural Network
RNN	Recurrent Neural Network
DNN	Deep Neural Network

SIFT	Scale Invariant Feature Transform
ANN	Artificial Neural Network
HMM	Hidden Markov Model
MLP	Multilayer Perceptron
LMC	Leap Motion Controller
BLSTM-NN	Bidirectional Long Short-Term Memory Neural Network
MDC	Minimum Distance Classifier
DCNN	Deep Convolutional Neural Networks
ROI	Region of Interest
LSTM	Long Short Term Memory
IMU	Inertial Measurement Unit
CapsNet	Capsule Network
SGD	Stochastic Gradient Descent
VGG	Visual Geometry Group
DTW	Dynamic Time Warping
DCT	Discrete Cosine Transform
LBP	Local Binary Patterns
GMM	Gaussian Mixture Model
RF	Random Forest
RBF	Radial Basis Function
LB-HMM	Level Building-HMM
CTC	Connectionist Temporal Classification
RGB	Red Green Blue
ELM	Extreme Learning Machines
KPCA	Kernel Principle Component Analysis

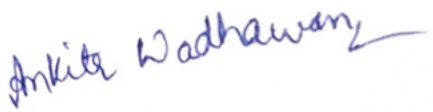
KDA	Kernel Discriminant Analysis
SSD	Single Short Detector
ESHR	Extra Spatial Hand Relation
PSL	Persian Sign Language
HP	Hand Pose
sEMG	Surface Electromyography
PHOG	Pyramid Histogram of Oriented Gradients
HRI	Human Robot Interaction
SAM	Skeleton Aware Multi-modal
GCN	Graph Convolutional Networks
SL-GCN	Sign Language-Graph Convolution Network
SSTCN	Separable Spatial Temporal Convolution Network
SDK	Software Development Kit
CMC	Carpometacarpal
MCP	Metacarpophalangeal
PIP	Proximal Interphalangeal
DIP	Distal Interphalangeal
ReLU	Rectified Linear Unit
Adam	Adaptive Moment Estimation
SLRS	Sign Language Recognition System

Acknowledgement

Firstly, I express my gratitude to the Almighty for endowing me with the desire and enthusiasm to accomplish this work successfully. I would like to thank my supervisor, Dr. Parteek Kumar, for his insightful guidance. He was always there to steer me in the proper path whenever I became stuck with my thoughts. I want to thank him for the important hours and weekends he spent working with me to complete the composition of this thesis. I shall be forever grateful to him for showing me the way I am grateful to the doctorate committee for monitoring the progress of my work and offering helpful ideas for its enhancement. I am really pleased and appreciative to the Department of Computer Science and Engineering at Thapar Institute of Engineering and Technology, Patiala, for giving all of the resources necessary for a successful research project. This dissertation would not have been feasible without the assistance of my family. My deep regards to my father Mr. Arun Kumar Wadhawan, my mother Mrs. Ruby Wadhawan, my husband Mr. Jatin Uppal and my in-laws for their blessings, inspiration, care, love and motivation. Without them, this work would have never been completed. Their humility and forbearance have always astonished me. I am also thankful to my beloved daughters Kridha and Lineysha whose smile always encouraged me to work. Finally, I would like to thank my friends Sugandhi Verma, Sujata Singla, Sawinder Kaur and Usha Mittal who have been stress busters for me.

Certificate

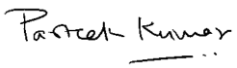
I hereby certify that the work submitted in this thesis titled "Indian Sign Language Recognition of Simple Manual Signs" in fulfilment of the requirements for the award of the degree of DOCTOR OF PHILOSOPHY in the Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology, Patiala, is an authentic record of my work carried out under the supervision of Dr. Parteek Kumar. This thesis contains content that has not been submitted for a degree at any other university.



(Ankita Wadhawan)

Regd No. 951503010

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge and belief.



(Parteek Kumar)

Professor, Computer Science and Engineering Department

Thapar Institute of Engineering and Technology, Patiala-147004 (INDIA)

Abstract

Sign language is the fundamental mode of communication among deaf community members. Each nation has its own, unique sign language. In a sign language, various hand gestures, body movements, and facial expressions are utilized to represent each sign. However, all these languages are not commonly recognized outside of these groups, there may be a communication barrier between hearing-impaired and non-hearing impaired individuals. The methods for recognizing signs created through this study enable the design of system that can help to reduce this barrier, either by giving computer tools to aid in the acquisition of sign language or, possibly, by creating portable sign-to-speech translation systems.

The research work presents the detailed description about the general process of sign language recognition system using two different datasets (static and dynamic) of signs. A systematic literature review related to the Sign Language Recognition System (SLRS) for static and dynamic signs is depicted in this research work. The current status of sign language recognition system w.r.t the dataset is classified into static and dynamic signs. On the basis of published works, the periodic development of sign language recognition and research studies has been evaluated. In addition, the review methodology is followed and provided, and sources of publications and research papers are retrieved according to inclusion-exclusion criteria. This study methodology will aid in the dissemination of results in a methodical manner, therefore allowing researchers working in comparable fields to pick the most effective strategies for recognizing static and dynamic signs of Indian sign language (ISL).

As no public dataset is available for the recognition of Indian signs this thesis presents the collection and development of datasets for static signs as well as for dynamic signs. It also describes the detailed procedure about how the dataset has been collected from the number of users under different environmental conditions and at different distances.

This thesis also presents different architectures for sign language recognition of static and dynamic signs of ISL. In this MediaPipe Hand and MediaPipe Pose techniques are used as data pre-processing for efficiently recognize static and dynamic signs. The SLRS described in this thesis is developed using deep learning based techniques. In this, different convolutional neural network architectures have been compared and the results are analyzed on the basis of accuracy, precision, recall, F1-score and loss curves. The implemented convolutional neural network architecture not only helps to enhance the accuracy of the model but also helps to increase the efficiency of the model. The experimental analysis show that the implemented model outperforms the traditional machine learning algorithms for sign language recognition. Further, to recognize Indian signs at real-time different Convolutional Neural Network (CNN) architectures like Visual Geometry Group 16 *(VGG16), VGG19 and GoogleNet are implemented and compared. It has been observed from the experimental analysis that VGG19 architecture using MediaPipe technique and CNN architecture using MediaPipe outperformed all the other CNN based architectures for static sign recognition and dynamic sign recognition respectively.*

This thesis also presents the developed Progressive Web Application of the proposed system. This web application promotes the communication between hearing-impaired and non-hearing impaired people. It serves the purpose to expedite the users to recognize different static signs in real-time. The goal to develop such an application is to outreach the hearing-impaired people to communicate with other persons in the society and learn new facts. This system can also help hearing-impaired people to get education, enhance their skills and make their career.

CHAPTER 1

Introduction

1.1 Introduction to Sign Language

Sign language is a method of communication used by hearing-impaired persons. It is used by hearing-impaired persons to express their thoughts and emotions through nonverbal communication. In the communicative hand/arm gesture taxonomies, sign language is the most structured form of several gesture types. Communication among the deaf and hearing-impaired is impossible without sign language. Instead of using various sounds and oral communication, hearing-impaired people make use of signs for communication.

Recognition of sign language is a collaborative area that involves matching of different patterns, computer vision, Natural Language Processing (NLP), and linguistics. The main objective of the SLRS is to recognize various signs and perceive their meaning. Human-Computer Interaction (HCI)-based SLRSs are developed to facilitate productive conversation. These systems follow a multidisciplinary approach of data acquisition, sign language technology, testing, and sign language linguistics. This system can be installed in public services like hotels, railway stations, airports, bus stations, resorts, banks, offices, *etc.*, to enable hearing-impaired people for their better communication and control emotional behavior [50].

1.2 Sign Language Symbols

Sign language linguistic studies started in the 1970s [24]. It is associated with linguistic data in the form of various symbols and letters. All the parameters of signs, including different forms of hands, movement, position, and orientation of palm, can be represented by sign language symbols. The categorization of sign language symbols is shown in Figure 1.1 [42]. Single-handed and double-handed signs are the two types of sign language symbols. These signs are further divided into static and dynamic sign categories.

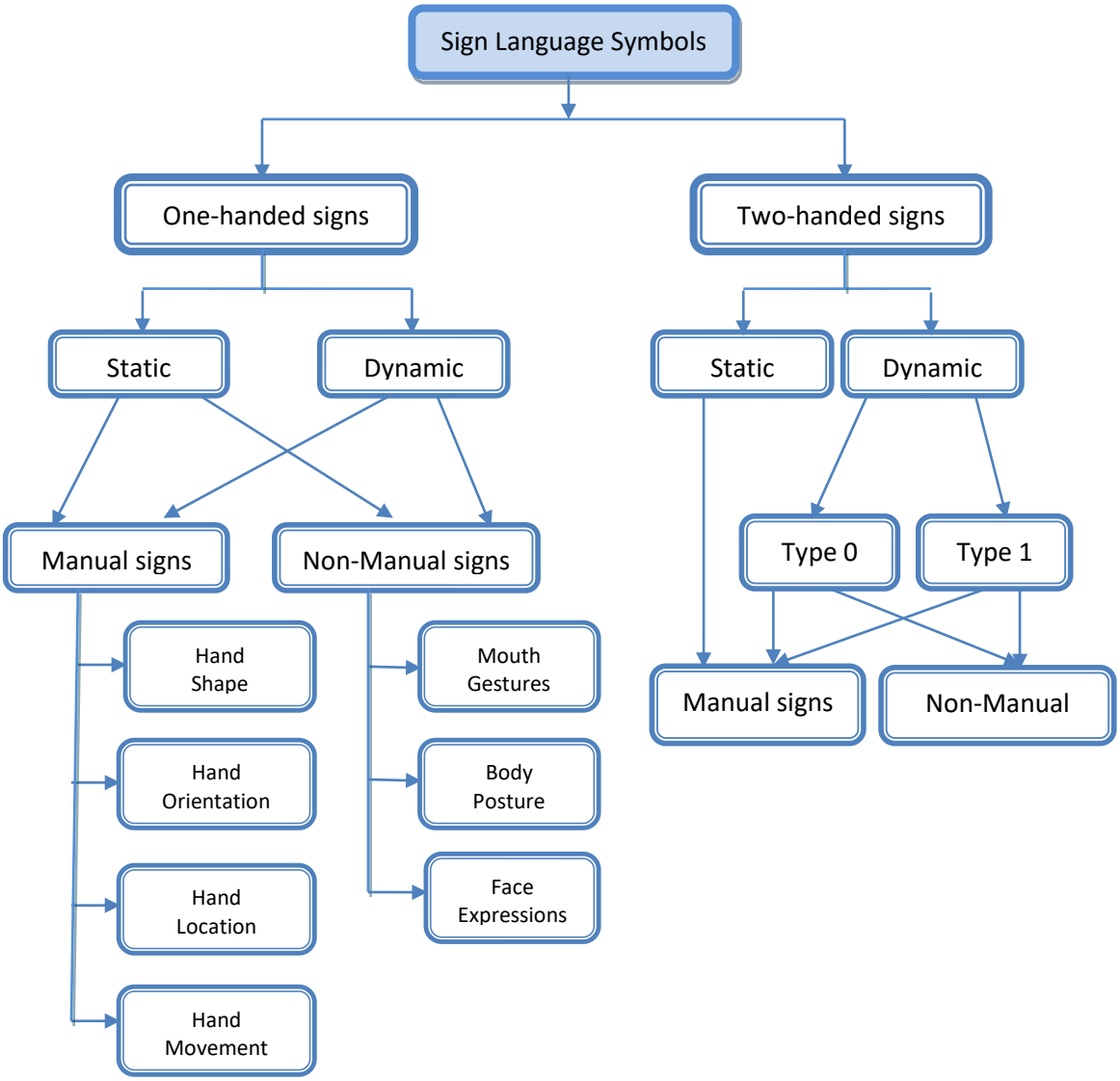


Figure 1.1: Hierarchy of signs [42]

One-handed signs: One-handed signs are depicted with a single dominant hand. Any static or moving gesture can be used to demonstrate it.

Two-handed signs: While signing, both the signs whether dominant and non-dominant are utilized to signify two-handed signs. There are two types of symbols: type 0 and type 1. Both hands are active in the type 0 symbol, as shown in Figure 1.2, whereas the dominant hand is more active than the non-dominant hand in Type 1 sign, as presented in Figure 1.3.

Figure 1.2: Two-handed Type 0 sign (both the hands are active)

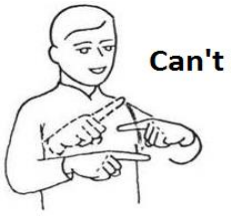


Figure 1.3: Two-handed Type 1 sign (only dominant hand is active)

Sign Language consists of manual as well as non-manual signs [18]. To represent manual signs hands are used, whereas non-manual signals incorporate postures of body, mouth gestures, and expressions of face. Manual and non-manual single-handed static signs are shown in Figures 1.4(a) and 1.4(b), respectively.

Figure 1.4(a): Single-handed static, manual sign

Figure 1.4(b): Non-manual sign [36]

1.3 Comparison of Various Sign Languages

As sign languages differ from one country to the other and from one region to the other and they are not universal. The 2021 edition of Ethnologue lists 150 sign languages [43]. These sign languages can be categorized into different families, i.e., Arab, French, German, Japanese, Swedish, Indo-Pakistani, and BANZSL. Arab family includes Iraqi, Levantine, Yemini, Egyptian, Kuwaiti, and Libyan sign languages. Western Europe, Francophone Africa, North America, and portions of Asia employ French sign languages. American Sign Language (ASL) also belongs to this family. The German language is used in Germany and within German-speaking communities in Belgium. Japanese Sign Language (JSL), Taiwanese Sign Language (TSL), and Korean Sign Language (KSL) belong to the Japanese Sign Language family. The Swedish Sign Language family includes the sign languages used in Finland, Sweden, and Portugal. Indo-Pakistani Sign

Language (IPSL), also called ISL, is the native sign language of South Asia. BANZSL stands for British, Australian, and New Zealand sign languages and is also used in Northern Ireland, South Africa, The Maritimes, Newfoundland, and Labrador [29]. In the context of sign language research, the term "number of signers" refers to the count of individuals who are proficient in using a particular sign language or dialect. These individuals are often members of the deaf or hard of hearing community. The number of individuals proficient in using a specific sign language in different countries are shown in Table 1.1[167]. Every sign in a sign language has a different semantic meaning, however there are some signs that indicate universal structures. For example, a sign 'goodbye' or 'hi' is having similar meaning in all the sign languages available worldwide.

Table 1.1: Estimated number of individuals using specific sign language [167]

Language	Country	Estimate
Brazilian Sign Language	Brazil	6,30,000 in year 2021
Indo-Pakistani Sign Language	India, Pakistan, Bangladesh	60,00,000 in India in year 2021
American Sign Language	USA	3,25,000 in year 2019
Hungarian Sign Language	Hungary	3,00,000 in year 2014
Kenyan Sign Language	Kenya	6,00,000 in year 2011
Japanese Sign Language	Japan	1,26,000 in year 2021
British Sign Language	UK	77,000 in year 2014
Russian Sign Language	Russia	120,000 in year 2010
Philippine Sign Language	Philippines	3,25,000-650,000 in year 2021

The primary finger spelling systems of ISL, ASL, and Brazilian Sign Language (BSL) are different. For example, BSL utilizes two hands to symbolize ‘WHERE’, ASL uses

only one hand, and ISL employs hand movement in both directions to signify ‘WHERE’, as represented in Figure 1.5.

(a) (b) (c)

Figure 1.5: (a) Sign ‘WHERE’ in BSL (b) Sign ‘WHERE’ in ASL (c) Sign ‘WHERE’ in ISL

Figure 1.6(a) represents ‘WOMAN’ sign symbol in BSL. In ASL, ‘WOMAN’ sign is denoted by using single hand in which the signer has to touch the tip of the thumb with the chin and then bringing it down to the chest as in Figure 1.6 (b) and Figure 1.6 (c) shows the sign of ‘WOMAN’ in ISL.

(a) (b) (c)

Figure 1.6: Sign representing ‘WOMAN’ (a) BSL (b) ASL (c) ISL

1.4 Sign Language Recognition

The main goal of a SLRS is to classify the signs played by the signer in real-time. This kind of system is beneficial in eradicating the communication barrier between hearing-impaired and non-hearing impaired people.

1.4.1 Need for Sign Language Recognition

The SLRS helps to identify various signs and is used by the deaf community for communication with non-hearing impaired persons. The need for SLRSs in various domains is discussed below.

i. A HCI system is required to eliminate the communication barrier between hearing-impaired and non-hearing impaired persons.

ii. Very few number of schools are available for educating hearing-impaired people. These schools do not have adequate resources and technology to provide knowledge and educate hearing-impaired people through sign language. So, there is a need for a SLRS that enables school children to learn and grow.

iii. One can reserve hotel rooms and resorts, make ticket reservations in railway, airlines, *etc* [30].

iv. It can be used for depositing electricity and telephone bills.

v. It is difficult for most people unfamiliar with sign language to convey messages without an interpreter. So there is a need for a sign language interpreter that helps eradicate the communication barrier among hearing-impaired and non-hearing impaired people.

vi. A SLRS imparts information, knowledge, and education to hearing-impaired people to grow and make a career [19].

vii. HCI system reduces the cost of educated and experienced tutors [59].

viii. To facilitate hearing-impaired people to perform their daily routine activities for the progress of intelligent action.

ix. To facilitate hearing-impaired people to learn new concepts and facts and control emotional behavior.

x. To facilitate hearing-impaired people to socialize among other persons in society [180].

Thus there is an urgent need for SLRSs as these are the HCI-based systems that provide facilities for hearing-impaired people to get knowledge, education, learn new facts and help them in making their careers.

1.4.2 Applications of Sign Language Recognition

SLRSs may be used in different application areas, as discussed below.

i. Education

Hearing-impaired students in education can use SLRSs. It helps in learning new concepts and essential information. Computer programs could help the parents of hearing-impaired children learn sign language and enhance their interest and confidence in learning.

ii. Healthcare

SLRSs can be used in health care settings to perform medical procedures, describe discharge and follow-up plans, explain medicine prescriptions, provide emergency care, and so on. With the help of this system, a hearing-impaired person can conduct meetings and consultations.

iii. Reservations

It can be used for booking hotel rooms, reserving resorts, and table booking in restaurants. For ticket reservations, a SLRS can be used on railway stations, bus stations, airports, cinema halls, and other places that help hearing-impaired people make reservations.

v. Professions

It can be used by hearing-impaired people for communication among police officers, advocates, civic servants, vendors, stock traders, restaurant managers, and many more.

vi. Arithmetic Computation

It can be used for making arithmetic computations. For this system, hand signs are used to give inputs to the system instead of a keyboard and mouse.

vii. Supermarkets and Restaurants

It can be used in supermarkets for buying groceries and other products. Hearing-impaired persons can talk to the non-hearing impaired person through a sign language interpreter and resolve their query. SLRSs can also be deployed in restaurants to eradicate the communication barrier between hearing-impaired and non-hearing impaired persons.

ix. Video Gaming

It can be used in video gaming and enhance the gaming experience by incorporating signs such as head movement, entire body movement, or finger-pointing [25].

x. Augmented reality

In the case of virtual reality, signs are helpful in manipulation and navigation [25].

In the next section, the general architecture of SLRS has been discussed.

1.4.3 General Architecture of SLRS

Most SLRSs follow a general architecture for identifying different signs, as shown in Figure 1.7 [13]. This architecture comprises five phases: sign acquisition, preprocessing of data, feature extraction, training, and testing. Each stage has its importance and is equally helpful in recognizing different signs.

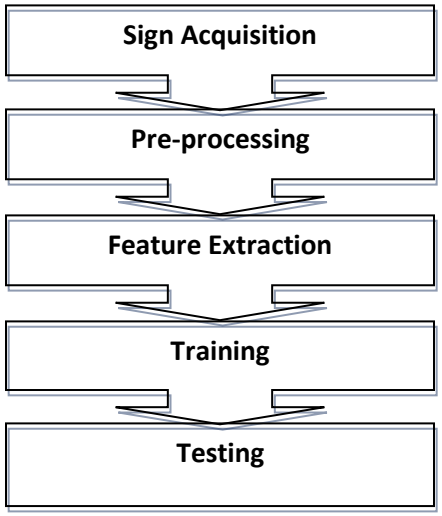


Figure 1.7: General Architecture of SLRS

The working of each phase is as follows.

Sign Acquisition: The initial step in sign language recognition is to capture input data. Various input devices used for data acquisition are arm bands, data gloves, mobile cameras, web cameras, and Kinect. The sensor-based and vision-based approaches are the two most popular data acquisition methods. The sensor-based acquisition includes data gloves and armbands, whereas vision-based acquisition includes web camera, mobile camera, and Kinect.

Pre-processing: Pre-processing phase is associated with eliminating unwanted and useless information in the context of feature extraction. This phase improves the valuable data and gets rid of unwanted data.

Feature Extraction: The feature extraction phase involves extracting features from the input data. It collects all the features from a sign and stores them as a feature vector.

Training: Training is feeding a machine learning algorithm with data retrieved from the previous phase and categorizing the data in one or another class. There are various machine learning algorithms, with supervised and unsupervised learning being the most common. As the collected data contains labels, models are trained using a supervised algorithm. Other than this, unsupervised learning involves determining patterns in the data, and the output is unknown.

Testing: This phase will identify the sign entirely and retrieve a suitable output with its meaning. In this phase, the matching of the trained model with the input data has been performed. After that, it will give the result of a specific sign with which the test data gets matched [25].

1.4.4 Challenges of Sign Language Recognition

The following are the primary challenges in designing a SLRS.

i. **Gesture Spotting:** Gesture spot is the recognition of hand movements while recognizing gestures. It is the biggest challenge as gesture spotting requires the hands to be detected from the background, and then gesture path points should be obtained [30].

ii. **Occlusion by other body parts:** Occlusion occurs when some fingers or hand portions would be disappeared or get hidden by some other parts of the body, due to which signs cannot be recognized accurately. The hand is often in front of the face, which causes uncertainty during the recognition process [180].

iii. **Change in Environment:** Clothing constraints like wearing short and long sleeves and cluttered backgrounds are common issues in the SLRS [180]. Sometimes a person is wearing light-colored clothes nearly similar to his skin, which causes difficulty in recognizing signs.

iv. **Between signer variations:** Everyone is different and has their way of performing sign. So, two signers may use various hand movements, orientations, and shapes to perform the same sign [180].

v. **Lightening changes:** The system must cope with different lightening conditions, as illumination changes affect the recognition results.

vi. **Tracking:** It is difficult to track and find the position of hands when they are very close. So, a robust hand detection algorithm should be required to track hands and recognize signs.

Developing a SLRS is challenging as it introduces various challenges like hand detection, tracking, lightning effects, occlusion, *etc.* So, there is a need to take care of these challenges for building a scalable and robust SLRS.

1.5 Gap Analysis

As discussed, there is an urgent need to develop a SLRS, as hearing impairment is a ubiquitous problem worldwide. Hearing-impaired persons found difficulty in communicating with the non-hearing impaired persons without a sign language interpreter and feel difficulties in performing their daily activities. Based on this and the challenges associated with data acquisition, some of the research gaps diagnosed for the proposed research work are presented below.

i. No detailed survey on the sign language recognition technique exists. So, there is a need for a thorough and systematic literature review that helps various researchers review the state-of-the-art algorithms and work done on different SLRSs.

ii. Some researchers use hand gloves/web cameras for sign language recognition. But, these systems suffer from limitations as the use of data gloves is expensive, and it restricts the natural movements of the hand and makes the user uncomfortable. There is an urgent need to overcome these data capturing limitations for collecting Indian signs.

iii. No standard image dataset is available publically for performing experimentation on static signs for Indian Sign Language. So, there is an urgent need to collect and develop an image dataset for sign language recognition of static signs.

iv. No standard video dataset is publicly accessible for recognizing dynamic signs for ISL. So, collecting and developing a video-based dataset for ISL recognition of dynamic signs is required.

v. There are many limitations associated with hand glove-based systems like signer has to wear the hardware sensor, complex signal processing algorithms are required for extracting gestures from the captured data information, and different people may have varying hand sizes, finger thickness, and height. While recognition, other locations of fingers for various users may overlap, which results in reduced accuracy. The outputs generated by using wearable devices lead to the noise as wear and tear [45], extension and flexion of wrist movements [41] [193], hand grip force [193], and poor calibration [78]. The limitations of a hand glove-based system can be removed by using a web camera. No dataset is publically available using a camera so it is required to collect and develop a dataset of Indian signs. There are many real-time sign language system exists, but they still face various challenges. These challenges include variations in signing styles, lighting conditions, background noise, and the need for continuous improvement in accuracy and robustness. So there is required to overcome these challenges.

vi. Most of the existing SLRSs using the web and mobile cameras for data capturing are sensitive to light conditions and cluttered backgrounds. It is required to eradicate these data capturing limitations and create a more robust SLRS.

vii. Established state-of-the-art literature is mainly based upon recognizing alphabets, digits, and few words. Therefore, the focus should be on recognizing more static and dynamic signs in real-time scenarios.

viii. Improvement of various algorithms used for static and dynamic hand recognition is required to extract more features and build a more robust system with larger datasets.

ix. There is a need for a web/mobile-based user interface for recognizing sign language and deploying it in various application areas like reservation systems,

hospitals, banks, restaurants, *etc.*, to communicate with hearing-impaired people effectively.

1.6 Research Objectives

The suggested research work's primary goal is to build and create a SLRS that can recognize simple Indian Sign Language manual signs. The following objectives are proposed for completing this work.

i. To study and analyze existing systems and techniques for sign language recognition.
ii. To perform data acquisition for collecting and developing datasets for Indian signs.
iii. To propose, implement and validate an Indian SLRS for simple manual signs of Indian Sign Language.
iv. To develop an interface for recognizing simple manual signs of Indian Sign Language in real-time.

1.7 Research Methodology

The following approach was used to fulfill the research work objectives.

i) To achieve the first objective.

- A comprehensive survey was conducted to comprehensively classify and compare all known methodologies and techniques for sign language recognition.
- A detailed analysis of different types of sign languages spread worldwide has been performed to learn existing SLRSs.
- A systematic review approach related to the research work has been followed from the various renowned electronic conferences, journals (national and international), and databases related to the research field.
- A systematic literature review has been performed that provides an academic literature database between 2007-2021 and proposes a classification mechanism to classify the research papers.
- The impact of research studies on different sign languages is compared based on six dimensions data acquisition devices, single/double-handed signs,

static/dynamic signs, isolated/continuous signs, classification technique, and recognition rate.

- The review for different sign languages like ASL, ISL, ArSL, CSL, Persian, Brazilian, Greek, Irish, Malaysian, Mexican, Taiwanese, Thai, German, Japanese, South African, Sri Lankan, Auslan, Bangladeshi, Ecuadorian, Ethiopian, Farsi, Italian, Polish, Spanish and Ukrainian Sign Languages have also been analyzed and documented.
- The various approaches for sign language identification utilized by researchers have been analyzed.
- The results are presented and analyzed using graphs based on data acquisition devices, single/double-handed signs, static/dynamic signs, isolated/continuous signs, classification techniques, and recognition rates for different sign languages.

ii) To achieve the second objective.

- A brief study has been conducted on wearable computing-based and vision-based acquisition devices.
- Different data acquisition devices have been analyzed and selected based on data capturing limitations.
- An image-based Dataset for static signs containing 35,000 sign images of 100 signs has been collected and developed from 35 different persons to recognize static, manual signs for the Indian sign language.
- A ground Dataset for dynamic signs containing 9,500 videos of 50 dynamic signs has been collected and developed from 19 different persons to recognize dynamic manual signs.
- Both datasets have also presented a detailed description of the experimental setup, camera setup, illumination setup, subjects, stimuli, and experiment procedure.

iii) To achieve the third objective.

- The collected static and dynamic signs are preprocessed using MediaPipe hands and MediaPipe pose techniques.

- The system has been trained on various deep learning approaches using different parameters to identify signs.
- An ideal classifier has been proposed for recognition of static signs and verify the model's effectiveness.
- A CNN based classification model is also proposed to identify dynamic signs and verify the model's effectiveness.
- The efficiency of the proposed model has been evaluated and compared with already existing deep learning models.
- The classification performance of the proposed system has been compared with existing state-of-the-art approaches.

iv) To achieve the fourth objective.

- A web-based application has been developed for the identification of sign language.
- A mobile-based application has also been developed to recognize static signs.
- The system can identify commonly used static signs and present results in the form of text.

1.8 Contribution to Thesis

Our main contribution to this thesis is to develop a SLRS for ISL. The overall contribution of this thesis is as follows.

i. Development of dataset: As no dataset of Indian signs is available publically, this thesis contributes to the collection and development of dataset for ISL. Firstly, a static dataset containing 35,000 sign images of 100 signs has been collected and developed using camera for Indian Sign Language. Another dataset of dynamic signs containing 9,500 video clips of 50 signs has been collected and developed using camera from 19 persons for Indian Sign Language.

ii. Novel Sign Language Recognition Architecture: One of the primary technical contributions of this research is the development of a novel CNN architecture. Various deep learning-based models are used to identify the best CNN algorithm for recognition of both static and dynamic signs. A novel method using CNN is

proposed to identify signs. The proposed deep learning architecture is compared and analyzed with other existing deep learning-based architectures like VGG16, VGG19, GoogleNet.

iii. Hyperparameter tuning and optimization: Hyperparameter tuning in CNN is a crucial step in achieving optimal model performance. In proposed work, the performance of CNN models is analyzed based on different hyperparameters like number of layers, number of epochs and kernel size. The proposed model is also tested based on different optimizers like SGD, Adam, RMSProp, Adadelta and Adagrad to minimize the loss.

iv. Efficient Real-Time Recognition: The development of efficient algorithms and optimizations for real-time sign language recognition using camera-based input is another technical contribution. A real-time web/mobile-based application has been developed to recognize signs in Indian Sign Language. The developed application is having intuitive design that allows users to interact with it easily.

1.9 Thesis Organization

The thesis has been arranged into seven chapters. A brief overview of all the chapters is as follows.

Chapter 1 covers the introduction of sign language, highlighting the concept of various sign language symbols like single-handed signs, double-handed signs, and manual and non-manual signs. After that, the comparison of different sign languages existing worldwide is presented. The need for a SLRS has also been documented in this chapter. The various applications of SLRSs in different domains such as education, courts, reservation, restaurants, supermarkets, *etc.*, are also discussed in detail. The general process of sign language recognition, which consists of five phases sign acquisition, preprocessing, feature extraction, training, and testing, is also presented in this chapter. The last section discusses about the challenges like gesture spotting, occlusion by other body parts, lightning changes, tracking, *etc*. The chapter concludes with thesis organization, research gaps, research methodology, research objectives, and thesis contributions.

Chapter 2 includes a systematic literature review of SLRSs. This systematic review has been conducted by following a review methodology. The review reported in this chapter was conducted by locating relevant research studies from well-known electronic resources and the most important conferences in the field. After this, inclusion and exclusion criteria were followed to cut down the count of selected studies. The conclusive research projects were chosen based on the design of research questions, and the results were produced following a thorough examination. The Literature Review is based upon various parameters like acquisition mode, static/dynamic signs, signing mode, single/double-handed signs, techniques used, and average accuracy. Based on these parameters, the review for different sign languages like ASL, ISL, ArSL, CSL, Persian, Brazilian, Greek, Irish, Malaysian, Mexican, Taiwanese, Thai, German, Japanese, South African, Sri Lankan, Auslan, Bangladeshi, Ecuadorian, Ethiopian, Farsi, Italian, Polish, Spanish and Ukrainian Sign Languages have been analyzed and documented respectively. The percentage of the status of research works for different parameters acquisition mode, signing mode, static/dynamic signs, single/double-handed signs, techniques used, and average accuracy has been presented graphically using pie charts which helps the researchers to get the information about the state-of-the-work carried out in the field of sign language recognition.

Chapter 3 explains the detailed information on wearable computing-based and vision-based acquisition devices. It also compares wearable computing-based devices, including hand gloves and armbands, and vision-based devices, including Kinect and web/mobile camera. Based on all the comparisons, the camera has been chosen as an acquisition device for recognizing signs. Further, it starts with collecting and developing the dataset required to implement the SLRS. This chapter describes establishing a new dataset containing signs of male and female participants of different age groups to recognize signs in Indian Sign Language. In this chapter, two datasets have been presented. Dataset for static signs consists of 35,000 sign images collected from 100 static signs of Indian Sign Language.

On the other hand, the dataset for dynamic signs consists of 9,500 video clips collected from 50 dynamic signs of Indian Sign Language. This chapter has also documented a detailed description of the collected dataset, camera setup, and the number of participants. It also demonstrates the categories of signs as single-handed signs, double-handed signs, facial expressions, single-handed signs with a face, and double-handed signs with the face. This chapter also explains how the dataset has been created and collected from users under different environmental conditions**.**

Chapter 4 explains the detailed information of the models used to detect and track hands, and various poses have been documented. Initially, in this chapter, the concept of preprocessing has been explained for processing the collected static and dynamic signs. This chapter focuses on the MediaPipe technique used for data preprocessing the given sign images and videos. The detailed description of MediaPipe hands and MediaPipe pose models is also documented. It also explains the solution APIs used for getting the output from all the models used for preprocessing. The preprocessed sign images and videos obtained after applying both MediaPipe hands and the MediaPipe pose model on all the signs are also documented in this chapter.

Chapter 5 describes the overall design and implementation of a SLRS to detect signs from Indian Sign Language. The proposed architecture of a SLRS that identifies static and dynamic signs has been presented. Firstly, two different CNN based models have been introduced to recognize static signs. The first one is the basic CNN model; in the second model, MediaPipe and CNN have been used to acknowledge static signs.

The first model is the basic CNN model, consisting of the input layer, two convolution layers, one max-pooling layer, and two fully connected layers. This model has been proposed for the recognition of static signs. The second model is also trained using CNN, but MediaPipe is used for preprocessing the collected sign images. This model has been developed to recognize static sign images available in the static sign dataset. Further, this chapter has also documented the comparison of the proposed system with other CNN-based architectures like VGG16, VGG19, and GoogleNet. The CNN model was also provided and tested with various parameter settings to examine its performance. Lastly,

all the models for recognition of static signs are compared, and finally, the VGG19 model using MediaPipe has been selected for developing a web/mobile interface.

Secondly, CNN architecture training and testing have been performed on dynamic sign datasets to recognize dynamic signs. The collected videos are divided into frames and then preprocessed using MediaPipe technology. After this, the 3D CNN model extracted different features and recognized dynamic signs using video clips as input. Further, the experiments have been performed using VGG16, VGG19, and GoogleNet on a dataset for dynamic signs. Different performance metrics like precision, recall, accuracy and f1-score, have been used to examine the results provided by the suggested SLRS. The system's performance has also been compared to that of other systems.

Chapter 6 presents the detailed online web-based and mobile based Graphical User Interface (GUI) of the SLRS for Indian Sign Language. This web-based system performs the recognition of static signs. For recognition of static signs, an image is used as input, or one can also capture the sign in real-time. Further the trained deep learning model has been used to recognize static signs in real-time.

Chapter 7 concludes the research given in this thesis and discusses the future direction of research work. This chapter concludes that the system's results are promising and can be used to benefit society. The system's accuracy can be improved in the future by increasing the dataset size and adding more signs. It has been concluded that the developed SLRS has huge potential and it can be extended to domain specific applications like railway stations, restaurants, hotels, airports, etc.

Chapter Summary

In this chapter, the introduction to SLRS and sign language symbols has been documented. As different sign languages exist all over the world, therefore the comparison of various sign languages has been described in detail in this chapter. This chapter highlights the importance and use of sign language systems in different applications. It explains the working principle of sign language recognition systems and the challenges faced by hearing impaired people in brief. It also highlights the research gaps and the objectives that were framed for this thesis. After that the major contributions of this research work have been documented. This chapter also discusses the complete thesis write up plan.

CHAPTER 2

Literature Review

In recent years, significant progress has been achieved in the study of sign language recognition. Developing a successful SLRS requires expertise in various fields, including computer vision, Natural Language Processing (NLP), HCI, computer graphics, linguistics, and deaf culture. Its objective is to build various methods and algorithms to identify signs and perceive their meaning. Systems based on HCI that recognize sign language are intended to promote efficient and exciting communication. These systems follow a multidisciplinary approach of data acquisition, sign language technology, training, testing, and language linguistics.

This chapter focuses on analyzing and studying various SLRS. This systematic literature survey provides an academic database of works published between 2007 and 2021. This search retrieved over four hundred and sixty-six research articles, which were then reduced to four hundred and three research articles based on their titles, two hundred and seventy-six research articles based on their abstracts and conclusions, and one hundred and seventy-nine research papers based on the entire text as the criterion of inclusion and exclusion. Each 179 selected papers was categorized based on different sign languages and compared based on six parameters (data acquisition techniques, signing mode, static/dynamic signs, classification technique, single/double-handed signs, and recognition rate). After doing an exhaustive survey on various SLRS, it has been found that the proposed SLRS will be a significant step that helps in removing the communication barrier.

2.1 Research Methodology

The research methodology consists of a philosophical analysis of all the assumptions associated with the particular field of study. The demand for researchers to thoroughly and objectively describe all available knowledge about a phenomenon raises the need for reviews. This could be done to derive more general conclusions about a phenomenon or as a stepping stone to more research projects. Generally, it includes the concepts of different phases, models, qualitative and quantitative techniques. This research follows the review

process suggested by Kitchenham and Charters (2007), which includes planning, conducting, and reporting the review, as shown in Figure 2.1. The stages of this literature review are to create a framework for the review process, execute the survey, investigate review results, record the review results and explore various research challenges [84].

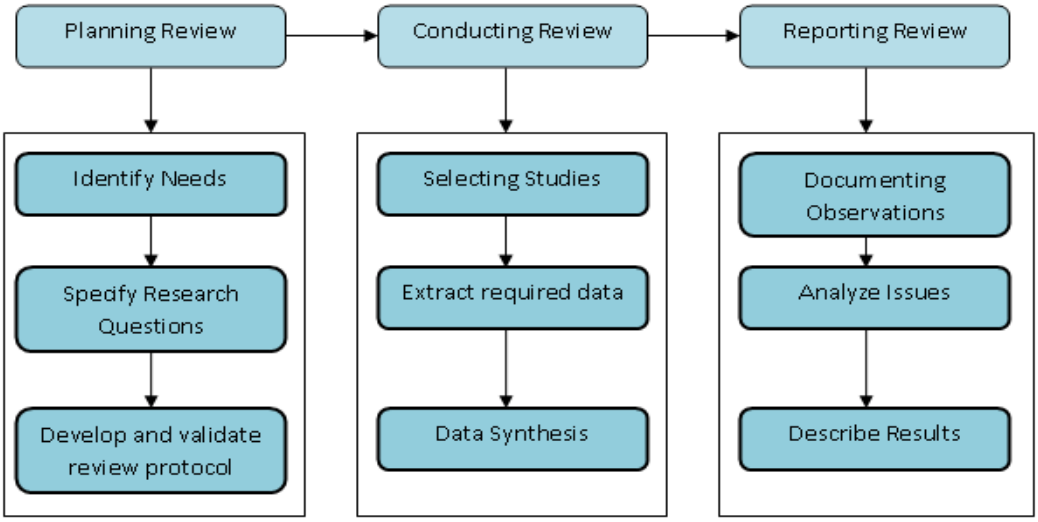


Figure 2.1: Overview of research methodology [31]

2.1.1 Planning Review

The planning process begins with determining the requirements for a Systematic Literature Review (SLR) and finishes with the creation and validation of the review methodology. The top conferences and electronic databases related to SLRS are considered to conduct a literature review. In the next step, the research questions are formulated, and further, an inclusion-exclusion criterion has been applied to extract the useful research articles.

To start with the search process, an electronic database of journals, conferences, and magazines like Google Scholar, IEEE Xplore, ACM Digital Library, and Science Direct were explored. Many surveys on sign language recognition have been noticed in the last years, but none of them target the SLR process. The main aim of this systematic review is to find and categorize the existing literature emphasizing different sign languages, SLRS, and techniques used to perform sign language recognition. So, this literature presents a comprehensive survey to systematically categorize and compare all the existing methods and approaches to

sign language recognition. To plan the review, a set of research questions were required. The research questions framed to perform SLR are presented in Table 2.1 below.

Table 2.1: Research Questions and their Motivation

Research Questions	Motivation
RQ1: Which data acquisition devices have been used mostly for capturing signs in SLRSs?	Identify and analyze various data acquisition devices required to capture data for sign language recognition.
RQ2: How much research is being carried out on static/dynamic signs in SLRSs?	To classify static/dynamic signs based on the research on sign language recognition.
RQ3: What signing techniques are taken into account when using sign language?	To identify different signing modes like isolated and continuous signs.
RQ4: How to classify and identify single and double-handed signs for SLRS?	To identify the work done on single and double-handed signs.
RQ5: What existing methodologies and techniques are available for recognizing sign language?	To identify and compare the existing methodologies and techniques used for sign language recognition.
RQ6: What is the accuracy and coverage of existing SLRSs?	To identify the recognition rate of existing SLRSs on the trained dataset.

2.1.2 Conducting Review

This stage involves selecting the studies, extracting required data, and synthesizing information. Selecting studies aims to choose the time frame for the review process. The SLR includes research papers published in the last fourteen years, from 2007 to 2021. It covers the research articles from conferences, workshops, symposiums, journals, and magazines. The studies were explored, and inclusion/exclusion criteria have been framed as shown in Figure 2.2 to select different papers. The "Sign Language Recognition" keyword has been used to search the research papers on the databases. The search fetched 466 research papers on sign language recognition from the sources of IEEE, ACM, Elsevier, and Springer. 403 research articles were extracted on the basis of titles, further research articles were declined to 276 based on their abstracts and conclusions, and 179 research articles were finally considered based on the entire text as shown in Figure 2.2.

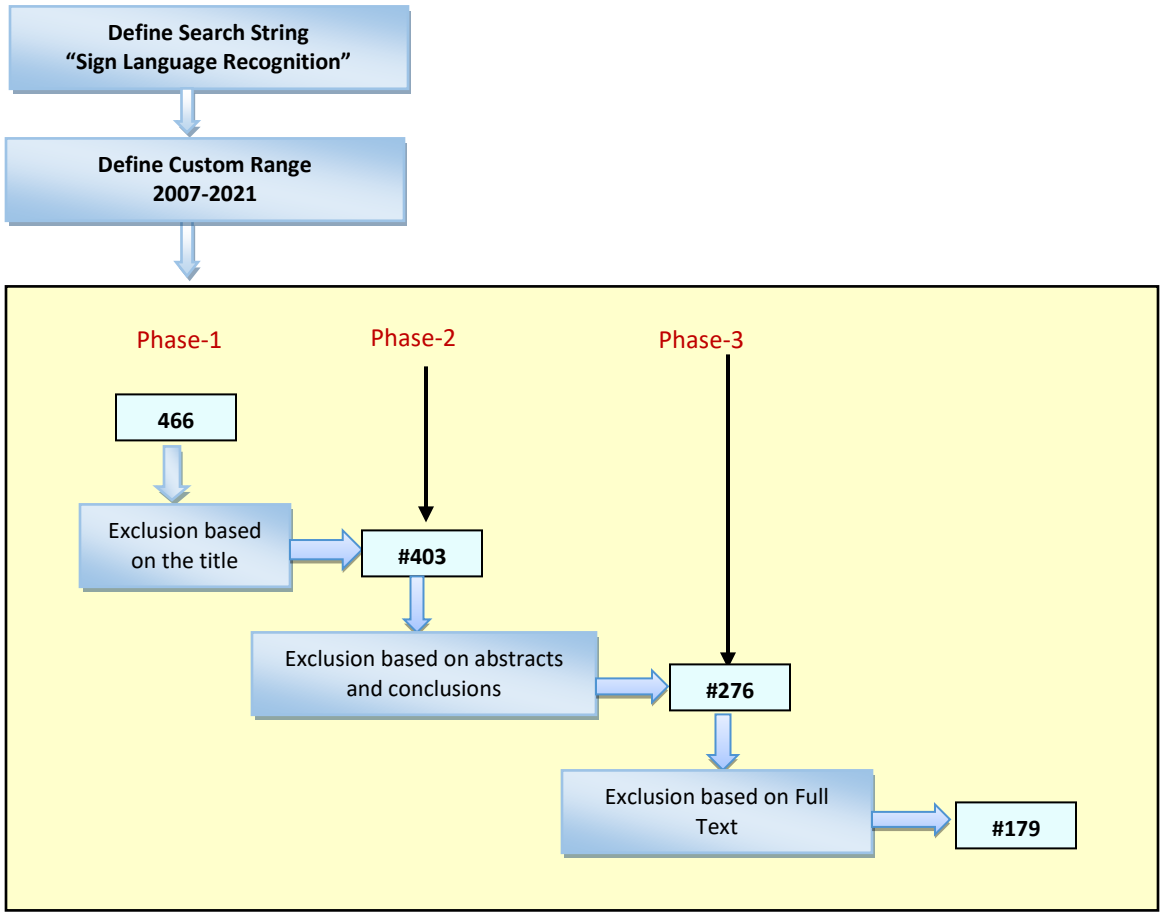


Figure 2.2: Inclusion/ Exclusion technique used in this systematic review

2.1.3 Extraction Outcomes

The main aim of this SLR is to analyze the increase in research studies related to sign language recognition year-wise. Figure 2.3 represents the source of publications and year-wise status of research studies, where the year-wise list of publications is evaluated for SLRS.

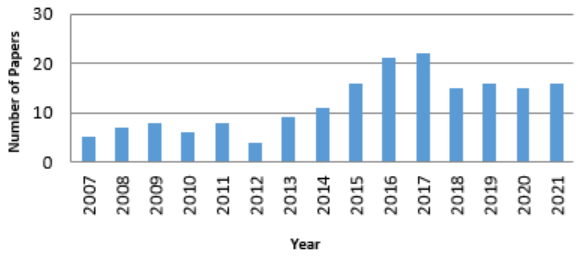


Figure: 2.3: Year-wise Number of Papers from 2007-2021

2.2 Comparative Analysis Based on Sign Languages

In this systematic literature review, a comparative analysis based on different sign languages that exist all over the world has been made. The strategy followed for the literature review includes acquisition mode, signing mode, single/double-handed signs, static/dynamic signs, techniques used, and average accuracy as their parameters are shown in Figure 2.4. Based on these parameters, the review for different sign languages like ASL, Thai Sign Language, ISL, Arabic Sign Language (ArSL), Chinese Sign Language (CSL), Persian, Brazilian, Greek, Irish, Malaysian, Mexican, Taiwanese, German, Japanese, South African, Sri Lankan, Auslan, Bangladeshi, Ecuadorian, Ethiopian, Farsi, Italian, Polish, Spanish and Ukrainian Sign Languages have been analyzed and documented respectively.

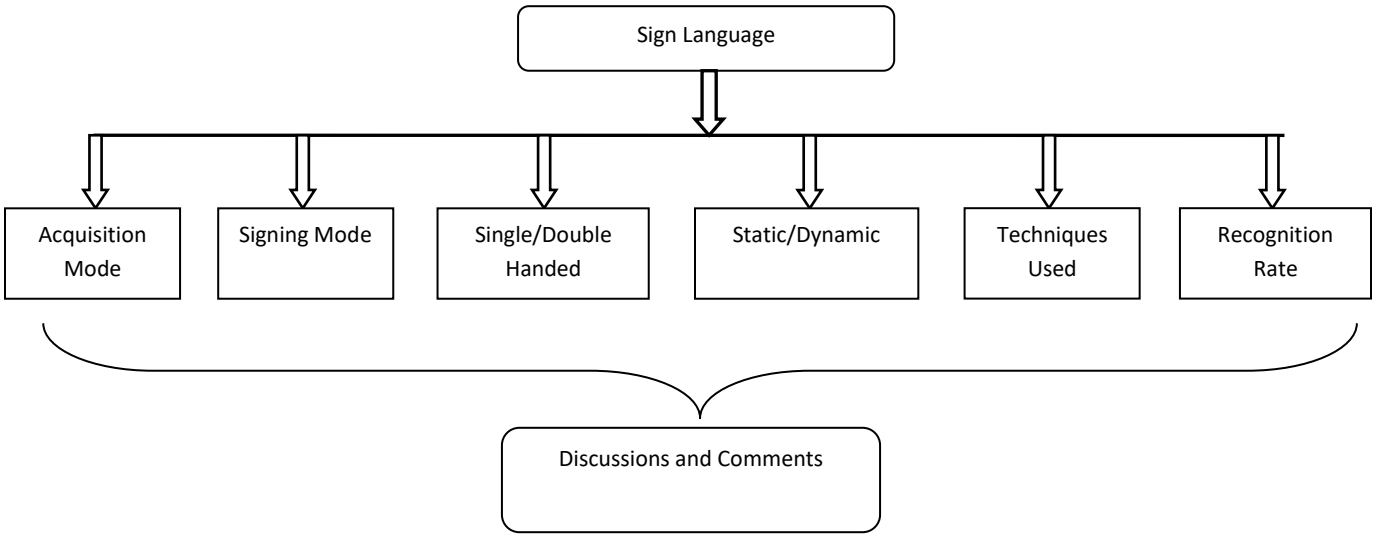


Figure 2.4: Comparison Parameters

2.2.1 American Sign Language

Most Anglophone Canada and the United States utilize American Sign Language (ASL), derived from French sign language. The estimate for ASL users range from 2,50,000 to 5,00,000 [113]. The dialects of ASL are also used in the west and central Africa. Signs in ASL consist of several phonemic components, including face and hand movements. The work done on American Sign Language recognition is presented in the next subsection.

2.2.1.1 American Sign Language Recognition Techniques

Munib *et al.* (2007) presented recognition of ASL for static words using Hough transform. They collected 300 samples of 20 sign images and extracted features of reference origin, shape orientation, and orthogonal scale factors. The results showed that the developed method is robust against changes in size, position, and direction of signs. Oz and Leu (2007) used a Cyberglove and a 'Flock of Birds', *i.e.*, a 3D-based motion tracker, to extract hand features. To get the joint angle values of fingers, Cyberglove was used. Features like shape of the hand, hand location, its orientation, its bounding box, movement, and its distance were extracted. The system was trained and tested by using an Artificial Neural Network (ANN) to identify American Sign Language, and the accuracy of 95% was achieved with a word recognition network.

Oz and Leu (2011) developed a recognition system using sensory gloves to convert ASL words into English. They collected static single-handed samples from 50 words and obtained an accuracy of 90% using the neural network classification technique.

Ragab *et al.* (2013) proposed a recognition method for one-handed static sign images acquired with the camera. To preserve the localization property of image pixels, Hilbert space-filling curve technique is adopted for feature extraction. This further helps in improved efficiency for representing shapes with uniform backgrounds. Sun *et al.* (2013a) developed a Latent Support Vector Machine (LSVM) model for classifying American sentences. They extracted Kinect features, Histogram of Oriented Gradient (HOG), and optic flow features, and the accuracy of 86% was achieved for continuous signs. Sun *et al.* (2013b) proposed an exemplar coding method for recognizing ASL. The dataset consisted of 1971 samples collected from 73 signs, and features of body pose, hand motion, and hand shape were extracted. The system was trained with mi-Support Vector Machine (mi-SVM), and the collected data were classified using Adaboost.

Chuan *et al.* (2014) proposed a system for recognizing ASL using a leap motion sensor. They used Support Vector Machines (SVM) and K-Nearest Neighbor (KNN) for classification, and the accuracy of 79.83% and 72.78% were achieved, respectively. Tangsuksant *et al.* (2014) presented a method for identifying ASL alphabets. They captured 2880 images, and features

of area and angle were extracted. The collected data were classified using feed-forward back propagation of the Neural Network (NN). Zamani and Kanan (2014) proposed a method for recognition of alphabets and numerals of ASL using camera. They collected 2520 one-handed static sign images and obtained an accuracy of 99.88%. Jangyodsuk *et al.* (2014) proposed an American SLRS in which they have used both camera and Kinect for capturing double-handed dynamic signs. The features of hand shape, velocity vector and motion vector were extracted using HOG. The experimental results showed that the system's accuracy improves using HOG features.

Wu *et al.* (2015) proposed an electromyography and arm sensor-based SLRS. They have collected 40 American signs and classified them using four classifiers named Nearest Neighbor, Naïve Bayes, LibSVM, and Decision tree. The results showed that the SVM outperforms all other methods of classification. Usachokcharoen *et al.* (2015) presented a method for identification of signs using Kinect and colored gloves. They extracted depth, motion, and color features from eight American signs. The color feature has been found to enhance the system's accuracy. Savur and Sahin (2015) proposed a recognition system using ASL and armbands. They collected 2080 samples of sign alphabets and classified them by using SVM. The results show that an accuracy of 82.3% was obtained in real-time. Sun *et al.* (2015) presented a LSVM modeling-based recognition system. To evaluate the developed method, 73 signs and 63 sentences in ASL were collected using a Kinect sensor, and the accuracy of 86% and 82.9% were achieved, respectively. The experiments showed that the accuracy of 96.67% was obtained by fusing the camera and data glove. Aryanie and Heryadi (2015) presented a recognition system based on a camera. They have captured 5000 samples of American alphabets in total. In this method, the color histogram was used to extract the features, and PCA was then used to condense the extracted feature set's dimensions. KNN was used to classify the gathered signs, and the best accuracy of 99.8% was attained.

Kumar *et al.* (2016) proposed an SLRS for recognizing static as well as dynamic signs in ASL. They employed Zernike moments for the identification of hand orientation in static signs, and for dynamic signs, the center of gravity of a fingertip was tracked. Savur and Sahin (2016) developed the ASL system using an armband for dynamic signs. They extracted

time and frequency domain features, and it has been observed that the system's accuracy gets enhanced by adopting the average power feature. Saha *et al.* (2016) proposed a framework for recognizing single-handed American alphabets. They employed a neural network for classification and observed that it had outperformed all other traditional networks.

AIQattan and Sepulveda (2017) proposed a NN-based ASL recognition system. They collected six single-handed dynamic signs, and discrete wavelet transform was used for extracting features. The system was classified using Linear Discriminant Analysis (LDA) and SVM, and the best 75% and 76% accuracy were achieved, respectively. Kim *et al.* (2017) used an impulse radio sensor for ASL recognition. The collected signs were classified using Convolutional Neural Network (CNN), with an average accuracy greater than 90%. Islam *et al.* (2017) presented a gesture prediction system using NN. They collected 1850 single-handed static signs using a mobile camera, and recognition rate of 94.32% was attained. Karayilan and Kilic (2017) presented an NN-based SLRS. They collected signs using the camera from which raw and histogram features were extracted. The average recognition rate of 70% and 85% was attained for raw and histogram features, respectively. Ferreira *et al.* (2017) developed a multimodal fusion SLRS. They collected 1400 single-handed static signs in total using Kinect and leap motion. The system was trained using CNN, and the best accuracy of 97% was obtained using color, depth, and leap motion data. Oyedotun and Khashman (2017) presented a hand recognition system for static signs in which they collected 2040 alphabet signs in total. The collected signs were segmented using median filtering, and the accuracy of 91.33% was achieved using CNN.

Bantupalli and Xie (2018) proposed a CNN-based SLRS. CNN is used for spatial feature recognition, and Recurrent Neural Network (RNN) trains temporal features. The experiments were performed on a dataset of size 2400 videos and obtained an accuracy of 93%. Joze *et al.* (2018) developed a real-life-based sign language dataset of 25000 videos. The proposed system was trained and tested using CNN and obtained an accuracy of 81.08%. Chong and Lee (2018) presented a leap motion controller based on American SLRS. They also extract features from fingers and hand movements which helps in the recognition of static and dynamic signs. SVM and Deep Neural Network (DNN) studies were conducted on 26

alphabets and 10 digits from ASL, with 72.79% and 88.79%, respectively. Rastgoo *et al.* (2018) developed a camera-based American SLRS using CNN. In this article, the experiments were implemented with four different datasets and achieved the highest accuracy of 99.31% on Massey University Gesture Dataset 2012. Lahoti *et al.* (2018) presented a mobile camera-based single-handed SLRS. They have implemented an android application for converting ASL to text. They have used the SVM model to classify signs and attained an accuracy of 89.54%. Taskiran *et al.* (2018) proposed a SLRS for recognizing static and dynamic signs. The experiments were performed on the already collected dataset by Massey University in 2011 and obtained an accuracy of 98.05% using CNN.

Gurbuz *et al.* (2020) presented Radio Frequency (RF) sensors for serving the deaf community. Unaffected by lighting conditions, non-invasive, non-contact measurements of ASL signing are obtained using a multi-frequency RF sensor network. The Short-Time Fourier Transform (STFT) is used in time-frequency analysis to identify the distinctive motion patterns found in the radio frequency data. The proposed method achieved an accuracy of 72% using a random forest classifier. Li *et al.* (2020) proposed a camera-based double-handed SLRS. They have implemented appearance-based and pose-based approaches and achieved a comparable performance of up to 62.63%. They also suggested pose based temporal graph convolution networks and achieved an accuracy of 67.83%.

Lee *et al.* (2021) developed a leap motion-based American SLRS. The sphere's radius, the angles between the fingers, and the separation between the finger's positions were extracted as features. They have adopted Long-Short Term Memory Recurrent Neural Network with the KNN method for classification. The experiments were performed on 26 ASL alphabets and the recognition rate of 99.44% was attained. Wen *et al.* (2021) proposed a SLRS using triboelectric smart gloves. They implemented the CNN model on 50 words and 20 sentences from ASL and achieved an average rate of 86.67% on unseen sentences.

The summarized literature review of American SLRS is represented in Table 2.2.

Table 2.2: A summarized review of American SLRS

Author(s)	Acquisition Mode	Single/Double Handed	Static/ Dynamic	Signing Mode	Technique Used	Recognition Rate
Munib *et al.*(2007) [121]	Camera	Both	Static	Isolated	Neural Networks	92.33%
Oz and Leu (2007) [128]	Gloves	Single-handed	Dynamic	Isolated	Neural Networks	95%
Oz and Leu(2011) [129]	Gloves	Single	Static	Isolate	Neural Networks	90%
Ragab *et al.* (2013) [140]	Camera	Single	Static	Isolated	SVM and Random Forest	94%
Sun *et al.* (2013a) [175]	Kinect	Both	Dynamic	Continuous	Latent Support vector machine	86%
Sun *et al.* (2013b) [176]	Kinect	Both	Dynamic	Isolated	Adaboost	86.8%
Chuan *et al.* (2014) [33]	Leap motion sensor	Single	Static	Isolated	KNN and SVM	72.78% (KNN) 79.83% (SVM)
Tangsuksant *et al.* (2014) [178]	Camera	Single	Static	Isolated	NN	95%
Zamani and Kanan (2014)[205]	Camera	Single	Static	Isolated	NN	99.88%
Jangyodsuk *et al.* (2014) [67]	DB1:Camera; DB2: Kinect	Double	Both	Isolated	DTW	DB1: 93.38%, DB2: 92.54%
Wu *et al.* (2015) [194]	Arm sensors	Single	Dynamic	Isolated	Decision tree, SVM, NN, Naïve Bayes	81.88, 99.09, 98.56, 84.11%
Usachokcharoen *et al.* (2015) [188]	Kinect	Single	Dynamic	Isolated	SVM	95%
Savur and Sahin (2015) [160]	Armband	Single	Both	Isolated	SVM	82.3% (real-time system)
Sun *et al.* (2015) [177]	Kinect	Both	Both	Both	Latent SVM	86% (words), 82.9% (sentences)
Aryanie and Heryadi (2015) [20]	Camera	Single	Static	Isolated	KNN	99.8% for k=3 best

Author(s)	Acquisition Mode	Single/Double Handed	Static/ Dynamic	Signing Mode	Technique Used	Recognition Rate
Kumar *et al.* (2016)[86]	Camera	Single	Both	Isolated	SVM	93% (static), 100% (dynamic)
Savur and Sahin (2016) [161]	Armband	Single	Dynamic	Both	SVM and ensemble learner	Average power: 60.85%
Saha *et al.* (2016) [154]	Camera	Single	Static	Isolated	Neural network	>90%
AIQattan and Sepulveda (2017)[14]	Electroence phalogram	Single	Dynamic	Isolated	LDA and SVM	LDA: 75%; SVM: 76%
Kim *et al.* (2017) [81]	Impulse radio sensor	Single	Static	Isolated	CNN	>90%
Islam *et al.* (2017) [66]	Camera	Single	Static	Isolated	ANN	94.32%
Karayilan and Kilic (2017) [74]	Camera	Single	Static	Isolated	NN	85% (histogram features)
Ferreira *et al.* (2017) [46]	Kinect	Single	Static	Isolated	CNN	97%
Oyedotun and Khashman (2017)[127]	Camera	Single	Static	Isolated	CNN	91.33%
Bantupalli and Xie (2018) [26]	Camera	Double	Dynamic	Continuous	CNN and RNN	93%
Joze *et* al. (2018) [71]	Camera	Double	Dynamic	Continuous	CNN	81.08%
Chong and Lee (2018) [32]	Leap Motion	Single	Both	Isolated	SVM and DNN	72.79% and 88.79%
Rastgoo *et* al. (2018) [143]	Camera	Single	Both	Isolated	CNN	99.31%
Lahoti *et* al. (2018) [92]	Camera	Single	Static	isolated	SVM	89.54%
Taskiran *et* al. (2018) [179]	Camera	Single	Both	Isolated	CNN	98.05%
Gurbuz *et* al. (2020) [54]	RF Sensor	Both	Both	Both	Random Forest	72%

Author(s)	Acquisition Mode	Single/Double Handed	Static/ Dynamic	Signing Mode	Technique Used	Recognition Rate
Li *et* al. (2020) [98]	Camera	Double	Both	Isolated	CNN	67.83%
Lee *et.al* (2021) [95]	Leap Motion	Single	Both	Isolated	RNN and KNN	99.44%
Wen *et al.* (2021) [192]	Gloves	Both	Both	Both	CNN	86.67%

2.2.1.2 Discussions

The objective of this study is to explore the present literature as per the research questions mentioned in Section 2.1. The results of the review regarding research questions are addressed below.

To address RQ1, *i.e.,* "Which data acquisition devices have been used mostly for capturing signs in SLRSs?" data has been analyzed to plot graph as shown in Figure 2.5 (a).

It has been observed that on American sign language, 48% of the research work has been done using cameras, followed by 16% using Kinect, 9% using armband, 9% using gloves, 9% using leap motion, and 3% using electroencephalogram, radio frequency sensors and impulse radio sensor as shown in Figure 2.5 (a).

To address RQ2, *i.e.,* "How much research is being carried out on static/dynamic signs in SLRSs?" data has been analyzed to plot graph as shown in Figure 2.5 (b).

It has been found that the majority of research in ASL has been accomplished using static signs (41%), which is further followed by dynamic signs (27%), while 32% of the work has been done on both static and dynamic signs as shown in Figure 2.5 (b).

To address RQ3, *i.e.,* "What various signing techniques are taken into account when using sign language?" data has been analyzed to plot the graph as shown in Figure 2.5 (c).

It has been found that the majority of work has been done on isolated signs (79%), followed by continuous signs (9%), while 12% on both isolated and continuous signs, as shown in Figure 2.5 (c).

To address RQ4, *i.e.,* "How to classify and identify single and double-handed signs for SLRS?" data has been analyzed to plot a graph, as shown in Figure 2.5 (d).

It has been seen that the maximum amount of work has been done on single-handed signs (68%) in ASL, 12% on double-handed signs, and 20% on both single and double-handed signs, as presented in Figure 2.5 (d).

To address RQ5, *i.e.*, "What are the existing methodologies and techniques available to recognize sign language recognition?" data has been analyzed to plot a graph, as shown in Figure 2.5 (e).

Figure 2.5 (e) depicts that a significant amount of work on ASL has been implemented using neural networks (23%), followed by SVM (21%), hybrid techniques (15%), and CNN (26%). In contrast, the least amount of work has been performed using AdaBoost, KNN, random forest, and DTW techniques.

To address RQ6, *i.e.*, "What is the accuracy and coverage of existing SLRSs?" data has been analyzed to plot graph as shown in Figure 2.5 (f).

It has been observed that 57% of SLRS achieved an average accuracy of greater than 90%, while 27% of the systems have an accuracy between 80-89%. Only 16% of systems have less than 80% accuracy, as represented in Figure 2.5 (f).

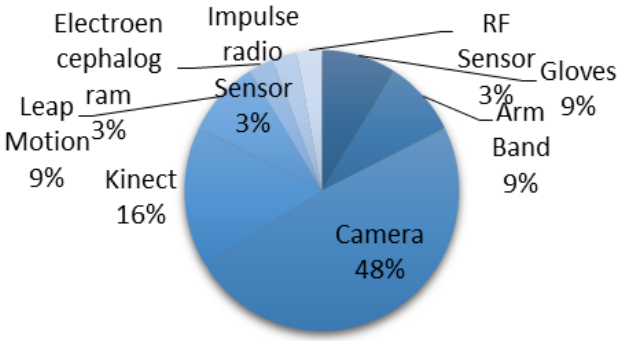


Figure 2.5 (a): Usage of different data acquisition techniques used in ASL systems

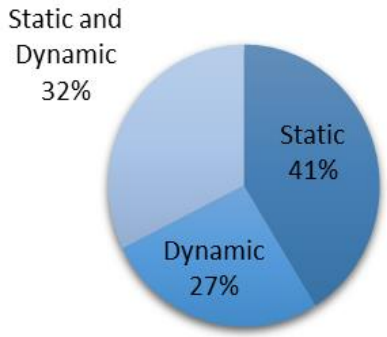


Figure 2.5 (b): Research work carried out on static/dynamic signs in ASL

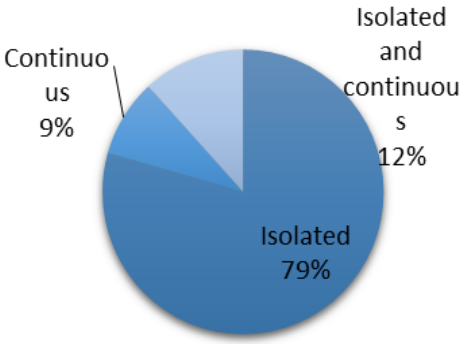


Figure 2.5 (c): Percentage of research work carried out based on signing mode in ASL

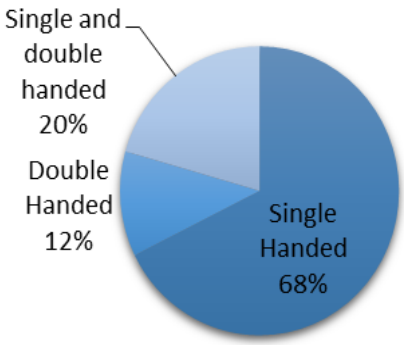


Figure 2.5 (d): Percentage of research work carried out based on single/double-handed signs in ASL

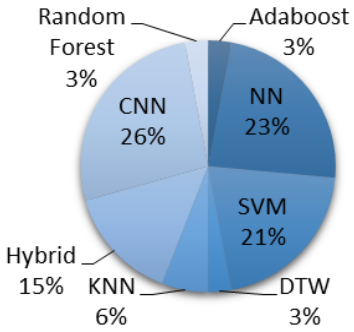


Figure 2.5 (e): Percentage of research work carried out on technique used for recognition of signs

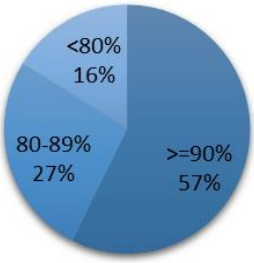


Figure 2.5 (f): Accuracy of research for different ASL systems

2.2.2 Indian Sign Language

Indian Sign Language (ISL) is a predominant sign language used in the areas of South Asia. The estimate of ISL users is 2,700,000 [97]. ISL has three regional dialects: Mumbai-Delhi Sign Language, Punjab-Sindh Sign Language, and Bangalore-Chennai-Hyderabad Sign Language.

Sign language recognition approaches for ISL, which have been noted, are given in the following subsection.

2.2.2.1 Indian Sign Language Recognition Techniques

Rekha *et al.* (2011) presented an ISL recognition system for double-handed signs. In this system, the authors collected a dataset from 26 signs, of which 23 are static and 3 are dynamic. All the static signs were classified using SVM, and the dynamic signs were

classified using DTW. Agrawal *et al.* (2012) presented a sign language system for recognizing double-handed signs. They captured 235 images of 36 signs in total using the camera. The results showed that the fusion of shape descriptors, HOG, and SIFT feature extraction methods enhances the system's performance.

Adithya *et al.* (2013) proposed a method for recognizing single and double-handed signs. They collected 720 Indian signs using the camera, consisting of alphabets and numbers. The collected signs were segmented using skin color and classified using ANN. Rahaman *et al.* (2014) proposed a SLRS using camera in real-time. The dataset contains 7200 double-handed Bengali signs, including signs of 6 vowels and 30 consonants. Features of finger position and fingertip were extracted, and signs were classified using KNN. The disadvantage of the system is that it cannot accurately segment the hand area if objects with skin color appear.

Mehrotra *et al.* (2015) presented a system for recognizing double-handed Indian signs. Microsoft Kinect captured 37 signs in this system, and the features from skeleton joints were extracted. The system was classified with multiclass SVM, and an accuracy of 86.16% was obtained. Tripathi *et al.* (2015) developed a system for recognizing sentences in ISL. This system captured 500 samples from 10 sentences, including single-handed and double-handed dynamic samples. In the feature extraction phase, key frame extraction was used, which helps in reducing the training and testing time. The signs were classified using Hidden Markov Model (HMM), and overall accuracy of 91% was achieved. Yasir *et al.* (2015) presented a Scale Invariant Feature Transform (SIFT) based approach for recognizing static double-handed Bangla signs. The system was trained with 150 samples of alphabets and words, and the features of gradient magnitude and orientation were extracted for classification using SVM.

Kishore *et al.* (2016) proposed a system for recognizing ISL sentences. The dataset consisted of 580 sentences, and the features named hand shape ad optic flow hand tracking were extracted. According to the testing findings, a recognition rate of 90.17 % was attained. Using a Leap Motion Controller (LMC), Naglot and Kulkarni (2016) presented an ANN-based ISL for number recognition. They classified single-handed dynamic signs with Multilayer Perceptron (MLP) and attained 100% accuracy. Hasan *et al.* (2016) developed a

machine learning-based system for detecting Bangla Sign Language. They captured 16 static signs using the web camera and collected 320 samples. The gradient direction at each pixel and magnitude were extracted, and the accuracy of 86.53% was achieved. Kumar e*t al.* (2016) presented a continuous SLRS in which they used the mobile's front camera for collecting signs. They extracted head and hand contour energies from the collected signs and obtained an accuracy of 90%. Ahmed *et al.* (2016) presented a vision-based hand gesture recognition approach that recognizes double-handed dynamic signs. They captured 24 isolated signs using the web camera, and features from hands and faces were extracted using trajectory tracking of the moving hand. The experiments represents that an accuracy of 90% was achieved. Uddin and Chowdhury (2016) developed a camera-based Bangla SLRS for recognizing double-handed static signs. The dataset consisted of 4800 samples, and the Gabor filter was used for extracting features.

Kumar *et al.* (2017a) presented a framework for a sensor-based recognition system for sign language. They used Kinect and a leap motion device to collect 7500 total samples from 50 sign words. The fingertip position features and direction were extracted using the leap motion API. The system was classified using HMM and Bidirectional Long Short-Term Memory Neural Network (BLSTM-NN). The results showed that the overall accuracy of 95.60% and 84.57% were obtained, respectively. Rao *et al.* (2017) proposed a continuous SLRS in which they have used the mobile front camera for capturing signs. Kumar *et al.* (2017b) proposed a coupled HMM-based SLRS. They collected single-handed dynamic signs from 25 words using Kinect and leap motion. Rao and Kishore (2017) presented a continuous Indian SLRS in which they made use of selfie videos. They performed filtering, segmentation, and contour detection with Gaussian filtering on the database of 18 signs. The collected sign was classified using Minimum Distance Classifier (MDC) and ANN and attained an accuracy of 85.58% and 90%, respectively. Kumar *et al.* (2017c) developed a sign language identification system in real-time. Using a leap motion sensor, they collected single-handed static signs from 56 sign samples. The experiments were performed using SVM and BLSTM-NN, and an accuracy of 63.57% was obtained. Kumar *et al.* (2017d) presented a framework for position

and rotation invariants to identify sign language using Kinect. They collected 2700 gestures in total from 30 signs, and an accuracy of 83.77% was obtained using HMM.

Shenoy *et al.* (2018) presented a hand pose and gesture recognition system for identifying Indian signs. The dataset contains 24,624 images in total, collected by using 33 hand poses and 12 gestures. Hand poses were classified using KNN and obtained an accuracy of 99.7%. On the other hand, HMM was used for categorizing gestures, and an accuracy of 97.23% was obtained. Hossen *et al.* (2018) developed a Deep Convolutional Neural Networks (DCNN) method for recognizing Bengali sign language. The system is developed to acknowledge 37 Bengali alphabets. The experiments were performed on 1147 images and achieved an accuracy of 96.33% using training dataset and 84.68% using validation dataset. Sajanraj and Beena (2018) presented a CNN-based ISL recognition system for identifying numbers from 0-9. The developed system has been trained with 3000 static symbols and achieved an accuracy of 99.56%. Kishore *et al.* (2018) developed a selfie-based system to recognize 200 ISL signs. They designed and tested different CNN architectures and achieved an average accuracy of 92.88%. Rao and Kishore (2018) presented an Adaboost multilabel multiclass learning algorithm for recognizing signs. They used extraction using key frame, face detection, identification of hand search space, extraction of head-hand portion, and segmentation of hand-head shape for multiple feature extraction. The system was tested for 10 continuous ISL sentences formed from 282 words and achieved an accuracy of 90%.

Mariappan and Gomathi (2019) presented a camera-based real-time ISL recognition system. They used the segmentation feature for identifying and tracking the Region of Interest (ROI) in sign language recognition. The developed system is trained using fuzzy c-means clustering and produced an accuracy of 75% for recognizing words. Sruthi and Lijiya (2019) presented a signer independent deep learning-based SLRS. They made use of a dataset that was supplemented with 5157 static images gathered from 24 alphabets. 4125 images were randomly picked to utilize in training, resulting in a training accuracy of 99.93%. With 1032 images used in validation and testing, the largest validation accuracy of 98.64% was attained.

Mittal *et al.* (2019) proposed a modified LSTM model for the recognition of sequence of gestures. 942 signed ISL sentences were used to test the proposed method. The accuracy for

signed sentences and isolated words was 72.3% and 89.5%, respectively. Abraham *et al.* (2019) proposed an approach for interpreting static and dynamic signs in ISL. They have used a sensor glove mounted with flex sensors for detecting hand fingers. The collected signs were classified using LSTM networks and attained an accuracy of 98%. Athira *et al.* (2019) presented a gesture recognition system for recognition of signs from live video. The collected data were preprocessed using skin color segmentation and classified using SVM. The proposed system attained an accuracy of 91% on alphabets and 89% on single-handed dynamic words. Bhagat *et al.* (2019) presented a hand gesture recognition system using Microsoft Kinect in real-time. They worked upon both static and dynamic signs from ISL and collected a dataset of 45,000 images. The proposed model was trained using CNN and attained an accuracy of 98.81% and 99.08% on static and dynamic signs, respectively. Suri and Gupta (2019) introduced a novel Inertial Measurement Unit (IMU)-based one-dimensional deep capsule network (CapsNet) architecture for continuous ISL detection. In comparison to CNN, which produces an accuracy of 87.99%, it has been noted that CapsNet achieved a better accuracy of 94%.

Gangrade *et al.* (2020) presented a SLRS in which Microfost Kinect sensor has been used for hand segmentation. The features were extracted using Oriented FAST and Rotated BRIEF (ORB), and the machine learning algorithm KNN was applied to images from 0-9. The developed system attained an accuracy of 93.26% on the ISL dataset. Raghuveera *et al.* (2020) presented a SLRS using Microsoft Kinect. The dataset of RGB and depth images of 140 gestures were used. It considers both single-handed and double-handed signs. The developed system was trained using SVM and obtained an accuracy of 71.85%. Wadhawan and Kumar (2020) proposed a CNN-based SLRS for simple static signs. They collected 35,000 sign images in total from 100 static signs. The results were evaluated based on various optimizers, and it has been found that the SGD optimizer achieved a higher accuracy of 99.72% on colored images and 99.90% on grayscale images.

Sharma *et al.* (2021) presented the comparison between CNN model and machine learning models. Three models were examined based on various trainable parameters: a pre-trained VGG16 with fine-tuning, a VGG16 with transfer learning, and a hierarchical neural network.

According to the results, the hierarchical model gave better results than the other two models, with the best accuracy of 98.52% for one-hand gestures and 97% for two-hand gestures. Shamrat *et al.* (2021) developed a CNN-based Bangla SLRS. The dataset consists of 310 sign images collected from 10 different signs. The collected dataset was preprocessed using bandlet transformation after this logarithm replacement had been applied to control the extra light effects on images.

The summarized literature review of ISL recognition systems is shown in Table 2.3.

Table 2.3: A summarized review of ISL recognition systems

Author(s)	**Acquisition Mode**	**Single/Double Handed**	**Static/ Dynamic**	**Signing Mode**	**Technique Used**	**Recognition Rate**
Rekha *et al.* (2011) [151]	Camera	Double	Both	Isolated	SVM (static) and DTW (dynamic)	86.3% (SVM), 77.2% (DTW)
Agrawal *et al.* (2012) [8]	Camera	Double	Static	Isolated	Multiclass SVM	93%
Adithya *et al.* (2013) [5]	Camera	Both	Static	Isolated	ANN	91.11%
Rahaman *et al.* (2014) [142]	Camera	Double	Static	Isolated	KNN	98.17% (vowels) and 94.75% (consonants)
Mehrotra *et al.* (2015) [111]	Kinect	Double	Both	Isolated	Multiclass SVM	86.16%
Tripathi *et al.* (2015) [184]	Camera	Both	Dynamic	Continuous	HMM	91%
Yasir *et al.* (2015) [201]	Camera	Double	Static	Isolated	SVM	86%
Kishore *et al.* (2016) [82]	Camera	Both	Dynamic	Continuous	ANN	90.17%
Naglot and Kulkarni (2016) [122]	Leap motion	Single	Dynamic	Isolated	ANN	100.00%
Hasan *et al.* (2016) [57]	Web camera	Both	Static	Isolated	SVM	86.53%
Kumar et *al.* (2016) [87]	Camera	Single	Dynamic	Continuous	ANN	90.00%
Ahmed *et al.* (2016) [11]	Web camera	Double	Dynamic	Isolated	DTW	90.00%

Author(s)	Acquisition Mode	Single/Double Handed	Static/ Dynamic	Signing Mode	Technique Used	Recognition Rate
Uddin and Chowdhury (2016) [186]	Camera	Double	Static	Isolated	SVM	97.70%
Kumar *et al.* (2017a) [88]	Kinect and leap motion	Both	Dynamic	Isolated	HMM and BLSTM-NN	95.60% (all signs)
Rao *et al.* (2017) [149]	Camera	Single	Dynamic	Continuous	ANN	91%
Kumar *et al.* (2017b) [89]	Kinect and Leap motion	Single	Dynamic	Isolated	Coupled HMM	90.80%
Rao and Kishore (2017) [147]	Camera	Single	Static	Continuous	ANN	90.00%
Kumar *et al.* (2017c) [90]	Leap motion	Single	Static	Isolated	SVM and BLSTM-NN	63.57%
Kumar *et al.* (2017d) [91]	Kinect	Both	Static	Isolated	HMM	83.77%
Shenoy *et al.* (2018) [166]	Camera	Both	Both	Isolated	KNN (Hand Pose) and HMM (Gestures)	99.7%(Hand Pose), 97.23%(Gesture)
Hossen *et al.* (2018) [62]	Camera	Single	Static	Isolated	CNN	84.68%
Sajanraj and Beena (2018) [155]	Camera	Single	Static	Isolated	CNN	99.56%
Kishore *et al.* (2018) [83]	Camera	Both	Both	Isolated	CNN	92.88%
Rao and Kishore (2018) [148]	Camera	Both	Both	Continuous	Adaboost	90%
Mariappan and Gomathi (2019) [107]	Camera	Both	Both	Both	Fuzzy c-mean clustering	75% (words)
Sruthi and Lijiya (2019) [173]	Camera	Both	Static	Isolated	CNN	98.64%
Mittal *et al.* (2019) [113]	Leap Motion	Both	Both	Both	LSTM Network	72.3% (sentences) and 89.5%(words)

Author(s)	Acquisition Mode	Single/Double Handed	Static/ Dynamic	Signing Mode	Technique Used	Recognition Rate
Abraham *et al.* (2019) [2]	Gloves	Both	Both	Isolated	LSTM Network	98%
Athira *et al.* (2019) [22]	Camera	Both	Both	Isolated	SVM	91% (alphabets), 89% (single-handed dynamic words)
Bhagat *et al.* (2019) [28]	Kinect	Both	Both	Isolated	CNN	98.81% (static words), 99.08% (dynamic words)
Suri and Gupta (2019) [174]	IMU sensor	Both	Both	Continuous	Deep Capsule Network and CNN	94% (CapsNet), 87.99% (CNN)
Gangrade *et al.* (2020)[48]	Kinect	Single	Static	Isolated	KNN	93.26%
Raghuveera *et al.* (2020) [141]	Kinect	Both	Static	Both	SVM	71.85%
Wadhawan and Kumar (2020) [190]	Camera	Both	Static	Isolated	CNN	99.90%
Sharma *et al.* (2021) [165]	Camera	Both	Static	Isolated	SVM	98.52% (Single handed), 97% (Double handed)
Shamrat *et al.* (2021) [163]	Camera	Single	Static	Isolated	CNN	99.80%

2.2.2.2 Discussions

The results of the review in ISL regarding research questions are addressed below.

To address RQ1, *i.e.*, "Which data acquisition devices have been used mostly for capturing signs in SLRSs?" data has been analyzed to plot graph as represented in Figure 2.6 (a).

It has been noticed that 67% of the work on ISL has been conducted using cameras, followed by 14% using Kinect, 8% using leap motion, 5% using both Kinect and leap motion, and 3% using Gloves and IMU sensor each as represented in Figure 2.6 (a).

To address RQ2, *i.e.,* "How much research is being carried out on static/dynamic signs in SLRSs?" data has been analyzed to plot graph as represented in Figure 2.6 (b).

Figure 2.6 (b) depicts that the majority of research in ISL has been performed on static signs (47%), followed by dynamic signs (22%) and both static and dynamic signs (31%).

To address RQ3, *i.e.,* "What signing techniques are taken into account when using sign language?" data has been analyzed to plot the graph as represented in Figure 2.6 (c).

It has been seen that the majority of research has been conducted on isolated signs (72%), followed by continuous signs (20%) and both isolated and continuous signs (8%), as shown in Figure 2.6 (c).

To address RQ4, *i.e.,* "How to classify and identify single and double-handed signs for SLRS?" data has been analyzed to plot graph as represented in Figure 2.6 (d).

Figure 2.6 (d) depicts that 53% of work in ISL has been carried out on single and double-handed signs, followed by 28% and 19% on single and double-handed signs, respectively.

To address RQ5, *i.e.,* "What are the existing methodologies and techniques available to recognize sign language recognition?" data has been analyzed to plot a graph, as represented in Figure 2.6 (e).

Figure 2.6 (e) depicts that the significant work on ISL has been implemented using SVM (22%), followed by Neural Networks (17%), CNN (19%), hybrid techniques (17%), HMM (8%), LSTM(6%), KNN(5%). In contrast, the least amount of work has been performed using DTW and fuzzy c-mean clustering.

To address RQ6, *i.e.,* "What is the accuracy and coverage of existing SLRSs?" data has been analyzed to plot graph as shown in Figure 2.6 (f).

It has been observed that for ISL there are 70% of SLRSs achieved an average accuracy of greater than 90%, while 19% of the systems have an accuracy between 80-89%. Only 11% of systems have less than 80% accuracy, as represented in Figure 2.6 (f).

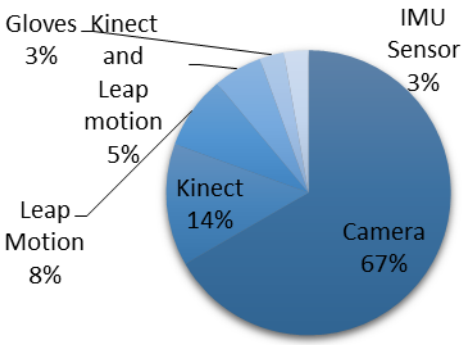


Figure 2.6 (a): Usage of different data acquisition techniques used in ISL systems

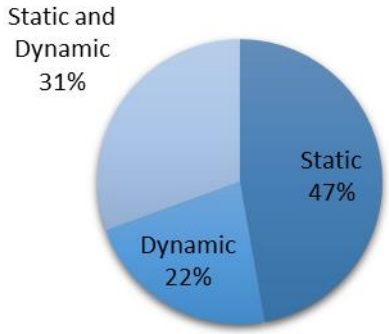


Figure 2.6 (b): Research work carried out on static/dynamic signs in ISL

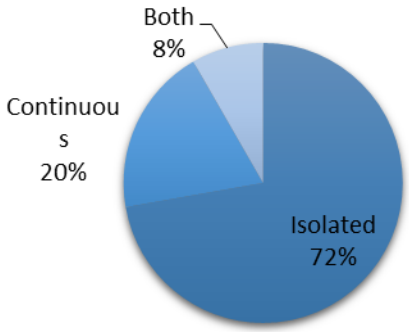


Figure 2.6 (c): Percentage of research work carried out based on signing mode in ISL

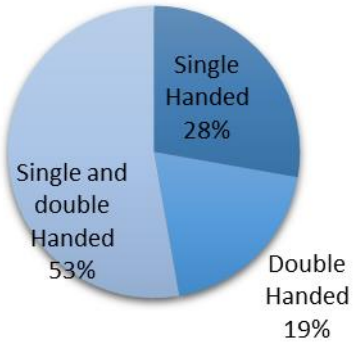


Figure 2.6 (d): Percentage of research work carried out based on single/double-handed signs in ISL

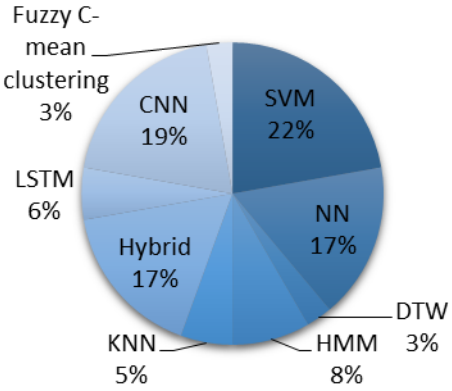


Figure 2.6 (e): Percentage of research work carried out on technique used for recognition of signs

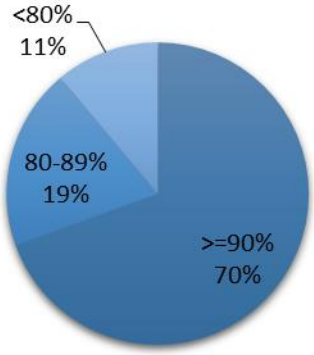


Figure 2.6 (f): Accuracy of research for different ISL systems

2.2.3 Arabic Sign Language

Arabic Sign Language (ArSL) is the language that is distributed across Mideast and North Africa regions. Sign language recognition approaches for ArSL, reported in the last years, are given below.

2.2.3.1 Arabic Sign Language Recognition Techniques

Mohandas and Deriche (2007) developed an ArSL recognition system in which they collected 4500 dynamic samples. They have used the region growing technique for extracting different features. The system is classified using 5-state HMM, and the highest accuracy of 97.3% was achieved with an equal number of training and testing samples.

Maraqa and Abu-Zaiter (2008) proposed a camera-based ArSL recognition system for recognizing single-handed static words. The collected dataset consisted of 1200 images, which were classified using a RNN, and an accuracy of 95.11% was achieved. Assaleh *et al.* (2008) proposed a camera-based user-dependent continuous SLRS. This system gathered dynamic words and sentences, and Discrete Cosine Transform (DCT) and zonal coding feature extraction were employed. The system was classified using 9-state HMM, and the accuracy of 75% for sentence recognition and 94% for recognition of words was achieved.

AL Rousan *et al.* (2009) proposed a camera-based isolated SLRS. They collected dynamic double-handed signs from 30 words. The location, movement, and orientation features were extracted using DCT and zonal coding. The system was classified using HMM, and the accuracy of 93.8% in signer-dependent mode and 90.6% in a signer-independent manner was achieved. Shanableh and Assaleh (2011) proposed a user-independent ArSL recognition system. They captured 3450 video segments of isolated signs using a camera and colored gloves. The collected signs were preprocessed using median filtering, and the features of bounding boxes were extracted. For further classification, KNN was applied, and an accuracy of 87% was achieved.

Mohandes *et al.* (2012) presented an ArSL system for recognizing signs. They use a camera and colored gloves to collect double-handed static signs. They have also used the region growing technique for tracking hands, and the features of centroids, the eccentricity of the bounded ellipse, and the first principal component's angle and area of both the hands were

extracted. It has been found that the system's performance increases as the number of training sequences per sign increases.

Dahmani and Larabi (2014) presented a model for identification of sign language alphabet. They used a camera and colored gloves to acquire static single-handed Arabic signs. The recognition is based on shape descriptors, Hu moments, and geometric features. The classification is performed using KNN and SVM. Elons *et al.* (2014) presented a leap motion sensor-based ArSL recognition system, in which they collected dynamic double-handed samples of isolated words. The features of finger position and distance between the fingers were extracted, and the signs were classified using multilayer perceptron neural networks. Ahmed and Aly (2014) developed an appearance-based ArSL recognition system. The dataset contains 3450 samples in total from 23 words. The texture and shape features were extracted using Local Binary Patterns (LBP) and Principal Component Analysis (PCA) and further classified using HMM. Mohandes *et al.* (2014) developed an ArSL recognition system, in which they used Leap Motion Controller (LMC) for capturing input data. They captured 6400 single-handed static signs. All the collected signs were classified using MLP neural network and Naïve Bayes classifiers. It has been observed from the results that Naïve Bayes outperforms NN.

An ArSL recognition system dependent upon user was proposed by Tubaiz et al. (2015). For gathering dynamic signs in the form of sentences, they employed gloves and a camera. Re-sampling and z-score normalization were utilized to preprocess the gathered data, and modified KNN was employed for classification. Sarhan *et al.* (2015) developed a SLRS using Kinect. They collected 215 dynamic samples from 16 Arabic words. The features of articulation point, hand orientation, hand shape, and hand movement were extracted using skeletal and depth information obtained from Kinect.

Hassan *et al.* (2016) developed a recognition system in which they used gloves and a Polhemus tracker for collecting sentences. The system was classified using the Modified KNN (MKNN) technique and HMM after the features were retrieved using a sliding window-based approach. According to the experimental findings, MKNN performs better at classifying sentences than HMM, and HMM performs better at classifying words than

MKNN. Hamed et al. (2016) proposed a recognition system based on HOG-PCA employing Kinect to recognize Arabic signs in complex backdrops. A mechanism for human-machine interaction for ArSL recognition was created by Darwish (2017). Using a webcam, they captured more than 6000 single-handed static signs. The developed system was classified using fuzzy HMM, and an accuracy of 92.4% was attained.

Abdel *et al.* (2018) introduced a translator based on DTW, where each word has been recognized and identified in the form of text. The experiments were performed using a Microsoft Kinect sensor on the set of 30 isolated words and achieved an accuracy of 97.58% and 95.25% for signer-dependent and signer-independent signs, respectively. Alzohairi *et al.* (2018) introduced an image-based SLRS. They collected the dataset from 30 Arabic alphabets and classified them using SVM. The experiments were performed using different descriptors, and it has been observed that the HOG descriptor outperforms other descriptors.

Zakariya and Jindal (2019) presented a smartphone-based ArSL recognition system. They collected 2000 images of 10 alphabets, classified them using SVM, and obtained an accuracy of 92.5%. Deriche *et al.* (2019) developed a leap motion-based ArSL recognition system. They collected 100 isolated dynamic signs, applied Gaussian Mixture Model (GMM) with two LMC, and attained an accuracy of 91.83%. Hayani *et al.* (2019) proposed a CNN-based SLRS. They collected 2030 images of numbers and 5839 images of letters of Arabic signs. It has been observed from the experimental results that the proposed system performed better as compared to the traditional KNN and SVM models.

Almasre and Al-Nuaim (2020) proposed a system using a Kinect sensor dynamically for the recognition of Arabic gestures. Using eleven prediction models from three algorithms (SVM, RF, and KNN) with various parameter settings, they collected gestures of dynamic words of Arce. The results of the studies demonstrated that SVM models with linear kernels produced the highest recognition accuracy rates for dynamic gestures. Aly *et al.* (2020) represented a framework for signer-independent sign language recognition using a Bi-directional LSTM (BiLSTM) network. The experiments were performed on 23 dynamic words and achieved an accuracy of 89.59% on grayscale images. Saleh and Issa (2020) presented a model based on CNN for sign language recognition. They collected 25,600 images from 32 Arabic gestures.

The experiments were performed on VGG16 and Resnet152 models of CNN and obtained an accuracy of 99.26% and 99.57%, respectively. Kamruzzaman (2020) demonstrated a vision-based system for identifying 31 letters based on Arabic hand signs. In this, the CNN model has been employed to recognize single-handed letters and translate them into speech, resulting in an accuracy of 90%. A CNN-based SLRS was introduced by Latif *et al.* (2020) and was trained and tested on 54,000 sign images from randomly selected participants. The experiments were performed on 32 Arabic signs and achieved an accuracy of 97.6%.

Luqman and EI-Alfy (2021) proposed a SLRS for both manual and non-manual signs. They collected 6748 videos from 50 signs and obtained the best results using MobileNET LSTM with transfer learning and fine-tuning. Hisham and Himouda (2021) introduced a leap motion controller-based sign recognition system. The proposed system has been implemented using SVM, KNN, and afterward, Adaboost has been applied to enhance the recognition rate. The results showed 92.3% and 93% accuracy using Adaboost on single-handed and double-handed signs, respectively.

The summarized literature review of ArSL recognition systems is shown in Table 2.4.

Table 2.4: A summarized review of ArSL recognition systems

Author(s)	**Acquisition Mode**	**Single/Double Handed**	**Static/ Dynamic**	**Signing Mode**	**Technique Used**	**Recognition Rate**
Mohandas and Deriche (2007) [118]	Camera	Both	Dynamic	Isolated	HMM	97.3 %
Maraqa and Abu-Zaiter (2008) [106]	Camera	Single	Static	Isolated	Recurrent Neural Network	95.11%
Assaleh *et al.* (2008) [21]	Camera	Both	Dynamic	Both	HMM	75% (Sentence), 94% (Word)
AL Rousan *et al.* (2009) [15]	Camera	Double	Dynamic	Isolated	HMM	93.8%(Signer dependent), 90.6%(Signer-independent)

Author(s)	Acquisition Mode	Single/Double Handed	Static/ Dynamic	Signing Mode	Technique Used	Recognition Rate
Shanableh and Assaleh (2011) [164]	Camera	Both	Dynamic	Isolated	KNN	87%
Mohandes *et al.* (2012) [117]	Camera	Double	Static	Isolated	HMM	95.2% (Signer dependent), 94.4% (signer independent mode)
Dahmani and Larabi (2014) [34]	Camera	Single	Static	Isolated	KNN and SVM	DB1: 88.87%, DB2: 96.88%
Elons *et al.* (2014) [44]	Leap Motion Sensor	Double	Dynamic	Isolated	Multilayer perceptron Neural networks	88%
Ahmed and Aly (2014) [9]	Camera	Both	Static	Isolated	HMM	99.97%
Mohandes *et al.* (2014) [116]	Leap motion	Single	Static	Isolated	MLP Neural Networks and Naïve Bayes	98%, and >99%
Tubaiz *et al.* (2015) [185]	Gloves	Both	Dynamic	Continuous	Modified KNN	98.90%
Sarhan *et al.* (2015) [158]	Kinect	Both	Dynamic	Isolated	HMM	80.47%, 64.61% (Signer independent)

Author(s)	Acquisition Mode	Single/Double Handed	Static/ Dynamic	Signing Mode	Technique Used	Recognition Rate
Hassan *et al.* (2016) [56]	DB1:gloves , DB2:Polhemus tracker	Both	Dynamic	Continuous	HMM and Modified KNN	DB1:97%(word), 86% (sentence); DB2:97%(word), 85% (sentence)
Hamed *et al.* (2016) [55]	Kinect	Single	Static	Isolated	SVM	99.2%
Darwish (2017) [35]	Camera	Single	Static	Isolated	Fuzzy HMM	92.40%
Abdel *et al.* (2018) [1]	Kinect	Both	Both	Isolated	DTW	97.58% (signer dependent),95.25(signer independent)
Alzohairi *et al.* (2018) [17]	Camera	Single	Static	Isolated	SVM	63.5%
Zakariya and Jindal (2019) [204]	Camera	Single	Static	Isolated	SVM	92.50%
Deriche *et al.* (2019) [39]	Leap Motion	Both	Dynamic	Isolated	GMM and LDA	91.83% and 89.62%
Hayani *et al.* (2019) [58]	Camera	Single	Static	Isolated	CNN	90.02%
Almasre and Al-Nuaim (2020) [13]	Kinect	Both	Dynamic	Isolated	SVM	83%
Aly and Aly (2020) [16]	Camera	Both	Dynamic	Isolated	BiLSTM	89.59%
Saleh and Issa (2020) [157]	Camera	Single	Static	Isolated	CNN	99.26% and 99.57%

Author(s)	Acquisition Mode	Single/Double Handed	Static/ Dynamic	Signing Mode	Technique Used	Recognition Rate
Kamruzzaman (2020) [72]	Camera	Single	Static	Isolated	CNN	90%
Latif *et al.* (2020) [94]	Camera	Single	Static	Isolated	CNN	97.6%
Luqman and El-Alfy (2021) [102]	Kinect	Both	Both	Isolated	CNN	99.7(signer dependent) 72.4(signer independent)
Hisham and Hamouda (2021) [60]	Leap Motion	Both	Both	Isolated	AdaBoost	92.3(single-handed), 93(double-handed)

2.2.3.2 Discussions

The results of the review in ArSL regarding research questions are addressed below.

To address RQ1, *i.e.*, "Which data acquisition devices have been used mostly for capturing signs in SLRSs?" data has been analyzed to plot graph as represented in Figure 2.7 (a).

It has been found that 59% of the research on ArSL has been done using cameras, 18% using Kinect, 15% using leap motion, 6% using gloves, and/ 2% using Polhemus tracker, as shown in Figure 2.7 (a).

To address RQ2, *i.e.*, "How much research is being carried out on static/dynamic signs in SLRSs?" data has been analyzed to plot graph as presented in Figure 2.7 (b).

Figure 2.7 (b) depicts that the majority of research in ArSL has been performed on static signs (48%), along with dynamic signs (41%) and 11% for both static and dynamic signs.

To address RQ3, *i.e.*, "What various signing techniques are taken into account when using sign language?" data has been analyzed to plot the graph as shown in Figure 2.7 (c).

It has been observed from Figure 2.7 (c) that the majority of work has been performed on isolated signs (89%), followed by continuous signs (7%) and isolated and continuous signs (4%) in ArSL.

To address RQ4, *i.e.,* "How to classify and identify single and double-handed signs for SLRS?" data has been analyzed to plot a graph, as represented in Figure 2.7 (d).

Figure 2.7 (d) demonstrates that 41% of work in ArSL has been carried out on single-handed signs, followed by 11% on double-handed signs and 48% on both single and double-handed signs.

To address RQ5, *i.e.,* "What are the existing methodologies and techniques available to recognize sign language recognition?" data has been analyzed to plot a graph, as shown in Figure 2.7 (e).

Figure 2.7 (e) depicts that the significant work on ArSL has been implemented using HMM and CNN (26%), followed by hybrid techniques and SVM (15%), while the minimum amount of work has been performed using NN, KNN, GMM, and LDA.

To address RQ6, *i.e.,* "What is the accuracy and coverage of existing SLRSs?" data has been analyzed to plot graph as shown in Figure 2.7 (f).

It has been observed that for ArSL there are 70% of SLRSs achieved an average accuracy of greater than 90%, while 23% of the systems have an accuracy between 80-89%. Only 7% of systems have less than 80% accuracy, as represented in Figure 2.7 (f).

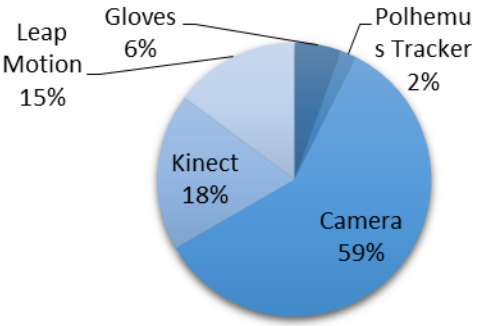


Figure 2.7 (a): Usage of different data acquisition techniques used in ArSL systems

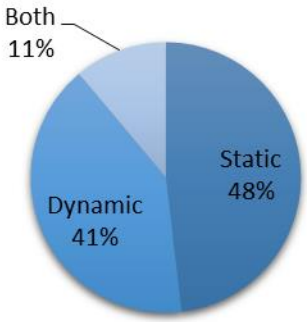


Figure 2.7 (b): Research work carried out on static/dynamic signs in ArSL

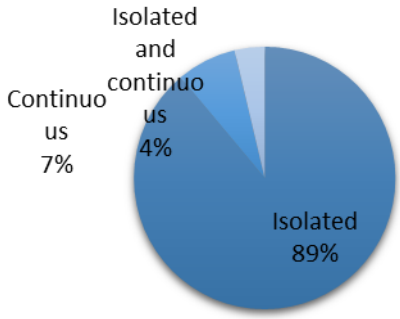


Figure 2.7 (c): Percentage of research work carried out based on signing mode in ArSL

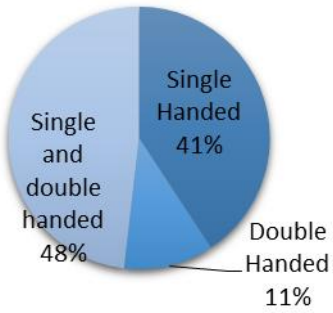


Figure 2.7 (d): Percentage of research work carried out based on single/double-handed signs in ArSL

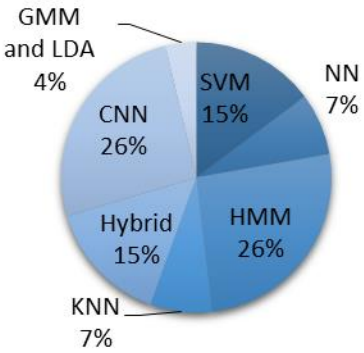


Figure 2.7 (e): Percentage of research work carried out on technique used for recognition of signs

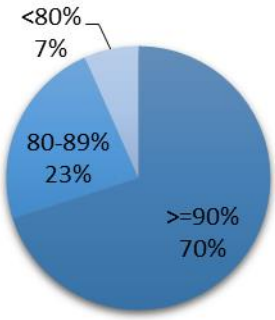


Figure 2.7 (f): Accuracy of research for different ArSL systems

2.2.4 Chinese Sign Language

Chinese Sign Language (CSL) is a unique language only utilized in some parts of China. Taiwan and Malaysian speakers also use it. According to estimates, there are 1 to 20 million CSL users [31]. The strategies for CSL recognition that have been reported in recent years are listed below.

2.2.4.1 Chinese Sign Language Recognition Techniques

Wang *et al.* (2008) developed a multilayer architecture for CSL recognition. They collected dynamic double-handed signs using cyber gloves and Polhemus 3 space position tracker. The features of hand shapes were extracted using cyber gloves, and features of orientation, position, and movement trajectory were extracted using a position tracker. The data classification was performed by first employing DTW and then HMM, and an accuracy of 87.39% was achieved. Quan and Jinye (2008) and Quan *et al.* (2009) presented a vision-

based CSL recognition system. They collected 30 single-handed static manual alphabets. They aimed to extract global and local features rather than focusing only on the local ones. The experimental results showed that the system was classified using SVM.

Yang Quan (2010) proposed a CSL recognition system for recognizing single-handed static alphabets. They used Spatio-temporal appearance modeling for extracting features, and an accuracy of 95.55% was obtained. Li *et al.* (2010) developed an automatic CSL recognition system based on arm sensors. They collected 2420 subwords in total, and features of change of velocity along x, y, and z-axis, 3-axis mean value of each accelerometer, 3-order autoregressive coefficients, and mean absolute value were extracted from arm sensors.

Agarwal and Thakur (2013) proposed a Chinese number, SLRS. They have collected depth and motion profiles of digits from 0-9 using Microsoft Kinect for recognition using an SVM classifier based on linear and RBF kernel.

Geng *et al.* (2014) proposed a novel feature descriptor for recognizing Chinese signs. They have combined the features from depth images and spherical coordinates to represent the feature vector. It has been observed that the system's accuracy increases by adding hand shape features. Zhang *et al.* (2014) presented a CSL system for recognizing double-handed continuous signs. They used Kinect to collect signs, and features were extracted using a threshold matrix. The system was trained with discrete HMM, and the collected signs were classified using DTW.

Zhang *et al.* (2015) developed a Microsoft Kinect system for recognizing 30 isolated signs. The features were extracted using a Histogram of Oriented Displacements, and an accuracy of 88% was attained. Yang *et al.* (2015) presented a Kinect-based SLRS, in which they collected data from 156 single and double-handed words. The features of position, hand shape, movement, and orientation were extracted, and weighted HMM was employed for classification, which led to an accuracy of 97.74%.

Yang *et al.* (2016) developed a continuous SLRS based on Kinect. They collected 20 dynamic sentences, and features of motion trajectory were extracted. The proposed system was classified using LB-HMM (Level Building HMM) and LB-Fast-HMM. The experimental results showed that the computational cost decreases using the LB-fast HMM

algorithm for classification. Pu *et al.* (2016) presented a trajectory modeling-based CSL recognition system. They have employed Kinect to collect 25000 sign samples from 100 words. The findings indicated that raising the testing sample count reduces the system's accuracy. Zhang *et al.* (2016) developed a CSL recognition model based on Adaptive HMM. They have maintained two datasets of dynamic words. The first dataset consists of 100 sign words, and the second consists of 500 sign words. The features of trajectory and shape were recovered using shape context and HOG. It has been noted that the adaptive HMM method performs better than the baseline methods.

Guo *et al.* (2017) developed an adaptive HMM-based SLRS. The features from the captured signs were extracted using HOG and PCA, and an accuracy of 67.34% was obtained.

Huang *et al.* (2018) presented an attention-based CNN for identification of sign language. During the training process of CNN, spatial attention is incorporated into the network to focus on the area of interest. After the features have been extracted, the motions that should be classified are chosen using temporal attention. The experiments were performed on 500 categories of signs and achieved an accuracy of 88.7%.

Pu et al. (2019) suggested a continuous sign language recognition framework. In the first module, 3D ResNet is used for feature learning, while encoder-decoder networks with Connectionist Temporal Classification (CTC) are utilized for sequence modeling. It has been noticed that the suggested strategy outperforms current best practices. Xiao et al. (2019) introduced a multimodal fusion method based on the Couple HMM and Long Short-Term Memory. They compiled and built the dataset using 50 phrases and 150 daily-use isolated terms. The results demonstrated that the suggested strategy outperformed the other alternative methods.

Jiang et al. (2020) introduced an eight-layer CNN for CSL recognition. They gathered a total of 1320 images, in which training set consisted of 1056 images and test set had 264 images. The proposed model achieved an accuracy of 90.91%. Xiao *et al.* (2020) developed a skeleton-based CSL recognition system. They collected the dataset of 500 sign words and implemented it using BiLSTM for both natural and synthetic data. The combination of

BiLSTM and probability model increased the recognition performance of the system to 85.24%.

Zhao *et al.* (2021) presented a SLRS based on the video stream. They collected a dataset of more than 5000 words and implemented CNN for real-time recognition of signs. The proposed method achieved an accuracy of 90.1% on RGB data.

The summarized literature review of Chinese SLRSs is represented in Table 2.5.

Table 2.5: A summarized review of Chinese SLRSs

Author(s)	**Acquisition Mode**	**Single/Double Handed**	**Static/ Dynamic**	**Signing Mode**	**Technique Used**	**Recognition Rate**
Wang *et al.* (2008) [191]	Cyber gloves	Double	Dynamic	Isolated	HMM and DTW	87.39%
Quan and Jinye (2009) [138]	Camera	Single	Static	Isolated	SVM	95.03%
Quan *et al.* (2009) [139]	Camera	Single	Static	Isolated	SVM	93.09%
Yang Quan (2010) [197]	Camera	Single	Static	Isolated	SVM	95.55%
Li *et al.* (2010) [100]	Arm sensors	Both	Both	Isolated	Decision Tree and HMM	95.78%
Agarwal and Thakur (2013) [7]	Kinect	Single	Static	Isolated	Multiclass SVM	81.48% (Linear), 87.67% (RBF)
Geng *et al.* (2014) [51]	Kinect	Single	Static	Isolated	ELM and SVM	80.36%
Zhang *et al.* (2014) [207]	Kinect	Double	Dynamic	Continuous	HMM and DTW	82.2%
Zhang *et al.* (2015) [208]	Kinect	Both	Dynamic	Both	Multi-SVM and DTW	88% (Isolated), 85.2% (Continuous)

Author(s)	Acquisition Mode	Single/Double Handed	Static/ Dynamic	Signing Mode	Technique Used	Recognition Rate
Yang *et al.* (2015) [200]	Kinect	Both	Dynamic	Isolated	Weighted HMM	97.74%
Yang *et al.* (2016) [199]	Kinect	Both	Dynamic	Continuous	HMM	88%
Pu *et al.* (2016) [136]	Kinect	Both	Dynamic	Isolated	HMM	89.8% (1000 samples), 82.7% (18000 samples)
Zhang *et al.* (2016) [209]	Kinect	Both	Dynamic	Isolated	Adaptive HMM	100% (100 words), 98.8% (500 words)
Guo *et al.* (2017) [53]	Kinect	Both	Both	Isolated	Adaptive HMM	67.34%
Huang *et al.* (2018) [64]	Kinect	Both	Static	Isolated	CNN	88.70%
Pu *et al.* (2019) [137]	Camera	Both	Dynamic	Continuous	DTW	67%
Xiao *et al.* (2019) [195]	Kinect	Both	Both	Both	LSTM + Coupled HMM	82.55%
Jiang *et al.* (2020) [69]	Camera	Single	Static	Isolated	CNN	90.91%
Xiao *et al.* (2020) [196]	Kinect	Both	Dynamic	Isolated	Bi-LSTM	85.24%
Zhao *et al.* (2021) [210]	Camera	Both	Dynamic	Isolated	CNN	90.1%

2.2.4.2 Discussions

The results of the review in CSL regarding research questions are addressed below.

To address RQ1, *i.e.,* "Which data acquisition devices have been used mostly for capturing signs in SLRSs?" data has been analyzed to plot graph as shown in Figure 2.8 (a).

It has been observed that 60% of the research work on CSL has been carried out using Kinect, followed by 30% using the camera, 5% using gloves, and the rest 5% using an armband, as shown in Figure 2.8 (a).

To address RQ2, *i.e.,* "How much research is being carried out on static/dynamic signs in SLRSs?" data has been analyzed to plot graph as shown in Figure 2.8 (b).

Figure 2.8 (b) depicts that significant research in CSL has been done for dynamic signs (50%), followed by static signs (35%) and both static and dynamic signs (15%).

To address RQ3, *i.e.,* "What various signing techniques are taken into account when using sign language?" data has been analyzed to plot the graph as represented in Figure 2.8 (c).

It has been seen that the majority of work has been carried out on isolated signs (75%), followed by continuous signs (15%) and both isolated and continuous signs (10%) in CSL, as shown in Figure 2.8 (c).

To address RQ4, *i.e.,* "How to classify and identify single and double-handed signs for SLRS?" data has been analyzed to plot graph as shown in Figure 2.8 (d).

It has been noticed that 30% of work in CSL has been carried out on single-handed signs, 15% on double-handed signs, and 55% on both single and double-handed signs, as shown in Figure 2.8 (d).

To address RQ5, *i.e.,* "What are the existing methodologies and techniques available to recognize sign language recognition?" data has been analyzed to plot a graph, as shown in Figure 2.8 (e).

Figure 2.8 (e) depicts that 30% of the work on CSL has been performed using hybrid techniques, followed by HMM (25%), CNN (20%), and SVM (20%), while the minimum amount of work has been performed using DTW (5%).

To address RQ6, *i.e.,* "What is the accuracy and coverage of existing SLRSs?" data has been analyzed to plot graph as represented in Figure 2.8 (f).

It has been observed that for CSL, there are 40% of SLRSs achieved an average accuracy of greater than 90%, while 50% of the systems have an accuracy between 80-89%. Only 10% of systems have less than 80% accuracy, as given in Figure 2.8 (f).

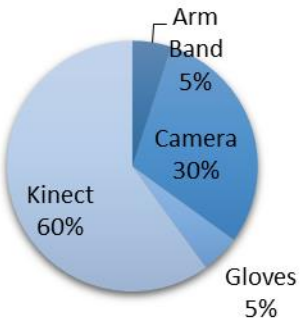


Figure 2.8 (a): Usage of different data acquisition techniques used in CSL systems

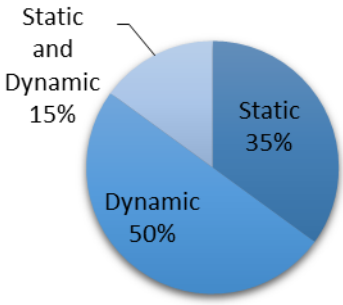


Figure 2.8 (b): Research work carried out on static/dynamic signs in CSL

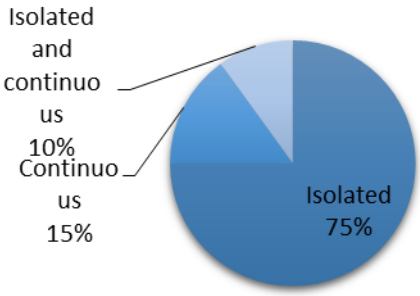


Figure 2.8 (c): Percentage of research work carried out based on signing mode in CSL

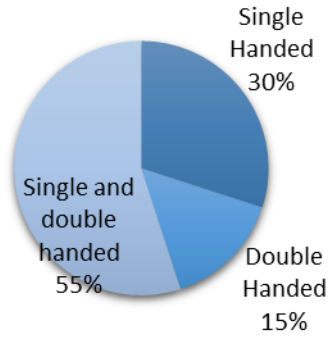


Figure 2.8 (d): Percentage of research work carried out based on single/double-handed signs in CSL

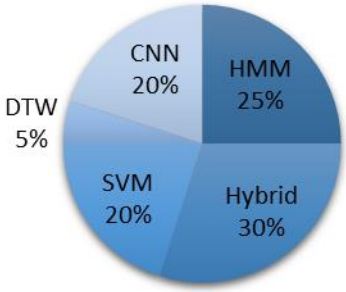


Figure 2.8 (e): Percentage of research work carried out on technique used for recognition of signs

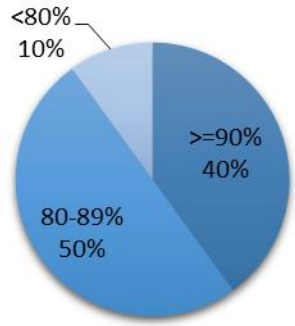


Figure 2.8 (f): Accuracy of research for different CSL systems

2.2.5 Persian Sign Language

The sign language used by deaf persons in Iran is Persian Sign Language (PSL). The strategies for sign language recognition for Persian SL that have been described in recent years are listed below.

2.2.5.1 Persian Sign Language Recognition Techniques

Sarkalehl et al. (2009) created a NN-based system for PSL recognition. The total dataset consisted of 240 single-handed static words captured using the camera. The accuracy of 98.75% was achieved by using 10 neurons present in the hidden layer of the multilayer perceptron neural network.

Karami *et al.* (2011) implemented a SLRS based on wavelet transform. They have captured static single-handed Persian alphabets, and discrete wavelet transform has been employed to extract features from different signs. Moghaddam *et al.* (2011) proposed a kernel-based feature extraction method for recognizing static signs in Persian SL. They used Kernel Principle Component Analysis (KPCA) and Kernel Discriminant Analysis (KDA) for feature extraction. The system was tested with 35 alphabets using SVM and NN classifiers, and the accuracy of 95.51% and 95.91% were achieved, respectively.

Madani and Nahvi (2013) presented a SLRS for 20 dynamic signs. They have used radon transform for extracting features and found that it has a good effect on obtaining maximum recognition rate.

Azar and Seyedarabi (2016) developed a dynamic PSL recognition system. The dataset contains 750 videos from 15 signs and extracts hand trajectory using the Spline interpolation method. The developed system was classified using HMM, and the accuracy of 95.3% in signer-dependent mode and 78% in independent signer mode was achieved.

Zadghorban and Nahvi (2018) provided a method for identifying and recognizing word boundaries in continuous PSL movies. They designed two classification models based on motion features: the HMM and a hybrid KNN-DTW method for classification by hand shape data. The results demonstrated that the suggested way has an average accuracy of 93.73%.

Rastgoo *et al.* (2020 a) presented a SLRS using a single shot detector (SSD), CNN, and LSTM. They collected 10,000 RGB videos from 100 PSL signs and extracted novel hand

skeleton features. The results represented that the implemented method obtained an accuracy of 99.80%. Rastgoo et al. (2020 b) suggested an SLRS for PSL in which features from hands, Extra Spatial Hand Relation (ESHR) features, and Hand Posture (HP) were retrieved. They used LSTM for extracting temporal features, and the model resulted in an accuracy of 98.42% using CNN.

Khomami and Shamekhi (2021) presented a KNN-based SLRS. They used surface electromyography (sEMG) and Inertial Measurement Unit (IMU) sensors to collect 20 PSL signs. The system obtained an accuracy of 96.13% with 25 highest ranked features. Rastgoo *et al.* (2021) presented a real-time isolated Persian SLRS using CNN, Singular Value Decomposition (SVD), and LSTM. SVD has been employed as a discriminative feature extractor from the estimated 3D hand keypoints coordinators and achieved an accuracy of 99.5%.

The summarized literature review of Persian SLRSs is shown in Table 2.6.

Table 2.6: A summarized review of Persian SLRSs

Author(s)	**Acquisition Mode**	**Single/Double Handed**	**Static/ Dynamic**	**Signing Mode**	**Technique Used**	**Recognition Rate**
Sarkalehl *et al.* (2009) [159]	Camera	Single	Static	Isolated	NN	98.75%
Karami *et al.* (2011) [73]	Camera	Single	Static	Isolated	NN	94.06%
Moghaddam *et al.* (2011) [115]	Camera	Single	Static	Isolated	NN and SVM	95.91% (NN), 95.51% (SVM)
Madani and Nahvi (2013) [103]	Camera	Both	Dynamic	Isolated	KNN, NN, and SVM	92.22% (SVM)
Azar and Seyedarabi (2016) [23]	Camera	Single	Dynamic	Isolated	HMM	95.3% (Signer dependent), 78% (Independent)

Author(s)	Acquisition Mode	Single/Double Handed	Static/ Dynamic	Signing Mode	Technique Used	Recognition Rate
Zadghorban and Nahvi (2018) [203]	Camera	Both	Dynamic	Both	KNN-DTW	93.73%
Rastogi *et al.* (2020a) [144]	Camera	Both	Static	Isolated	CNN	99.80%
Rastogi *et al.* (2020b) [145]	Camera	Both	Static	Isolated	CNN	98.42%
Khomami and Shamekhi (2021) [79]	Arm Band	Single	Dynamic	Isolated	KNN	96.13%
Rastgoo *et al.* (2021) [146]	Camera	Both	Both	Isolated	CNN	99.50%

2.2.5.2 Discussions

The results of the review in PSL regarding research questions are addressed below.

To address RQ1, *i.e.,* "Which data acquisition devices have been used mostly for capturing signs in SLRSs?" data has been analyzed to plot graph as shown in Figure 2.9 (a).

Figure 2.9 (a) specifies that 90% of the research on PSL has been done using a camera and 10% using an armband as an acquisition device.

To address RQ2, *i.e.,* "How much research is being carried out on static/dynamic signs in SLRSs?" data has been analyzed to plot graph as shown in Figure 2.9 (b).

Figure 2.9 (b) depicts that the significant research in PSL has been performed on static signs (50%), and the rest 50% on dynamic signs.

To address RQ3, *i.e.,* "What various signing techniques are taken into account when using sign language?" data has been analyzed to plot the graph as represented in Figure 2.9 (c).

Figure 2.9 (c) specifies that 90% of the research on PSL has been done for isolated signs, followed by 10% for both isolated and continuous signs.

To address RQ4, *i.e.,* "How to classify and identify single and double-handed signs for SLRS?" data has been analyzed to plot a graph, as shown in Figure 2.9 (d).

It has been noticed that the 50% of the research work in PSL has been carried out on signs using a single hand, followed by both single-handed and two-handed signs (50%), as shown in Figure 2.9 (d).

To address RQ5, *i.e.*, "What are the existing methodologies and techniques available to recognize sign language recognition?" data has been analyzed to plot a graph, as shown in Figure 2.9 (e).

Figure 2.9 (e) depicts that 30% of the work on PSL has been performed using CNN and hybrid techniques, while 20% uses NN, followed by 10% using KNN and HMM.

To address RQ6, *i.e.*, "What is the accuracy and coverage of existing SLRSs?" data has been analyzed to plot graph as shown in Figure 2.9 (f).

It has been observed that for Persian SL, 95% of SLRSs achieved an average accuracy of greater than 90%, while only 5% of systems whose accuracy is less than 80%, as shown in Figure 2.9 (f).

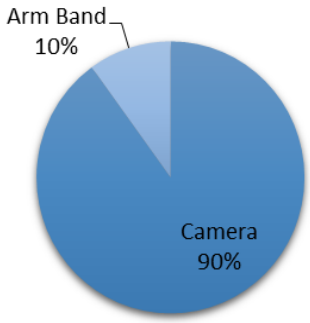


Fig 2.9 (a): Usage of different data acquisition techniques used in PSL systems

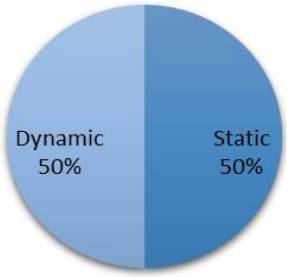


Figure 2.9 (b): Research work carried out on static/dynamic signs in PSL

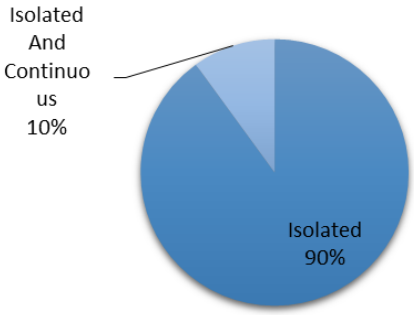


Figure 2.9 (c): Percentage of research work carried out based on signing mode in PSL

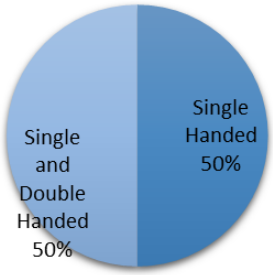


Figure 2.9 (d): Percentage of research work carried out based on single/double-handed signs in PSL

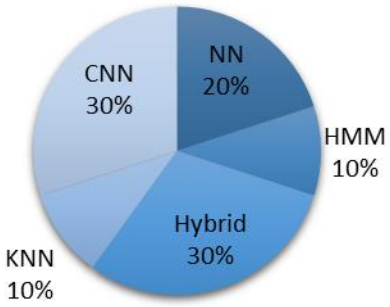


Figure 2.9 (e): Percentage of research work carried out on technique used for recognition of signs

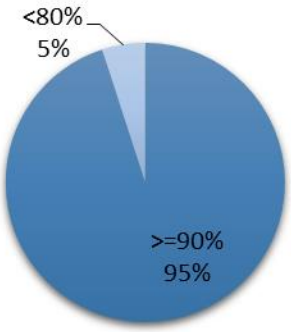


Figure 2.9 (f): Accuracy of research for different PSL systems

2.2.6 Brazilian Sign Language

Brazilian SL is used by deaf people in the region of Brazil. The estimate for Brazilian SL speakers is 3,000,000 [40]. The techniques to sign language recognition for Brazilian SL that have been documented in recent years are outlined in the following section.

2.2.6.1 Brazilian Sign Language Recognition Techniques

Dias *et al.* (2009) developed a hand movement recognition system for Brazilian SL. The collected single-handed signs were classified using distance-based neural networks. The experimental results showed that the unsupervised fuzzy learning vector quantization model outperformed, with an accuracy of 98.89%.

De Paula Neto *et al.* (2015) proposed an Extreme Learning Machine (ELM) based system for real-time recognition of Brazilian signs. They captured 990 single-handed static signs from 18 alphabets. In the feature extraction phase, features of magnitude and direction of the edges were extracted, and an accuracy of 95.92 % was obtained. Abreu *et al.* (2016) evaluated the performance of the electromyogram data collected by the armband. They captured 20 static single-handed Brazilian signs and achieved a cross-validation accuracy of 98.56%.

Lima et al. (2019) presented a CNN-based approach for Brazilian Sign Language fingerspelling recognition. They gathered 2,24,000 images with varying backdrops, body parts, hand postures, and lighting patterns. According to the findings, the algorithm attained an average accuracy of 99% for independent person scenarios and 71% for independent person scenarios.

Junior *et al.* (2020) presented an approach to analyze the impact of segmentation, features, and classifier on the recognition of Brazilian sign language. They collected a dataset of 26 signs of Libras alphabets using an armband. The collected signs were classified using different classifiers like KNN, Naïve Bayes, random forest, neural networks, *etc.*, and obtained an average accuracy of 99%.

Rezende *et al.* (2021) presented the implementation and validation of the Brazilian sign language public dataset. They collected 1200 sign samples from 20 signs and implemented the technique of CNN, which resulted in an average accuracy of 93.3%.

The summarized literature review of Brazilian SLRSs is given in Table 2.7.

Table 2.7: A summarized review of Brazilian SLRSs

Author(s)	Acquisition Mode	Single/Double Handed	Static/ Dynamic	Signing Mode	Technique Used	Recognition Rate
Dias *et al.* (2009) [40]	Camera	Single	Dynamic	Isolated	NN	93% (supervised), 98.89% (unsupervised)
De Paula Neto *et al.* (2015) [38]	Camera	Single	Static	Isolated	ELM	95.92%
Abreu *et al.* (2016) [3]	Myo Armband	Single	Static	Isolated	SVM	41.15% (real-time), 98.56% (cross-validation)
Lima *et al.* (2019) [99]	Camera	Single	Both	Isolated	CNN	71%
Junior *et al.* (2020) [112]	Arm Band	Single	Both	Isolated	Hybrid	99%
Rezende *et al.* (2021) [152]	Camera	Both	Both	Isolated	CNN	93.3%

2.2.6.2 Discussions

The results of the review in BSL regarding research questions are addressed below.

To address RQ1, *i.e.*, "Which data acquisition devices have been used mostly for capturing signs in SLRSs?" data has been analyzed to plot graph as shown in Figure 2.10 (a).

The bulk of the study on Brazilian sign language has been conducted using cameras (67%) and armbands for the remaining 33%, as shown in Figure 2.10 (a).

To address RQ2, *i.e.,* "How much research is being carried out on static/dynamic signs in SLRSs?" data has been analyzed to plot graph as shown in Figure 2.10 (b).

It has been found that significant research in Brazilian SL has been carried out on static signs (33%) followed by dynamic signs (17%) and 50% by using both static and dynamic signs, as shown in Figure 2.10 (b).

To address RQ3, *i.e.,* "What various signing techniques are taken into account when using sign language?"

The review on Brazilian sign language depicts that 100% of the research has been done only on isolated signs.

To address RQ4, *i.e.,* "How to classify and identify single and double-handed signs for SLRS?"

It has been noticed that significant work in Brazilian SL has been carried out on single-handed signs (83%) and the rest 17% on both single and double-handed signs.

To address RQ5, *i.e.,* "What are the existing methodologies and techniques available to recognize sign language recognition?" data has been analyzed to plot a graph, as given in Figure 2.10(c).

Figure 2.10 (c) depicts that 33% of the work on Brazilian SL has been performed using NN and CNN, followed by 17% using SVM and Hybrid techniques individually.

To address RQ6, *i.e.,* "What is the accuracy and coverage of existing SLRSs?" data has been analyzed to plot graph as shown in Figure 2.10 (d).

It has been observed that for Brazilian SL, 75% of SLRSs achieved an average accuracy of greater than 90%, while only 25% of systems whose accuracy is less than 80%, as shown in Figure 2.10 (d).

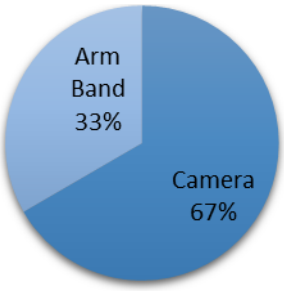


Figure 2.10 (a): Usage of different data acquisition techniques used in Brazilian SL systems

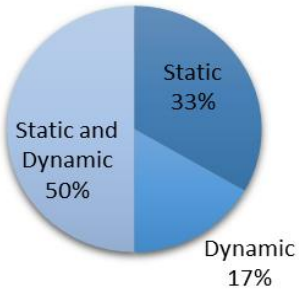


Figure 2.10 (b): Research work carried out on static/dynamic signs in Brazilian SL

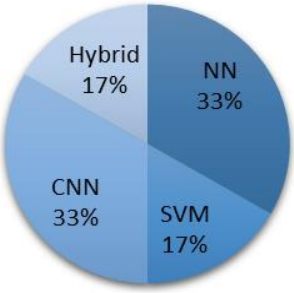


Figure 2.10 (c): Percentage of research work carried out on technique used for recognition of signs

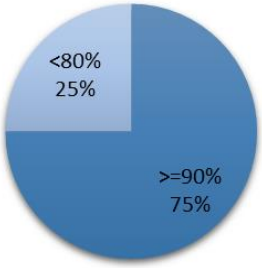


Figure 2.10 (d): Accuracy of research for different Brazilian SL systems

2.2.7 Thai Sign Language

Thai sign language is the official sign language of Thailand's deaf population. It is a member of the same family as American Sign Language. 20 percent of the estimated 56,000 pre-linguistically deaf persons in the United States utilize it [150]. Techniques for recognizing Thai sign language that has been described in recent years are mentioned below.

2.2.7.1 Thai Sign Language Recognition Techniques

Saengsri *et al.* (2012) developed a Thai finger-spelling SLRS. They employed data gloves and a motion tracker for capturing single-handed static signs. The collected signs were classified using Elman BPNN, abductions between fingers, positions, and orientation features of the hand. Adhan and Pintavirooj (2016) presented a SLRS based on geometric invariants and ANN. The dataset consisted of 1470 Thai alphabets, of which 1050 were used for training and 420 for testing. The collected signs were identified using feed-forward NN, and an accuracy of 96.19% was obtained.

Pariwat and Seresangakul (2017) developed a camera-based fingerspelling recognition system to identify 15 Thai alphabets. The experimental findings demonstrate that integrating local and global data improved the system's accuracy, while the RBF kernel outperformed polynomial, linear, and sigmoidal-based classification.

Pariwat and Seresangtakul (2019) presented a finger spelling recognition system using local features and the Pyramid Histogram of Oriented Gradients (PHOG). They collected 375 images in total, and experiments were performed using KNN, which resulted in an accuracy of 97.6%. Nakjai and Katanyukul (2019) proposed a Thai SLRS using CNN and HOG-based approaches. They collected 1375 sign images from single-handed 25 sign postures and obtained an accuracy of 91.26%. Sripairojthikoon and Harnsomburana (2019) suggested a CNN-based model for identifying Thai sign language that can automatically learn spatial and temporal characteristics. They gathered 64 isolated signs using Kinect to obtain data on color, depth, skeleton, the shape of hands, and the whole body. The results showed that the highest accuracy of 97.7% has been achieved.

Pariwat and Seresangtakul (2021) developed a SLRS using deep learning. They collected 1,25,000 sign samples from 25 signs. The experiments were performed on different CNN structures, and it was found that the AlexNet model results in the highest accuracy of 92.03%.

The summarized literature review of Thai SLRSs is shown in Table 2.8.

Table 2.8: A summarized review of Thai SLRSs

Author(s)	**Acquisition Mode**	**Single/Double Handed**	**Static/ Dynamic**	**Signing Mode**	**Technique Used**	**Recognition Rate**
Saenger *et al.* (2012) [153]	Data gloves	Single	Static	Isolated	NN	94.44%
Adhan and Pintavirooj (2016) [4]	Camera	Both	Both	Isolated	NN	96.19%
Pariwat and Seresangtakul (2017) [131]	Camera	Single	Static	Isolated	SVM	91.20% (RBF kernel)
Pariwat and Seresangtakul (2019) [132]	Camera	Single	Static	Isolated	KNN	97.60%

Author(s)	Acquisition Mode	Single/Double Handed	Static/ Dynamic	Signing Mode	Technique Used	Recognition Rate
Nakjai and Katanyukul (2019) [123]	Camera	Single	Static	Isolated	CNN	91.26%
Sripairojthikoon and Harnsomburana (2019) [171]	Kinect	Both	Static	Isolated	CNN	97.7%
Pariwat and Seresangtakul (2021) [133]	Camera	Single	Static	Isolated	CNN	92.03%

2.2.7.2 Discussions

The review results for Thai sign language research questions are addressed below.

To address RQ1, *i.e.,* "Which data acquisition devices have been used mostly for capturing signs in SLRSs?" data has been analyzed to plot graph as shown in Figure 2.11 (a).

It has been found that 72% of research in Thai SL has been implemented using a camera, followed by 14% using gloves and Kinect individually, as shown in Figure 2.11 (a).

To address RQ2, *i.e.,* "How much research is being carried out on static/dynamic signs in SLRSs?" data has been analyzed to plot graph as given in Figure 2.11 (b).

Figure 2.11 (b) depicts that the significant research work on Thai SL has been performed on static signs (86%) followed by both static and dynamic signs (14%).

To address RQ3, *i.e.,* "What various signing techniques are taken into account when using sign language?"

The review on Thai SL specifies that 100% of the research in this sign language has been done on isolated signs.

To address RQ4, *i.e.,* "How to classify and identify single and double-handed signs for SLRS?" data has been analyzed to plot graph as shown in Figure 2.11 (c).

It has been noticed that the significant work in Thai SL has been done on signs with one hand (71%) and the rest 29% on both single and double-handed signs, as shown in Figure 2.11 (c).

To address RQ5, *i.e.*, "What are the existing methodologies and techniques available to recognize sign language recognition?" data has been analyzed to plot a graph as shown in Figure 2.11 (d).

It has been found that 43% of the work on Thai SL has been implemented using CNN, followed by 29% using NN and 14% using SVM and KNN individually, as shown in Figure 2.11 (d).

To address RQ6, *i.e.*, "What is the accuracy and coverage of existing SLRSs?"

It has been observed that for Thai SL, there are 100% of SLRSs achieved an average accuracy of greater than 90%.

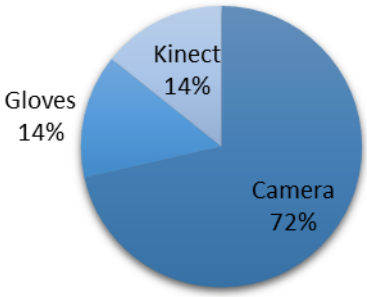


Figure 2.11 (a): Usage of different data acquisition techniques used in Thai SL systems

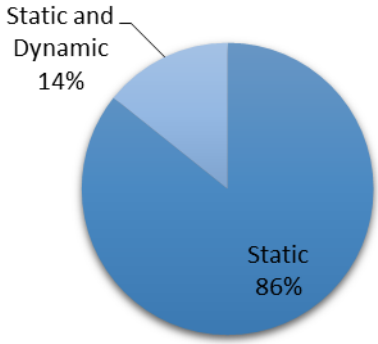


Figure 2.11 (b): Research work carried out on static/dynamic signs in Thai SL

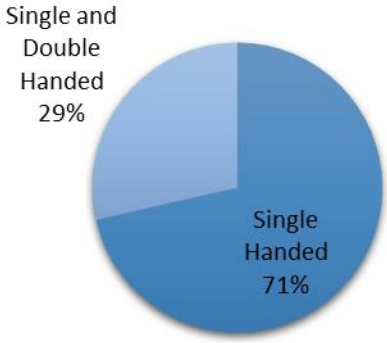


Figure 2.11 (c): Percentage of research work carried out based on single/double-handed signs in Thai SL

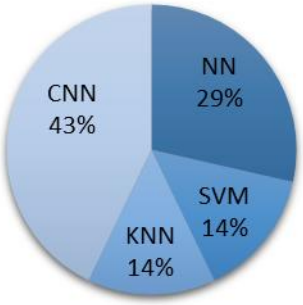


Figure 2.11 (d): Percentage of research work carried out on technique used for recognition of signs

Most of the research on sign language recognition has been done for ASL, ISL, ArSL, CSL, PSL, Brazilian, and Thai sign languages. The minimum amount of work done for sign languages like Greek, Irish, Malaysian, Mexican, Taiwanese, German, Japanese, South

African, Sri Lankan, Auslan, Bangladeshi, Ecuadorian, Ethiopia, Farsi, Italian, Polish, Spanish, and Ukrainian sign language has been discussed in the next section.

2.2.8 Other Languages

Kosmidou and Hadjileontiadis (2008) proposed an intrinsic mode entropy-based Greek SL gesture recognition system, in which they have collected both single and double-handed words using an armband. The results showed that an accuracy of 100% was attained [85]. Theodorakis *et al.* (2009) developed a SLRS for recognizing Greek words. They collected dynamic double-handed signs from 93 words and extracted movement and hand shape features. The experimental results showed increased performance by fusing movement and hand shape information with Product HMM [183]. Simos and Nikolaidis (2016) presented a method for recognizing Greek SL alphabets collected using Leap Motion. The system extracts features using Leap motion APIs, and the Radial Basis Function (RBF) kernel of SVM was applied [169].

Kelly *et al.* (2009a) developed a model for identification of sentences. They have captured double-handed dynamic signs in the Irish language using the camera. The features of the right and left hands were extracted using the mean shift algorithm. The system was classified using Multi-channel HMM, which leads to an accuracy of 95.7% [75]. Kelly *et al.* (2009b) proposed a framework for recognizing Irish SL sequences. They have incorporated facial features along with a multi-channel gesture recognition system. The system was classified using HMM, and an accuracy of 95.10% was attained [77]. Kelly *et al.* (2010) presented a person independent hand posture recognition system. They have used a camera and colored gloves to collect two different Irish Sign Language datasets. Features with Hu moments and Eigen space size function were extracted from the data collected using a camera, and the contour of the hand blob from signs was collected using colored gloves [76].

Akmeliawati *et al.* (2007) developed a real-time Malaysian SLRS. They gathered 36 static and 10 dynamic signs using the camera. The collected numbers, alphabets, and words were classified using a neural network [12]. Paulraj *et al.* (2008) presented an approach for extracting head and hand gestures that helps in recognition of Malaysian SL. The collected signs were classified using BPNN, and an accuracy of 92.07% was obtained [134]. Majid *et*

al. (2015) presented a Malaysian SLRS using skeleton data received from Kinect. They captured 375 signs in total from 15 double-handed dynamic signs. The features of 3D coordinates from 8 joints were extracted, and it has been observed that the spherical coordinate features performed better than the Cartesian coordinate system [105].

Luis-Pérez *et al.* (2011) implemented a NN based system that controls a service robot using Mexican SL. They collected 23 static single-handed alphabets segmented using active contours, and an accuracy of 95.80% was achieved [101]. Galicia *et al.* (2015) proposed a model for converting Mexican SL into the Spanish language. They utilized a Kinect sensor for capturing 867 images, which were learned using a decision tree algorithm, and an accuracy of 76.19% was achieved [47]. Bautista *et al.* (2017) presented a Mexican SL recognition model using Kinect. They gathered 700 samples of 20 Mexican words, from which skeleton data was extracted. The signs were then classified using DTW, and the accuracy of 98.57% was attained on real-time data [49].

Lee and Tsai (2009) presented a Taiwanese Sign Language (TSL) identification system in which data was collected using the camera. They managed static single-handed 2788 signs and extracted 15 geometric distance values. Experiments were performed on 1438 signs of testing data, and an accuracy of 94.65% was achieved [96]. Huang and Tsai (2010) developed a vision-based TSL recognition model in which they combined a camera and colored gloves to acquire data. On this hand, shape features were extracted using Fourier descriptors and Hu moments, and an accuracy of 89% for dynamic signs was obtained [63]. Yu *et al.* (2011) presented a vision-based continuous TSL recognition system. The Product HMM was employed for classifying the collected signs, and an average accuracy of 67% was achieved [202].

Kim *et al.* (2008) proposed a Bi-channel sensor fusion-based automatic German SLRS [80]. Lang *et al.* (2012) presented a German SLRS using Kinect [93]. Sako and Kitamura (2013) introduced Japanese SLRs classified using Product HMM [156]. Mukai *et al.* (2017) developed a Japanese fingerspelling and SVM-based recognition system [120]. Hosoe *et al.* (2017) proposed a Japanese fingerspelling model in which the CNN was employed for classification, and an accuracy of 93% was achieved [61].

Nel *et al.* (2013) presented an integrated South African Sign Language recognition method in which signs were captured using the camera [124]. Seymour and Tšoeu (2015) developed a mobile application for recognizing South African sign language [162].

Vanjikumaran and Balachandran (2011) described a vision-based approach for automatically identifying Sri Lankan Tamil finger spelling [189]. Madushanka *et al.* (2016) created a framework for identifying Sinhala sign language [104]. Thang *et al.* (2017a) studied the effectiveness of vector-based machine learning methods named SVM, Simplification of SVM, and Relevance Vector Machine (RVM) for Auslan signs [181]. Thang *et al.* (2017b) compared the effectiveness of SimpSVM and RVM for Auslan sign language recognition [182].

Admasu and Raimond (2010) presented an Ethiopian SLRS based on camera and ANN [6]. Zare and Zahiri (2016) proposed a signer independent static Farsi SLRS based on a camera [206]. Ahmed and Akhand (2016) presented a camera-based Bangladeshi SLRS [10]. Jiménez *et al.* (2017) implemented a system for recognition of single and double-handed Bangladeshi words [70].

Infantino *et al.* (2007) presented a framework for recognizing sentences in Italian sign language [65]. Oszust and Wysocki (2013) proposed a recognition system for recognizing Polish signs using Kinect [126]. Parcheta & Martínez-Hinarejos (2017) developed a Spanish SLRS using HMM [130]. Davydov *et al.* (2010) proposed a real-time camera-based Ukrainian SLRS [37].

Basiri *et al.* (2021) developed a dynamic Iranian SLRS using hand gloves and DNN. This system is required to create Human-Robot Interaction (HRI) platforms that can interact with human beings through sign language. The collected signs were optimized using a genetic algorithm, and it has been observed that the accuracy gets raised to 99.7% using the optimization procedure [27].

Gruber *et al.* (2021) developed an ensemble-based approach for sign language identification. The final ensemble comprises 13 3D models, 1 pose transformer model, and 3 virtual learning extractors. The collected dataset from Turkish sign language showed a weighted accuracy of 95.46% [52]. Jiang et al. (2021) presented a Skeleton Aware Multi-modal (SAM)

architecture for Turkish sign language. They proposed the Sign Language Graph Convolution Network (SL-GCN) for modeling embedded dynamics and the Separable Spatial-Temporal Convolution Network (SSTCN) for exploiting skeletal characteristics. The experiments showed that the SAM model obtained the highest accuracy of 98.53% on RGB depth images [68]. Sincan et al. (2021) demonstrated a recognition system based on Turkish Sign Language. The authors summarized the ChaLearn LAP Large Scale Signer Independent Isolated SLR Challenge. The results revealed that the top two winning solutions benefitted from Graph Convolutional Networks (GCN) and obtained a recognition rate greater than 96% [170]. Moryossef *et al.* (2021) presented a SLRS for Turkish sign language. They fused both the OpenPose and MediaPipe Holistic methods for pose estimation. It has been found that both the pose estimation techniques using LSTM performed well on unseen data and attained the highest accuracy of 84.16% on the validation set [119].

2.2.9 Overall observation by considering the research work on all SLRSs

The comprehensive report for the results on SLRSs regarding research questions is addressed below.

To address RQ1, we observed that 56% of the research work on SLRSs had been done using cameras, followed by 21% using Kinect, 7% using gloves, 5% using armband, 7% using leap motion, and the rest 4% using other acquisition devices as shown in Figure 2.12 (a). To address RQ2, Figure 2.12 (b) depicts that the majority of research on SLRSs has been carried out for static signs (43%), followed by dynamic signs (32%) and for both static and dynamic signs (25%). To address RQ3, it has been observed from Figure 2.12 (c) that majority of work has been performed on isolated signs (82%), followed by continuous signs (11%) and isolated and continuous signs (7%) SLRSs. To address RQ4, we found that 44% of work on SLRSs has been carried out on single-handed signs, followed by 15% on double-handed signs and 41% on both single and double-handed signs, as shown in Figure 2.12 (d). To address RQ5, Figure 2.12 (e) depicts that the majority of the work on SLRSs has been performed using CNN (23%), followed by NN (18%), hybrid techniques, and SVM (17%) individually, HMM (13%). In contrast, the minimum work has been performed using DTW, KNN, and other techniques. To address RQ6, we observed that for all the sign language

systems, there are 66% of SLRSs achieved an average accuracy of greater than 90%, while 22% of the systems have an accuracy between 80-89%. Only 12% of systems have less than 80% accuracy, as shown in Figure 2.12 (f).

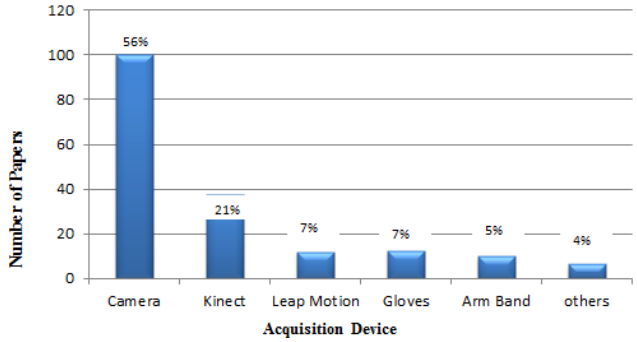


Figure 2.12 (a): Usage of different data acquisition techniques used in sign language systems

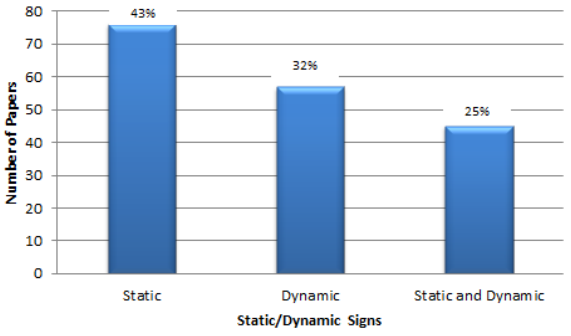


Figure 2.12 (b): Research work carried out on static/dynamic signs in different sign languages

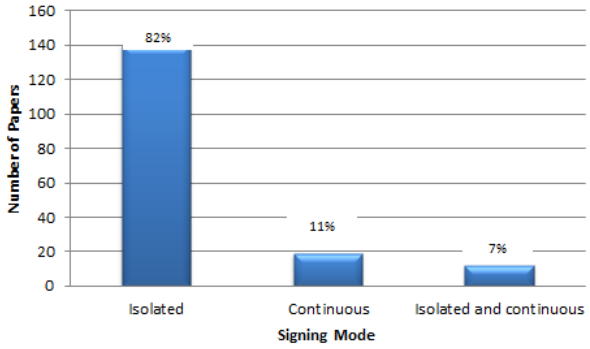


Figure 2.12 (c): Research work carried out based on signing mode in different sign languages

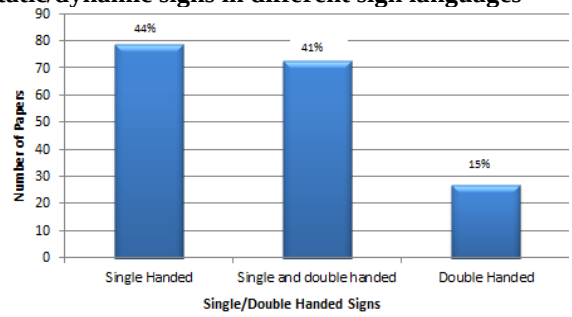


Figure 2.12 (d): Research work was carried out based on single/double-handed signs in different sign languages

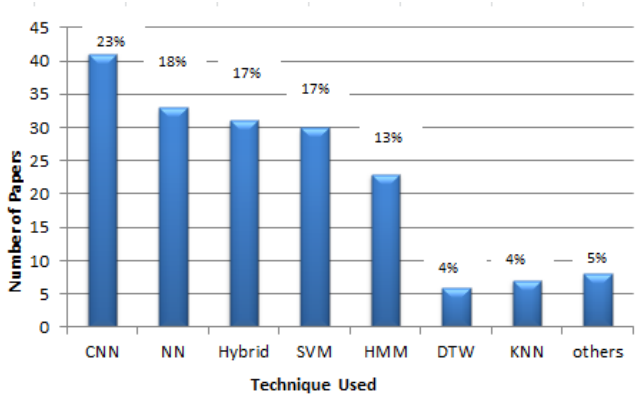


Figure 2.12 (e): Research work was carried out on the technique used for the recognition of signs

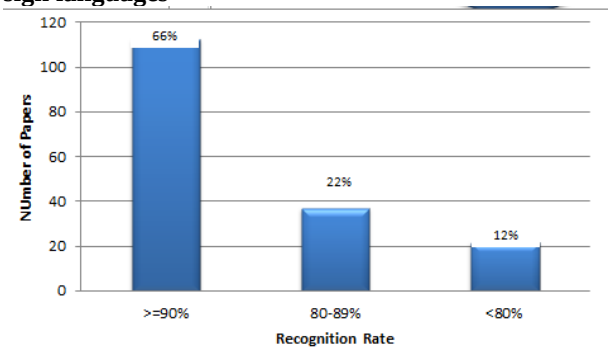


Figure 2.12 (f): Accuracy of research for different sign language systems

The application of SLRSs is an emerging social trend. It has attracted the interest of academicians and practitioners. This review gives valuable insights and demonstrates the prevalence of studies for SLRS. The findings provided in this chapter have many significant consequences:

- Based on historical publication rates and growing interest in the field, SLRS research will rise dramatically in the future.
- During Systematic Literature Review, we observed that the maximum research had been performed for Indian Sign Language (20.1%). 18.9% of studies focused on American Sign Language, followed by Arabic Sign Language (15%), Chinese Sign Language (11.1%), Persian Sign Language (5.5%), and rest 29% on other sign languages, which include Brazilian, Greek, Irish, Malaysian, Mexican, Taiwanese, Thai, German, Japanese, Italian, Ethiopian, Ecuadorian, Farsi, South African, Sri Lankan, Auslan, Bangladeshi, Spanish, Polish and Ukrainian sign languages.

2.2.10 Gaps in Literature Survey

The challenges and issues related to sign language recognition using various machine learning and deep learning approaches are listed below.

i. Small datasets: Small datasets has become a major confront for research scientists due to the limited availability of the sign language dataset for training purposes. The large dataset size must be appropriate for the training process; otherwise, the training procedure will not perform well. In real-life situations, researchers have faced the challenge of small dataset for training different machine learning and deep learning models.
ii. Data collection and annotation: One of the crucial challenges in sign language recognition is the lack of large, diverse, and well-annotated datasets. Many sign languages have regional variations, and collecting sufficient data to cover these variations can be challenging.
iii. Variability and adaption: Systems for recognizing sign language must be resistant to changes in signing patterns, illumination, camera angles, and backgrounds. It is difficult to create models that can adjust to such diversity and still function accurately.

iv. Real-time processing: To achieve real-time performance in sign language recognition is essential for live interpretations. Many existing systems have latency issues and reducing the same without sacrificing for accuracy is an ongoing challenge.

v. Cross-language Adaptation: Another challenge in sign language recognition is cross language adaptation. In this the sign language recognition model should be developed in such a manner so that it will be able to recognize signs from different sign languages without requiring extensive data.

vi. Multimodal Integration: Combining video based sign language recognition with audio could enhance the overall experience of users. More research is required to explore how these modalities can be integrated effectively.

vii. Variability of Image Quality: The image quality is based on the system's settings and specifications. The images with inaccurate sharpness, contrast, poor resolution, and elevated noise can affect the model's efficiency while training.

Chapter Summary

In this chapter, the literature review based on the research carried out by researchers on sign language recognition has been documented. The systematic literature review identified one hundred and seventy-nine research articles related to sign language recognition and published between 2007-2021. This chapter aims to offer an overview of research-based sign language, which is further classified using several factors such as data collecting technique, static/dynamic signs, signing mode, single/double-handed signs, classification approach, and recognition rate. Since doing a Systematic Literature Review requires a great deal of time and work, this literature evaluation seeks to save future researchers time and effort by offering a comprehensive and exhaustive review of SLRSs for various sign languages.

After an exhaustive survey on various SLRSs, it has been observed that existing sign language recognition systems have made significant progress, but they also have several limitations that impact their performance and usability. These limitations include small datasets, dataset collection and annotation, variability and adaption, real-time processing, cross language adaptation, multimodal integration and variability of image quality. So the proposed SLRS will be a significant step in removing these challenges for sign language recognition and to reduce the language barrier between hearing-impaired and non-hearing impaired persons. This literature review not only serves as a foundation for the research conducted in this thesis but also highlights the existing challenges, unaddressed gaps, and the potential of the proposed SLRS to make a significant difference in the lives of individuals with hearing impairments.

CHAPTER 3

Data Acquisition

Automatic recognition of spontaneous signs and facial expressions is a challenging task in computing. In practical applications, rotation of the head, occlusion, illumination variation, etc., are the properties that lead to the increase in the complexity of sign recognition.

To develop a SLRS, data is the most important. By going through existing research articles and sign language datasets, it has been found that no public dataset is available for sign language recognition on Indian signs. Many researchers worked upon different techniques for recognizing signs, but their work lacks evaluation on a benchmark database. In this work, we have created two datasets. The first dataset is for static signs, and the second dataset have been created for dynamic signs. This chapter explains the whole procedure followed for collecting and developing the dataset for static signs as well as for dynamic signs. It also describes how the dataset has been collected from the number of users under different environmental conditions and at different distances. In this research, a new dataset have been created for the recognition of static and dynamic signs whose detail has been discussed in next sections.

3.1 Data Acquisition

The data acquisition phase aims to collect and develop a dataset for SLRSs. To prepare the dataset, it has been found that the data acquisition for sign language and gesture recognition systems has been performed by various researchers by using wearable computing-based and vision-based devices, as shown in Figure 3.1. Wearable computing-based acquisition involves hand gloves and armbands in which detection of hand and face is eliminated by using sensors available in the gloves/ armbands. On the other hand, vision-based devices include web cameras and Kinect. These methods are more natural and more helpful for real-time applications.

3.1.1 Wearable Computing-based Acquisition

For the acquisition of gestures and signs, wearable computing devices are used. Typically, sensors are connected to hand gloves or arm bands for recognition of sign language. This acquisition method provides various technologies for capturing hand shape and movement and can translate sign language into text and even voice.

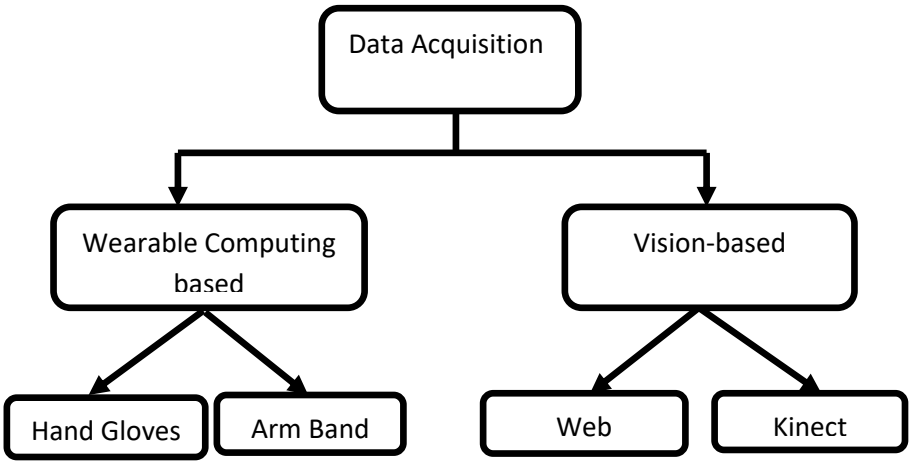


Figure 3.1: Data Acquisition Approaches

Indeed, wearable devices for recognizing sign languages are popular among researchers. However, there are several obstacles related to acquisition based on wearable computing, which are mentioned below.

i. The signer has to wear the hardware sensor.
ii. It requires the use of complex signal processing algorithms for extracting information about gestures from the captured data.
iii. Different people may have varying hand sizes, finger thicknesses, and heights. While recognition, other locations of fingers for various users may overlap, which results in reduced accuracy.
iv. The outputs generated by using wearable devices lead to the noise as wear and tear [45], extension and flexion of wrist movements [130] [45], hand grip force [45], and poor calibration [41].

To overcome the drawbacks of this approach, researchers are using vision-based methods.

3.1.2 Vision-based acquisition

Vision-based systems, such as web/mobile cameras and Kinect, use image and video processing techniques to recognize hand and finger motions and extract various characteristics. The benefit of vision-based systems is that they do not need the attachment of sensory equipment to the user's body, which is unpleasant for the user. However, vision-based systems are challenging to construct due to the computationally intensive nature of designing algorithms for a feature and movement detection. The detailed description of vision-based acquisition devices has been presented as follows.

Web Camera

A web camera is a device used for capturing images and video clips. The video signal contains a series of individual image frames that presents an instant snapshot of the view present in front of it. The webcam software sends each image frame to the computer for further processing. If the frame rate is more than 25 frames per second, then the set of all the images appears as a motion video.

Kinect

Microsoft's Kinect is a motion sensor input device used for the Xbox One and Xbox 360 gaming consoles and Windows PCs. It enables users to interface with the console computer without a gaming controller. Kinect provides a Natural User Interface (NUI) for interaction through movement of the body, gestures, and voice instructions.

In June 2011, the first iteration of the Kinect Software Development Kit (SDK) was launched.

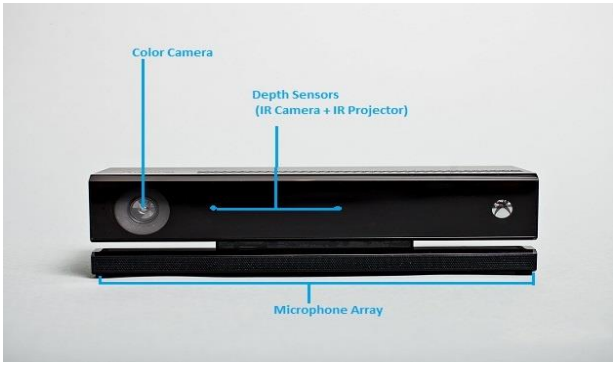


Figure 3.2: Microsoft Kinect

Kinect is a horizontal device that consists of a color camera, depth sensors, and a microphone array, as shown in Figure 3.2. All of these components are embedded inside a small and flat box with which a small motor is attached that helps the device to be tilted horizontally. The Kinect sensor provides data on image stream, depth stream, skeletal stream, and audio stream. The color sensor returns the image stream. On the other hand, the depth sensor produces the deep and skeletal streams, and the microphone array is used for returning the audio stream.

Challenges of using Kinect for SLRS

Building an application using Kinect is a significant challenge. Some of the challenges that are being faced by the researchers using Kinect are as follows.

i. Data smoothing: The quality of the raw depth data is pretty low, even with a maximum resolution of 320 x 240. The data noise occurs from the object's shadow and the light scattered by the object itself. So, smoothing of the raw data is required.
ii. Illumination Condition: The Kinect is designed for developing indoor applications only, and its performance degrades in outdoor applications.
iii. Frame Synchronization: It is impossible to synchronize frames from multiple Kinect devices. One can find the frame in two streams from two separate Kinect and compute the relative delta between the events.
iv. Interference is a significant challenge while tracking objects using multiple Kinect devices. Kinect measures the depth data by reading Infrared patterns, and when numerous sensors are projected in the same area, they start interfering with one another.

As discussed above, the usage of wearable computing-based acquisition devices leads to noise as wear and tear of the devices, hand and wrist movements, and poor calibration, which results in a low recognition rate. To overcome this, vision-based methods have been used. But the challenges still exist while using Kinect as a vision-based device like the noise in the captured data, lightning conditions, and interference by other Kinect devices. These drawbacks will affect the recognition rate of the system. So, to overcome this, a web camera is used for data acquisition. In this study, we have created two datasets

for identifying static and dynamic signs. The process followed for the development of static and dynamic signs for the proposed research work has been discussed in detail in the following section.

3.2 Procedure for preparing the dataset

Broadly, the signs are divided into two classes: static signs and dynamic signs. Static signs are those signs that do not have any movement and are static. On the other hand, dynamic signs are the signs with motion. In this research, two datasets have been developed and named as the dataset for static signs and the dataset for dynamic signs. The dataset for static signs consists of the collection of RGB images, and the dataset for dynamic signs consists of videos of different signs from the Indian sign language. The detailed procedure for preparing both datasets has been discussed below.

3.2.1 Subjects Participated in Dataset Preparation

The dataset has been collected for building the machine learning model. We have collected the dataset from participants of different age groups and from both genders whose details are given in Table 3.1. Each subject has performed the signs at an average of 10 times. All the collected signs are colored sign images/video clips with white background. The signs were collected from 35 healthy participants, including 20 males and 15 females.

Table 3.1: Summary of Subjects who participated in Dataset Elicitation

Age Group / **Gender**	**11-20**	**21-30**	**31-40**	**41-50**
Male	7	10	2	1
Female	2	9	4	
Total Participants	9	19	6	1
	35			

3.2.2 Camera Setup

All the signs were recorded with a DSC-WX150 SONY camera at different distances between the camera and the subject. The collected sign images captured for the static sign dataset had dimensions 640 x 480 pixels and were captured at the resolution of 350dpi. On the other hand, for dynamic signs, the camera records videos at 30 frames per second with a 1920 x 1080. The camera was placed on a tripod stand at the height of 1.5m from the ground. The distance between the subject and the camera varies from 1 m to 1.2 m. The experimental setup for collecting signs is shown in Figure 3.3.

3.2.3 Illumination Setup

Proper lighting is necessary while collecting signs for sign language recognition at a high frame. In both datasets, the signs were collected under different environmental conditions. The signs has been captured by varying the light intensity. We have recorded signs both under room light as well as under sunlight. Sign images collected under different environmental conditions are shown in Table 3.2.

All the sign images/videos were captured and collected from different participants, and they were asked to perform the sign naturally. As a result of natural performance, there are many variances between individuals while performing. The visual illustration of various sign variations done by multiple persons is presented in Appendix A Table 1.

Figure 3.3: Experimental Setup

Table 3.2: Sign Images under Different Environmental Conditions

Environmental Condition / Name of Sign	Room light	Natural Sunlight
Promise		
Two		
Bowl		
Five		
Pray		

Since the sign performed by the individual participant does not remain the same in all the situations and varies across time. Moreover, the intensity of each sign varies while

collecting other video clips of a participant with the same sign. The different ways of performing the same dynamic sign by the individual participant are shown in Appendix A Table 2.

3.3 Dataset Collection

In this work, two datasets were created by using several participants of Red Cross School for Deaf, Jalandhar and Patiala School for Deaf and Blind, Patiala. As discussed earlier, two datasets, one for static signs and another for dynamic sign recognition, have been collected. A detailed description of both datasets has been given in the following sections.

3.3.1 Dataset for Static Signs

The dataset for static signs consists of collecting the RGB images for different static signs. The dataset comprises 35,000 images, including 350 images for each static sign. The 100 distinct sign classes include 23 alphabets of English, 0-10 digits, and 66 commonly used words (e.g., bowl, water, stand, hand, fever, etc.). The dataset consists of static sign images of various sizes and colors taken under different environmental conditions. The list of specific signs captured by the participant is given in Appendix A Table 3.

Each sign has been assigned a different category based upon the single-handed signs, double-handed signs, signs with facial expression, single-handed signs with face, and double-handed signs with the face. The division of signs is shown in Appendix A Table 4.

3.3.2 Dataset for Dynamic Signs

Automatic recognition of dynamic signs is a significant challenge in affective computing. The rotation of the head, the posture of the face, illumination conditions, occlusions, etc., are the main attributes that lead to the increase in the complexity of the recognition of dynamic signs practically. The practical recognition of signs depends significantly on the quality of the database used.

In this research, a new database containing the videos of dynamic signs has been prepared. The dataset for dynamic signs consists of collecting the RGB video clips for

different signs. The dataset is collected from male and female participants of Indian origin. The dataset consists of 50 video clips of dynamic signs collected from 19 participants. Each participant is allowed to perform each sign on an average of 10. We collected 9,500 video clips of dynamic signs of Indian sign language. All the collected signs are colored video clips with a plain background. A few examples of video frames are provided in Appendix A Table 5.

The list of the dynamic signs collected for eliciting the dataset for dynamic signs is shown in Appendix A Table 6.

Chapter Summary

In this chapter, a brief study about wearable-based computing and vision-based devices has been presented. After this, we have discussed about the dataset collection and its development using Indian sign language. Both the datasets used for static and dynamic signs recognition were collected at different distances and created under various environmental conditions. It also describes how signs are categorized as single-handed signs, double-handed signs, signs with facial expressions, single-handed signs with faces, and double-handed signs with the face. This chapter has also documented a detailed description of the procedure of dataset preparation, camera setup, illumination setup, and subjects. The static and dynamic signs used for the recognition process have also been listed in this chapter with their pictorial representation.

CHAPTER 4

Data Pre-Processing

Data pre-processing is an important step in building a SLRS. This data pre-processing or data cleansing is crucial and requires a lot of effort before building a trained model for sign language recognition. As in this research, data has been collected in the form of images and videos, so pre-processing has been applied to the images and video frames at a very abstract level. Pre-processing aims to enhance the image and frame data by reducing unwanted distortions or improving attributes applicable to subsequent processing and analysis tasks.

This chapter focuses on the MediaPipe technique, used for data pre-processing on the given set of sign images and videos. MediaPipe provides cross-platform architecture for developing multimodal (e.g., video, audio, and series data) Machine Learning (ML) models. It solves numerous problems like face detection, face mesh, iris detection, hair segmentation, hand detection, pose detection, object detection, box tracking, etc. In this research, MediaPipe Hands, and MediaPipe Pose are used for tracking the hands and pose of a sign respectively.

4.1 Data Pre-Processing

The capacity to detect the shape and movement of hands may be a crucial factor in enhancing the user experience across several platforms, such as sign language recognition, gesture recognition, and augmented reality. Hand detection in real-time is challenging for computer vision since hands lack high contrast patterns and naturally obscure themselves.

In this research, we have used the MediaPipe python package to perform the pre-processing task. This package is available on PyPI for windows. This MediaPipe python package provides two solutions: MediaPipe hands and MediaPipe pose. MediaPipe hands are used for hand and finger tracking. On the other hand, MediaPipe pose offers a solution to machine learning for pose tracking.

Preprocessing using Mediapipe, involves preparing and enhancing the input data for sign language recognition. The use of MediaPipe for data pre-processing and feature extraction requires various steps that have been presented below.

i. Frame Extraction: In this step the frames are extracted from videos at regular intervals to create a dataset of individual images. This step allows you to process each frame independently for hand and pose detection.

ii. Image Resizing: this step is used to resize the captured frames or images while keeping the aspect ratio to a suitable scale, like 256x256 pixels. This process contributes to standardizing the MediaPipe models' input size.

iii. Normalization: in this the pixel value of the images should be normalized to a range of 0 to 1or -1 to 1.

iv. Input Format: in this the images are converted to RGB format since MediaPipe Hands and Pose models typically expect input in this format.

v. Hand-Pose detection: use MediaPipe hands to detect and track the hands in each frame and MediaPipe pose to estimate the body poses.

vi. Key point representation: MediaPipe hands provide 21 hand keypoints for each hand and total 42 keypoints for both the hands. Each keypoint is represented by a 3D coordinate (x, y, z), where (x, y) are the pixel coordinates in the image, and z represents the depth (distance) of the keypoint from the camera. On the other hand, Mediapipe Pose is used for human pose estimation, and it detects and tracks 33 body keypoints per person. These keypoints represent various body parts such as shoulders, elbows, wrists, hips, knees, ankles, etc. Similar to the hands, each keypoint is represented by a 3D coordinate (x, y, z).

vii. Sequence generation: this step is used to combine sequence of frames (image frames with detected hand keypoints) and form a video to feed into the recognition model.

viii. Label preparation: The labels in sign language for each frame or sequence should be added to your dataset as annotations. Make sure the labels correspond to the precise hand gesture shown in the video or image.

ix. One-hot encoding: one hot encoding is used to encode the class labels.

x. Data Loading: last step is to create a data loader that effectively loads the preprocessed data and sends it to the model for recognizing signs during training.

A detailed explanation of MediaPipe hands and MediaPipe posture has been discussed in the following subsection.

4.1.1 MediaPipe Hands

The MediaPipe hands use a machine learning pipeline with a two-stage approach. It consists of two ML models, including a palm detection model and a hand landmark model. A palm detection model has been applied to the whole picture, and a hand bounding box has been returned. It is only executed on the first frame or when a landmark signals a hand miss. On the other hand, a hand landmark model acts on the picture area clipped and determined by the palm detector. This model is going to return high-fidelity 3D hand key points [110]. MediaPipe hands are used for tracking the fingers and hand, inferring twenty-one 3D landmarks from a single frame, as presented in Figure 4.1.

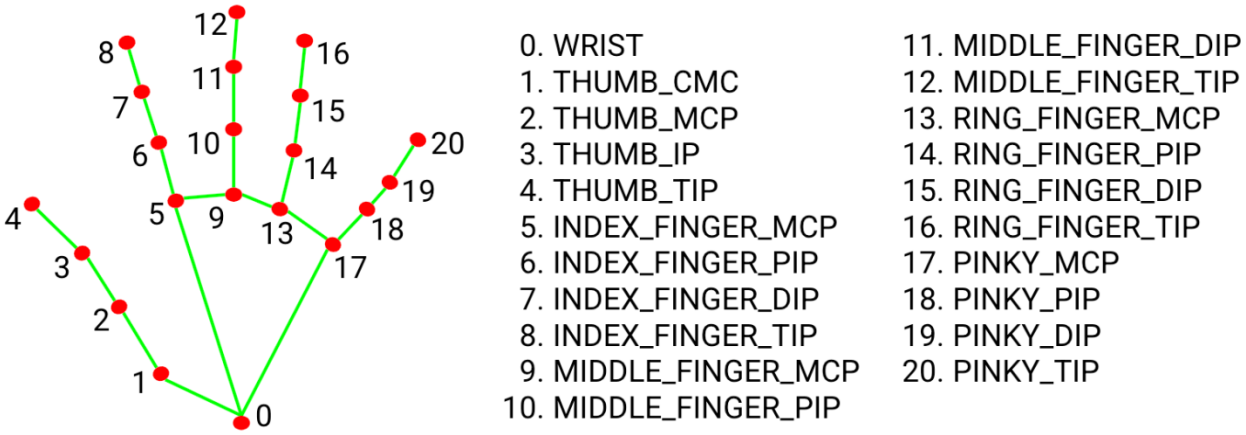


Figure 4.1: Twenty-One Hand Landmarks [110]

Normal hand anatomy consists of five fingers. Each finger is labeled with a Roman number, as shown below:

i. Digits-I or dig-I = the thumb

ii. Digits-II or dig-II = the forefinger

iii. Digits-III or dig-III = the middle finger

iv. Digits-IV or dig-IV = the ring finger

v. Digits-V or dig-V = the little finger

Multiple divisions between the hand and fingers, as shown in Figure 4.2.

- Carpometacarpal (CMC) joints: represents the articulation between the carpus and metacarpal bone components.
- Metacarpophalangeal (MCP) joints: represent the joint between the metacarpals and the proximal phalanges.
- Proximal interphalangeal (PIP) joints: show the articulation of the proximal and middle phalanges.
- Distal interphalangeal (DIP) joints: represent the joint between the proximal and middle phalanges.

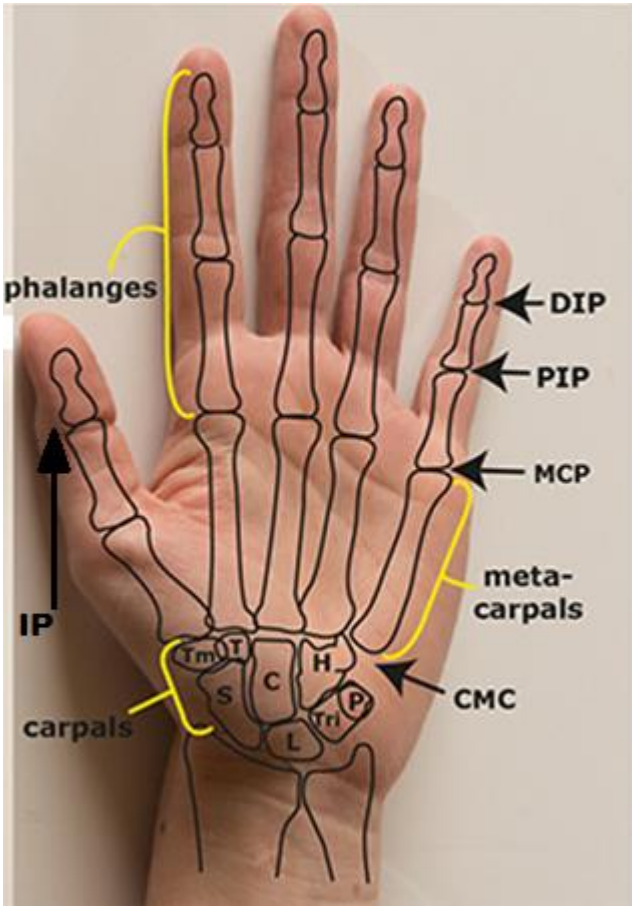


Figure 4.2: Normal anatomy of the hand. “CMC, MCP, PIP, DIP and IP” [108]

It has been shown in Figure 4.2 that dig I, i.e., the thumb has two joints that are MCP and IP. On the other hand, dig II-V has three joints: MCP, PIP, and DIP. The internal representation of a hand and fingers in real-time is shown in Figure 4.3.

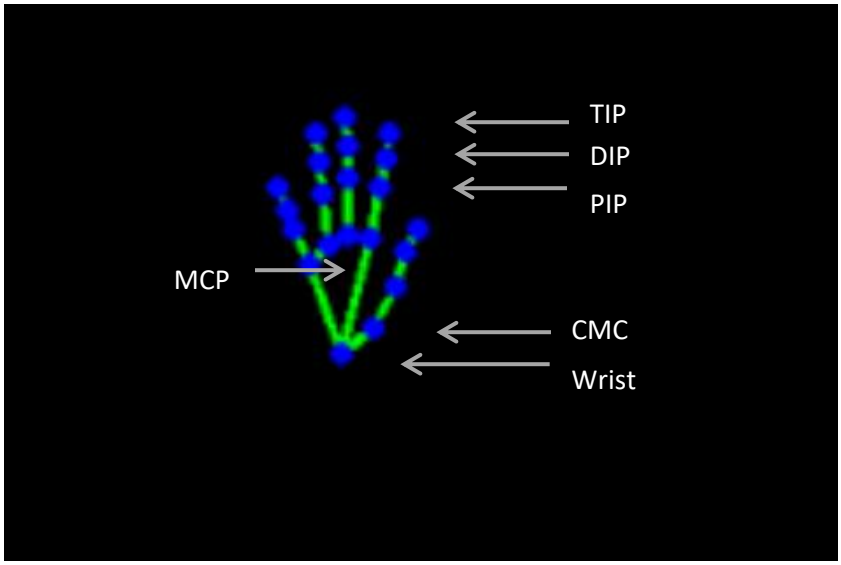


Figure 4.3: Internal Representations of Twenty-One Hand Landmarks at Real-Time

4.1.1.1 Palm Detection Model

The palm detection model involves the detection of both hands and fingers. The recognition of hands is a complex issue because the model must deal with hands of varying sizes and a high scale span (20x) relative to the image frame, as well as identify hands that are obscured by other hands. In contrast, the mouth and eye regions of human faces have patterns with high contrast. The absence of such characteristics in hands makes detection relatively challenging. Instead, using additional information like features of a person's arm and body leads to accurate hand localization [110].

The method used in this research addresses the challenges mentioned above using different strategies.

- First, a palm detector instead of a hand detector has been trained. Estimating the bounding boxes of fixed objects such as palms and fists is less complicated than identifying hands with articulated fingers.
- Second, an encoder-decoder feature extractor is employed to provide tiny objects with larger scene context-awareness.

- And last, the loss is minimized during the training process for supporting many anchors resulting from the high-scale variation.

4.1.1.2 Hand Landmark Model

After detecting palms, a hand landmark model has been imposed. In this model, the need for data augmentation (translation, scaling, and rotations) has been reduced to a greater extent by supplying the precisely cropped hand image as an input. It enables the network to devote most of its resources in coordinating prediction accuracy. Additionally, the ML pipeline can produce the crops depending on the hand landmarks discovered in the preceding frame [110]. The architecture of the hand landmark model is given in Figure 4.4.

The model processes the input image and outputs the coordinates of the detected hand landmarks. This architecture consists of two subparts- one is used for detecting hands and another one for hand landmark computation also called as keypoints [125]. These landmarks correspond to specific points on the hand, such as fingertips, knuckles, and the base of the palm. The direct prediction of coordinates was accomplished by conducting critical point localization of twenty-one 3D hand coordinates inside the observed hand areas. The perception pipeline can be created using MediaPipe as a directed graph of Calculators, which are modular building blocks. These calculators are used for image cropping, rendering and neural network computations on GPU as shown in Figure 4.4. The high-quality synthetic hand model has been rendered onto multiple backgrounds and mapped to the associated 3D coordinates to provide more coverage of hand positions and further oversight on the nature of hand geometry.

As explained in Chapter 3, the signs are divided into five different classes based upon single-handed signs, double-handed signs, signs with facial expressions, single-handed sign with faces, and double-handed sign with faces; the output is obtained by applying the hand landmark model on two different classes of sign images using only hands is shown in Table 4.1. The pre-processed sign images obtained for single-handed and double-handed signs 'one', ‘chest’, ‘bent’ and ‘Bottle’, ‘A’, ‘Skin’ after applying the MediaPipe hand technique are shown in Table 4.1, respectively.

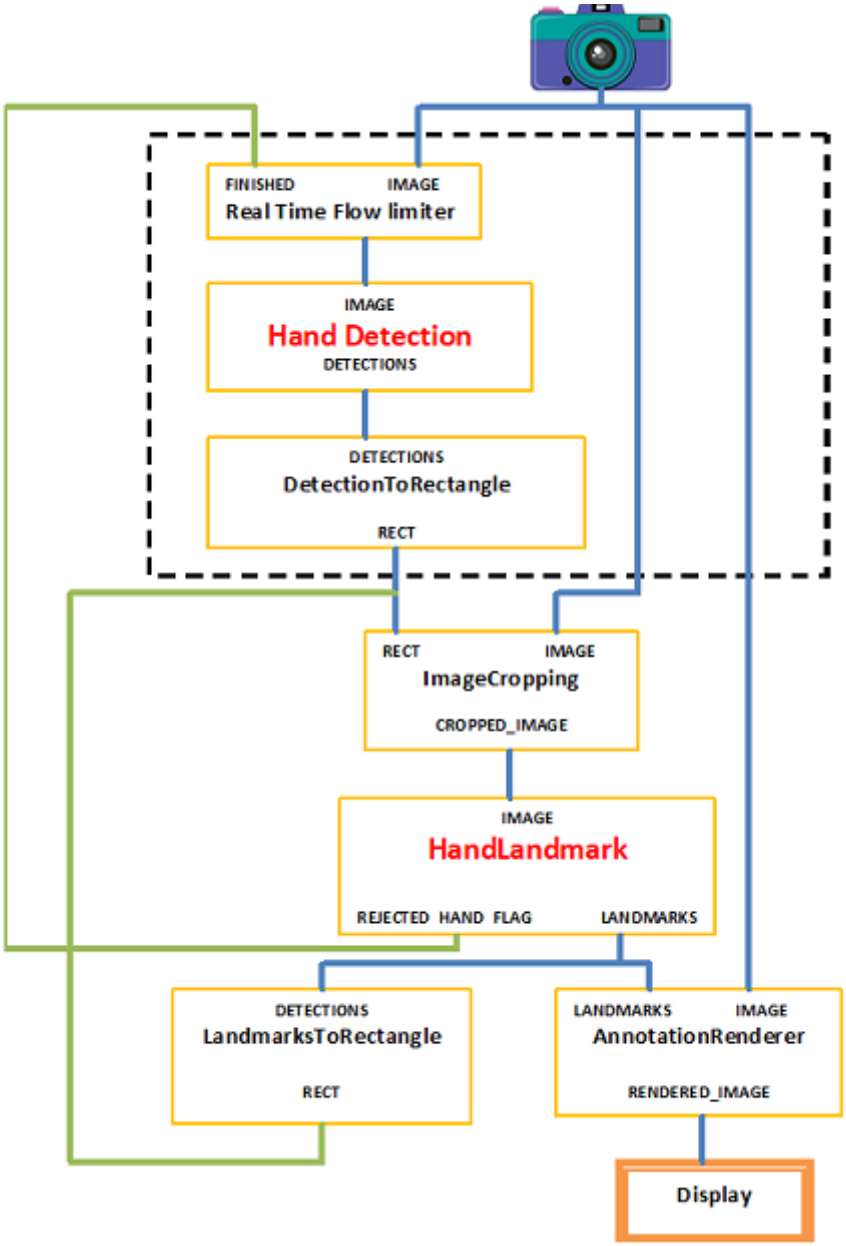
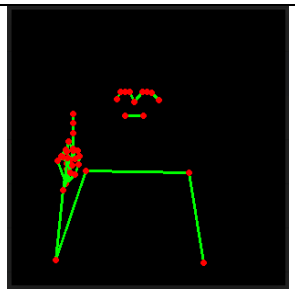
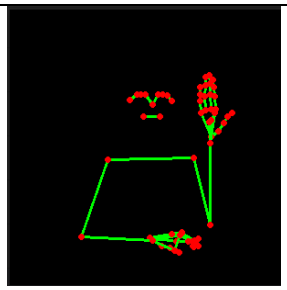
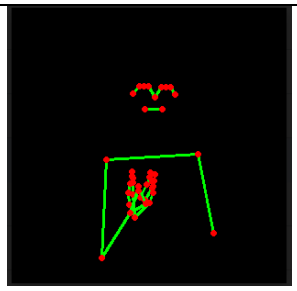
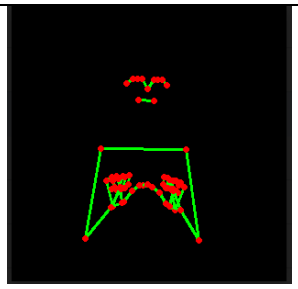
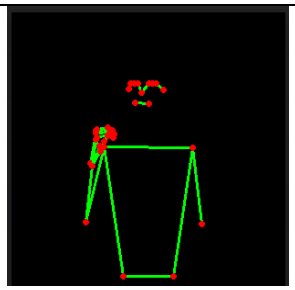
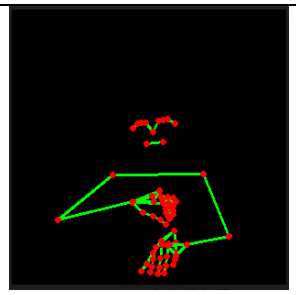


Figure 4.4: Architecture of Hand Landmark Model [125]

Table 4.1: Pre-processed Images after Hand Landmark Model

Type of Sign	Single-handed Signs	Double Handed Signs
Pre-Processed image		
Name of Sign	One	Bottle
Pre-Processed image		
Name of Sign	Chest	A
Pre-Processed image		
Name of Sign	Bent	Skin

4.1.1.3 Solution APIs

API is the Application Programming Interface for providing the connectivity between the real-time application and the server. It is used for connecting and sending data to a server. The server then receives the data, processes it, and executes the required actions before returning it to the user. The software then analyses the data and provides the user with the desired information. APIs used for delivering the solution to MediaPipe hands are discussed as follows.

- STATIC_IMAGE_MODE: The research has been performed on the static images, so the value of STATIC_IMAGE_MODE is true. This API aims to detect hands and localize all the hand landmarks after successful detection.
- MAX_NUM_HANDS: It defines the maximum number of identifiable hands. By default, it is set to 2.
- MIN_DETECTION_CONFIDENCE: Minimal confidence value of [0.0, 1.0] for the detection of hands to be considered from the hand detection model. By default it is [0.5].

4.1.1.4 Output

The output of each sign is represented by MULTI_HAND_LANDMARKS in which, each hand is represented by a list of twenty-one hand landmarks, and each landmark consists of x, y, and z. The x and y denote the breadth and height, while z denotes the depth and the wrist serves as the origin.

4.1.2 MediaPipe Pose

Pose estimation in humans plays a crucial role in number of applications like sign language recognition, controlling the gestures of the whole body, and physical exercises. It can be used in dance, yoga, and other fitness applications. MediaPipe pose provides the solution for tracking high-fidelity body pose. It infers thirty-three 2D landmarks on the full-body, including twenty-five upper body landmarks, as shown in Figure 4.5. This solution enables real-time performance on mobile phones, laptops, desktops, and even the web, while conventional methods for inference depend primarily on robust desktop systems [198].

MediaPipe pose uses a two-step detector-tracker method. Firstly, the person's pose region of interest (ROI) is located within a frame using a detector. After this, the pose landmarks within the ROI using the ROI cropped frame as input are predicted subsequently by the tracker. The sensor has been invoked only when it is required for video use cases, i.e., for the very first frame and when the tracker was unable to recognize the existence of body position in the preceding frame. For subsequent frames, the ML process derives the ROI

from the prior frame's pose landmark. The pre-processed image with thirty-three landmarks of MediaPipe is shown in Figure 4.6.

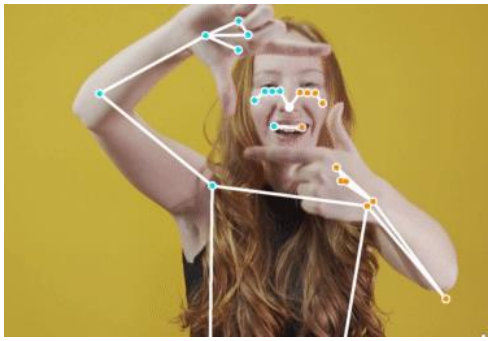


Figure 4.5: MediaPipe Pose for Upper Body Pose Tracking [198]

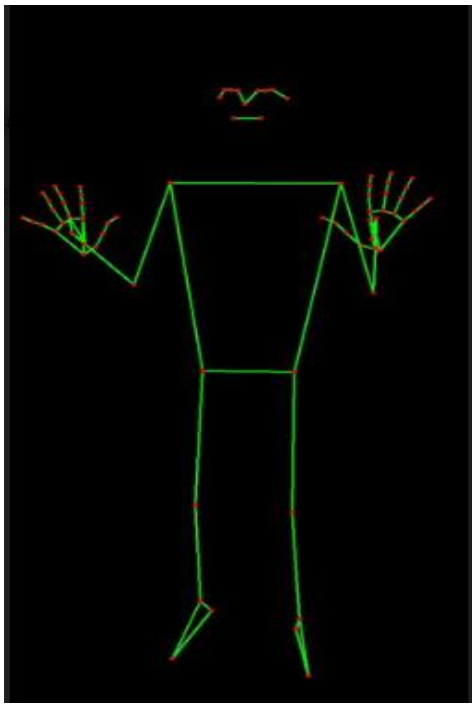


Figure 4.6: Internal Representations of Pose Landmarks at Real-Time

The media pipe pose consists of two models that work together to detect and track different poses. It uses person/pose detection and landmark models for detecting and monitoring postures.

4.1.2.1 Person/Pose Detection Model (BlazePose Detector)

The pose detection model, also called BlazePose Detector, predicts two additional virtual key points, describing the center of the human body, rotation, and a scale as a circle. The midpoint of a person's hips, the radius of a circle circumscribing the whole body, and the inclination angle of the line connecting the shoulder and hip midpoints are projected using Leonardo's Vitruvian man as inspiration. Vitruvian man aligned using two estimated virtual key-points using BlazePose detector is shown in Figure 4.7.

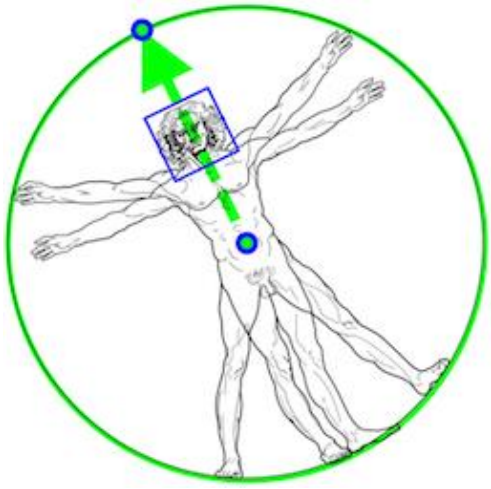
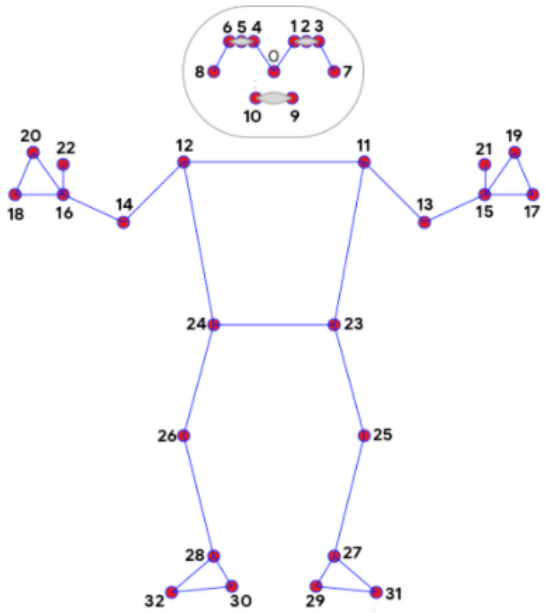


Figure 4.7: Vitruvian man aligned via two virtual key-points [198]

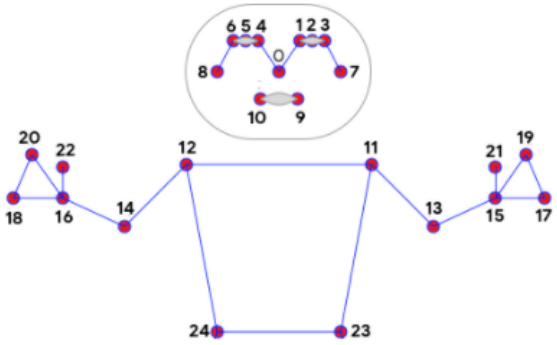
4.1.2.2 Pose Landmark Model (BlazePose Tracker)

The MediaPipe posture landmark model has two variations. The first is a full-body model that predicts the position of 33 posture markers, as shown in Figure 5.8. The second one is an upper-body model used for predicting only the first 25 locations of the body, as shown in Figure 5.9. The second version performs better than the first one in situations where lower body parts are mostly hidden from the view [198].



0. nose
1. left_eye_inner
2. left_eye
3. left_eye_outer
4. right_eye_inner
5. right_eye
6. right_eye_outer
7. left_ear
8. right_ear
9. mouth_left
10. mouth_right
11. left_shoulder
12. right_shoulder
13. left_elbow
14. right_elbow
15. left_wrist
16. right_wrist
17. left_pinky
18. right_pinky
19. left_index
20. right_index
21. left_thumb
22. right_thumb
23. left_hip
24. right_hip
25. left_knee
26. right_knee
27. left_ankle
28. right_ankle
29. left_heel
30. right_heel
31. left_foot_index
32. right_foot_index

Figure 4.8: Thirty-Three Pose Landmarks [198]

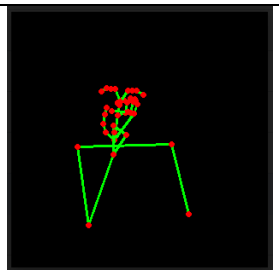
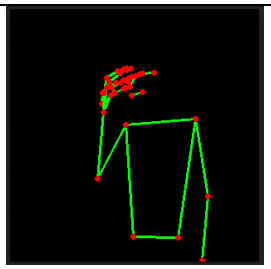
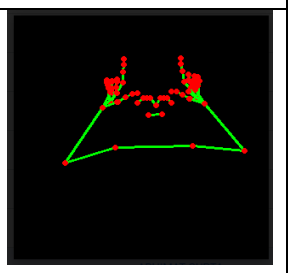
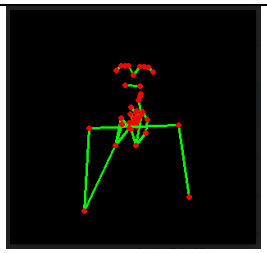
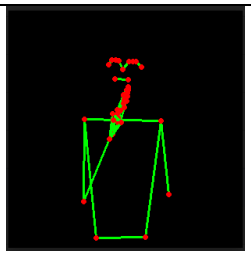
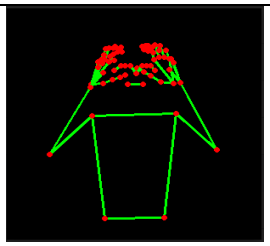


0. nose
1. left_eye_inner
2. left_eye
3. left_eye_outer
4. right_eye_inner
5. right_eye
6. right_eye_outer
7. left_ear
8. right_ear
9. mouth_left
10. mouth_right
11. left_shoulder
12. right_shoulder
13. left_elbow
14. right_elbow
15. left_wrist
16. right_wrist
17. left_pinky
18. right_pinky
19. left_index
20. right_index
21. left_thumb
22. right_thumb
23. left_hip
24. right_hip

Figure 4.9: Twenty-Five Upper Body Landmarks [198]

The output obtained by applying the pose landmark model to three different classes of sign images for signs 'water',’cough’,’food’, 'trouble', ‘owl’ and 'cow' is shown in Table 4.2.

Table 4.2: Pre-processed Images after Pose Landmark Model

Type of Sign	Signs with Facial Expression	Single-Handed signs with face	Double Handed signs with face
Pre-Processed image			
Name of Sign	Water	Trouble	Cow
Pre-Processed image			
Name of Sign	Cough	Food	Owl

4.1.2.3 Solution APIs

APIs used for providing the solution to MediaPipe pose are discussed as follows.

STATIC_IMAGE_MODE: This mode will identify the most notable individual in the initial image, and STATIC_IMAGE_MODE is set to true.

UPPER_BODY_ONLY: It is set to false and outputs the whole set of 33 pose landmarks. If set to true, the solution will only output the 25 upper-body position landmarks.

MIN_DETECTION_CONFIDENCE: For this solution, the minimum confidence value from the pose detection value ranges from [0.0, 1.0], and it is set to 0.5.

4.1.2.4 Output

The output is represented by POSE_LANDMARKS which contains the value of x, y, z, and visibility. Where,

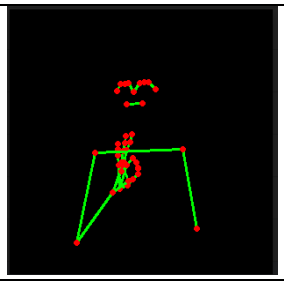
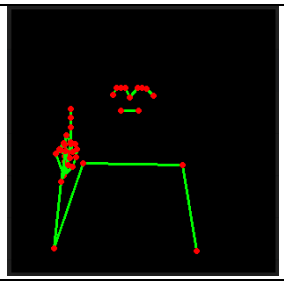
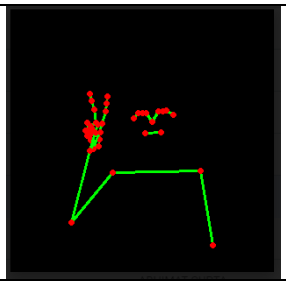
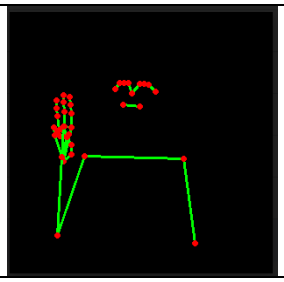
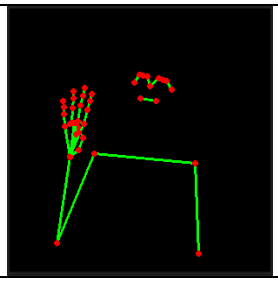
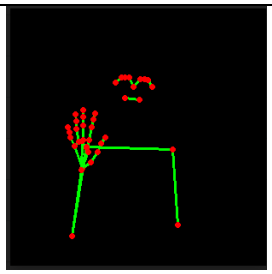
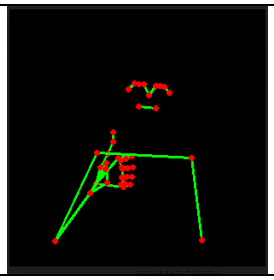
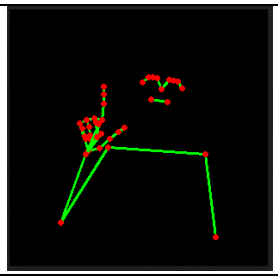
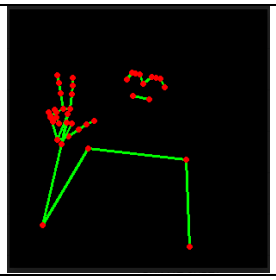
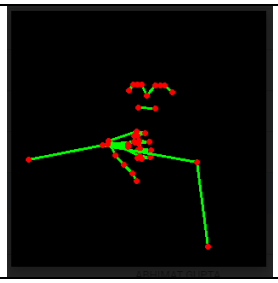
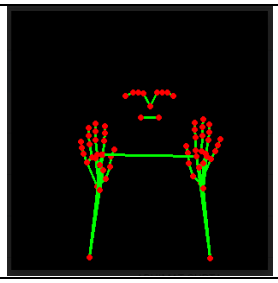
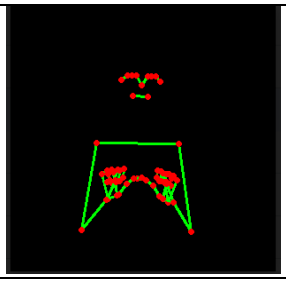
x and y: Pose landmark coordinates are normalized to the range [0.0, 1.0] using the image width and height.

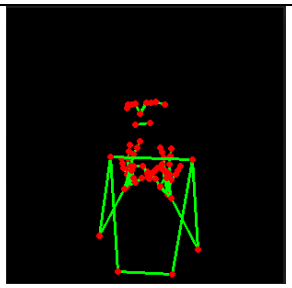
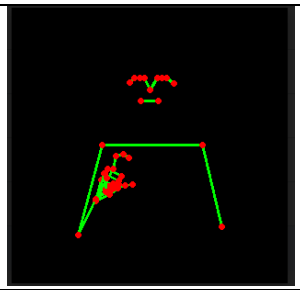
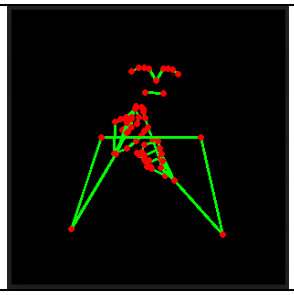
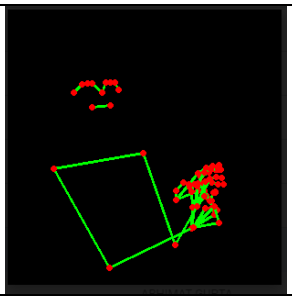
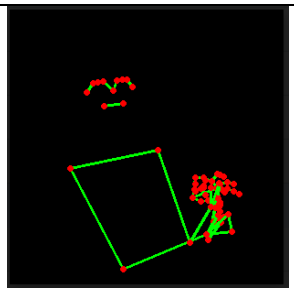
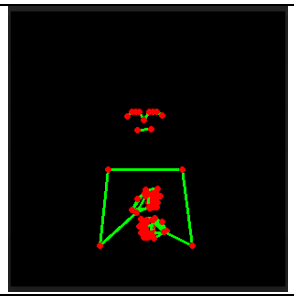
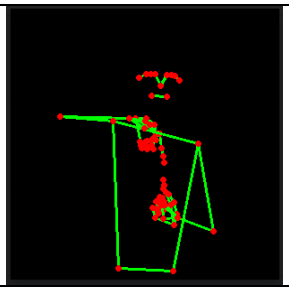
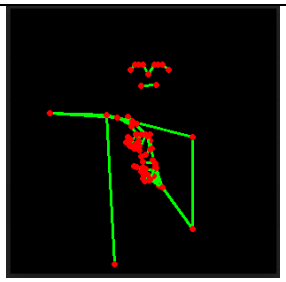
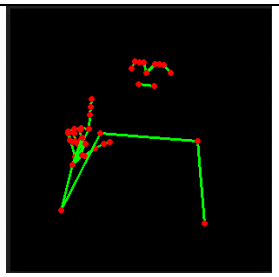
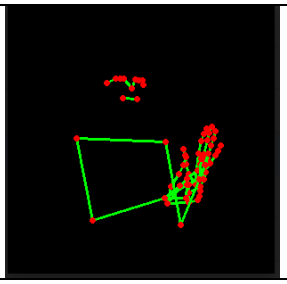
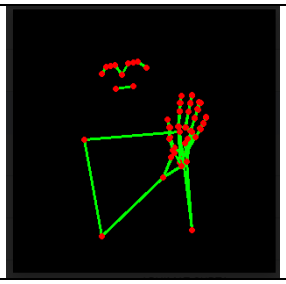
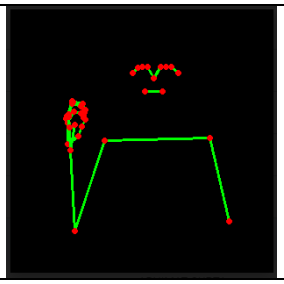
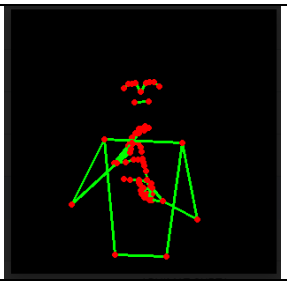
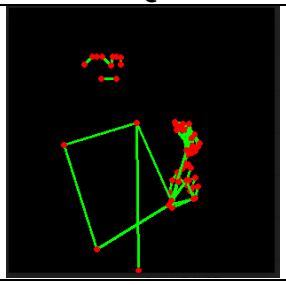
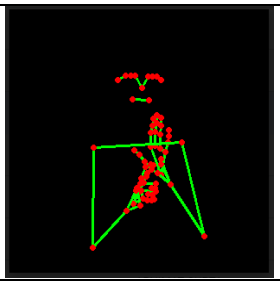
z: should be rejected because the model has not been trained to predict depth.

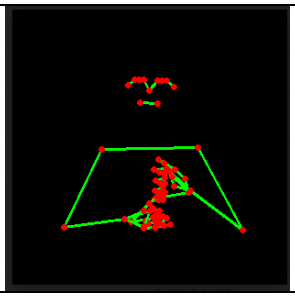
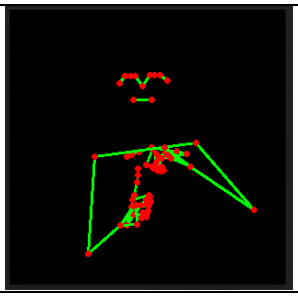
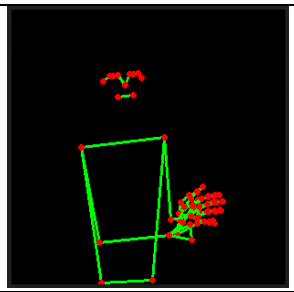
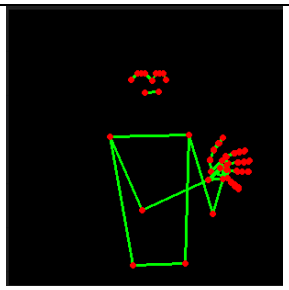
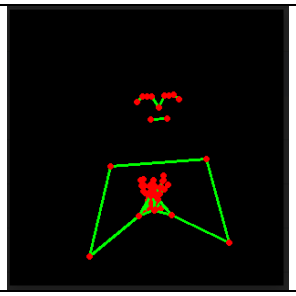
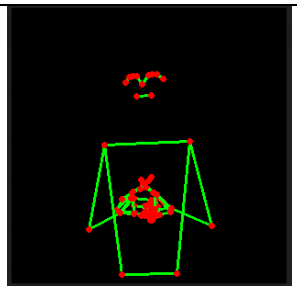
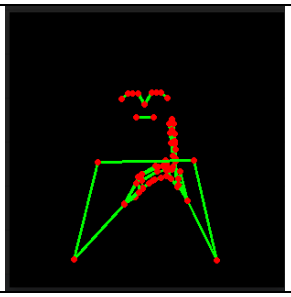
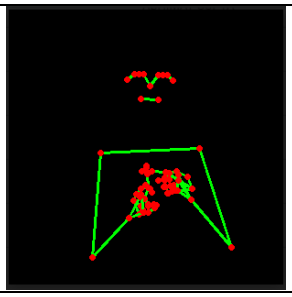
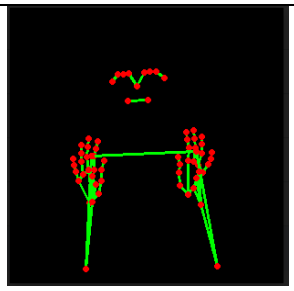
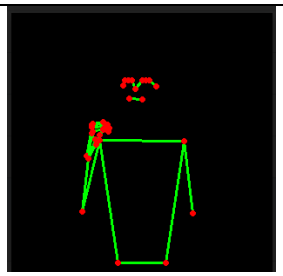
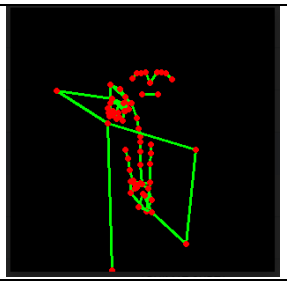
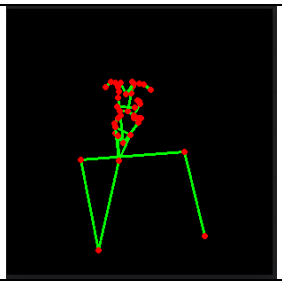
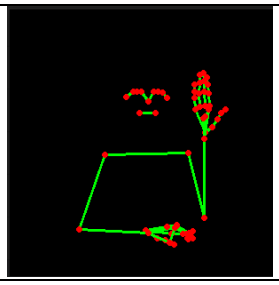
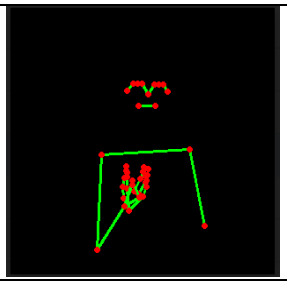
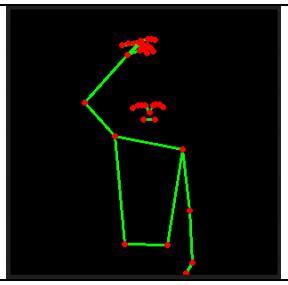
Visibility: It ranges from [0.0 to 1.0] and represents the chance that the landmark is visible in the image.

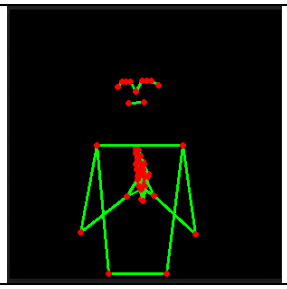
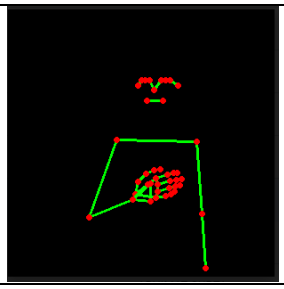
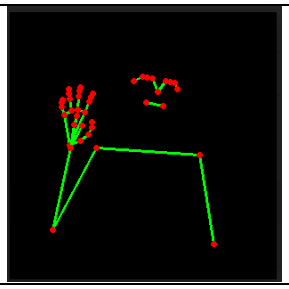
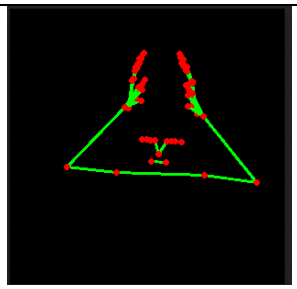
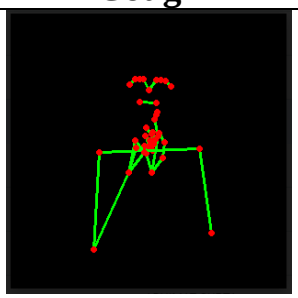
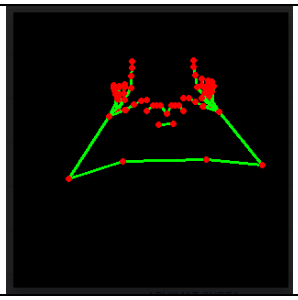
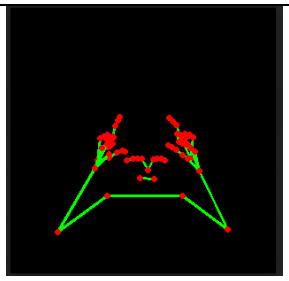
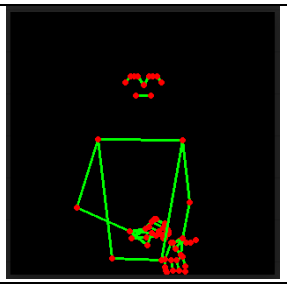
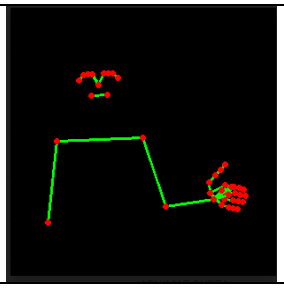
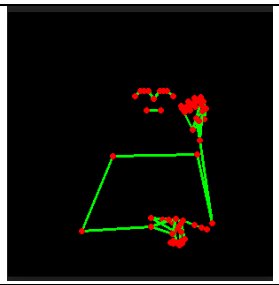
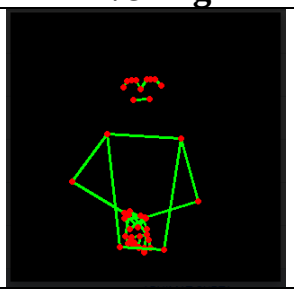
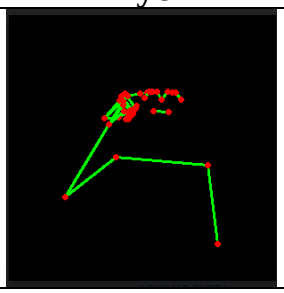
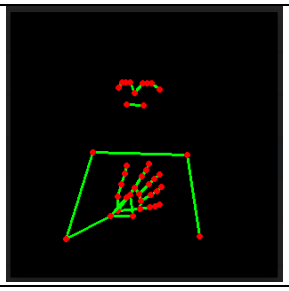
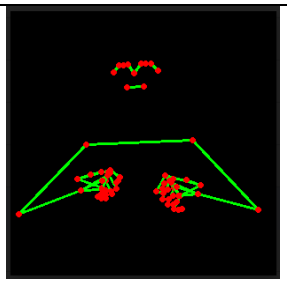
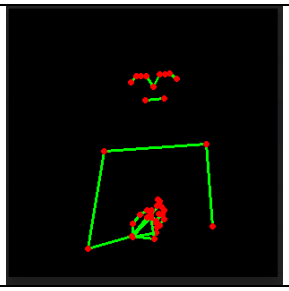
The pre-processed images were obtained after applying MediaPipe hands and MediaPipe Pose techniques, as presented in Table 4.3.

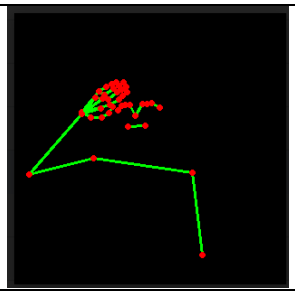
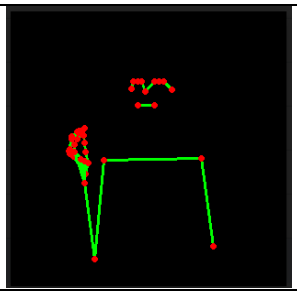
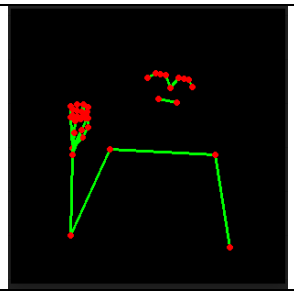
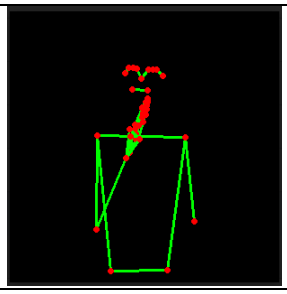
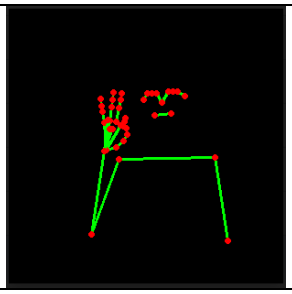
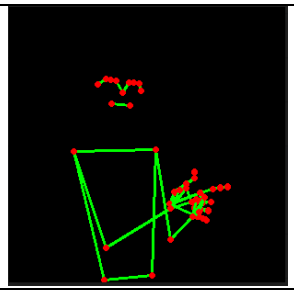
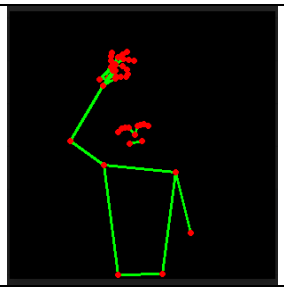
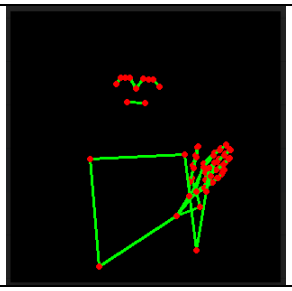
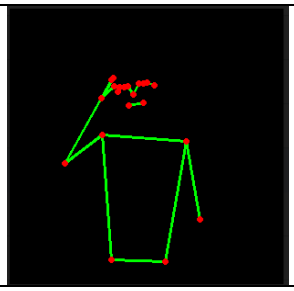
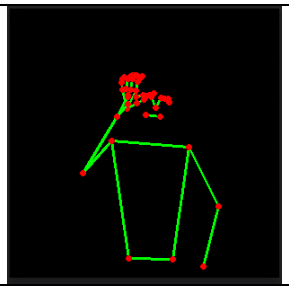
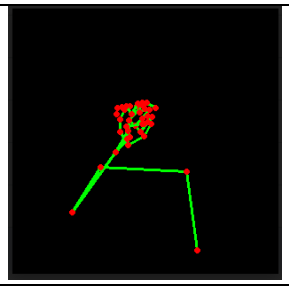
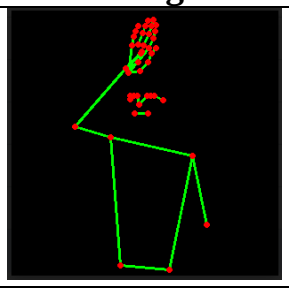
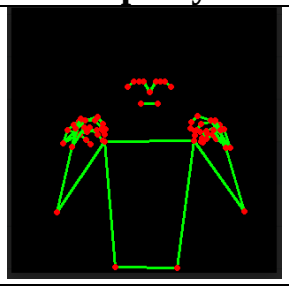
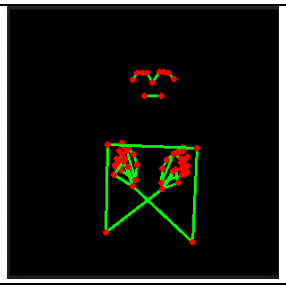
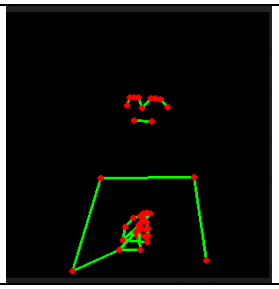
Table 4.3: List of Pre-Processed Static Sign Images

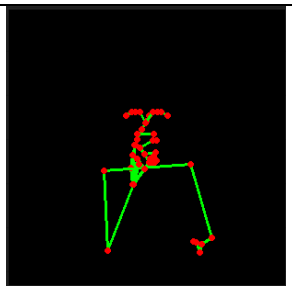
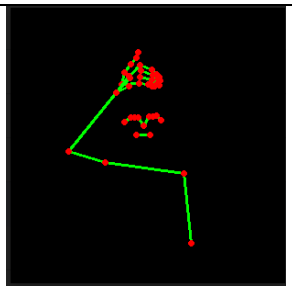
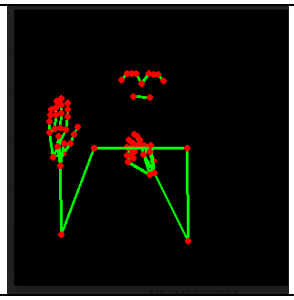
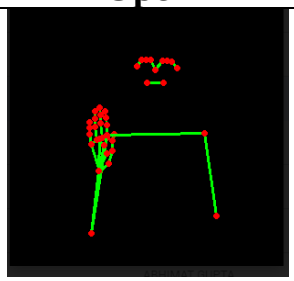
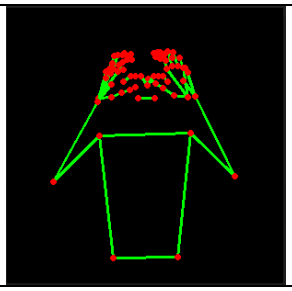
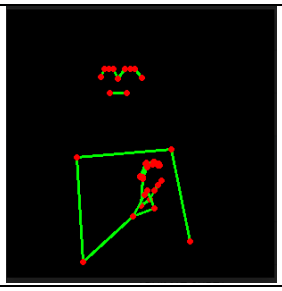
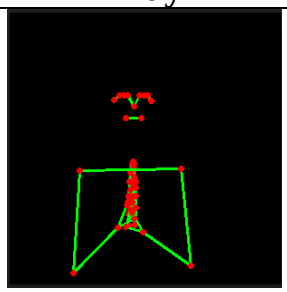
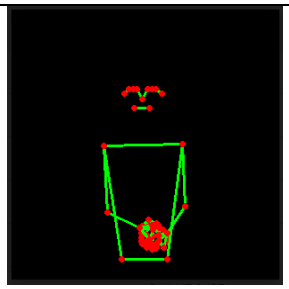
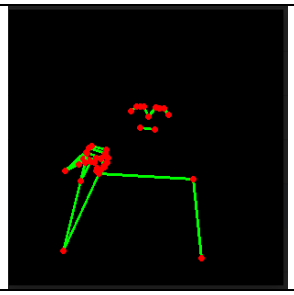
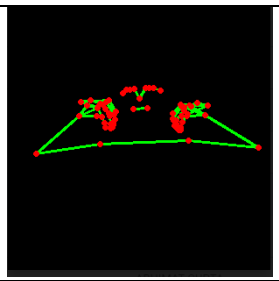
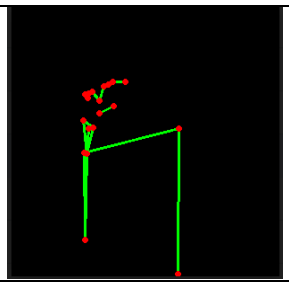
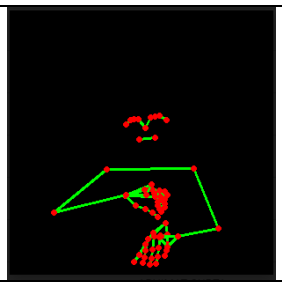
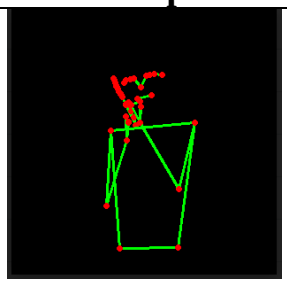
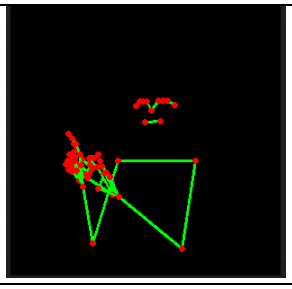
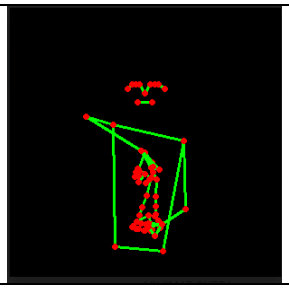
Sample Sign Image		
0	**1**	**2**
		
3	**4**	**5**
		
6	**7**	**8**
		
9	**10**	**A**
		

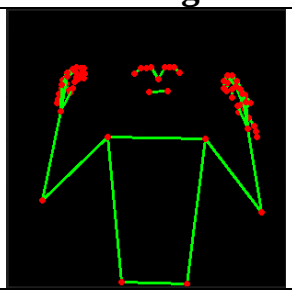
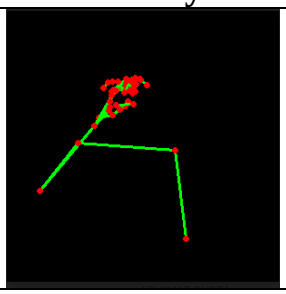
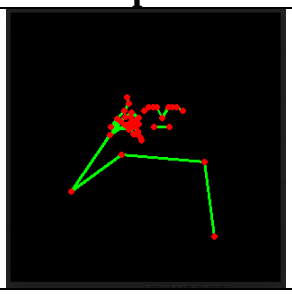
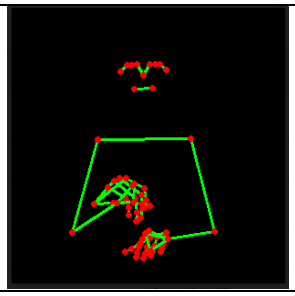
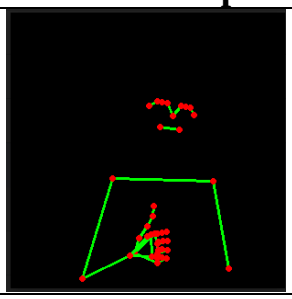
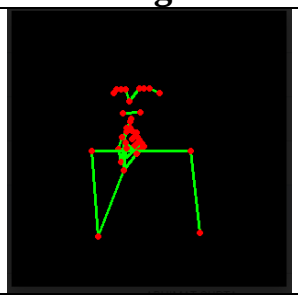
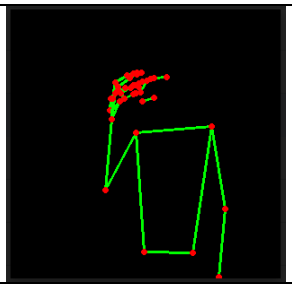
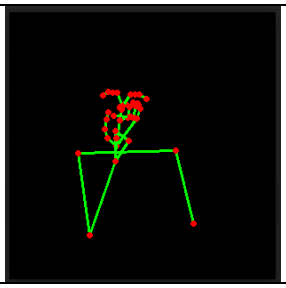
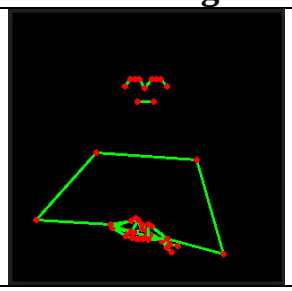
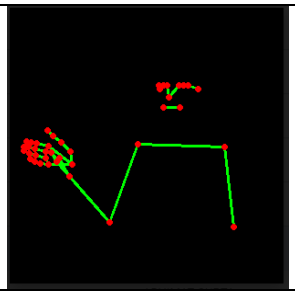
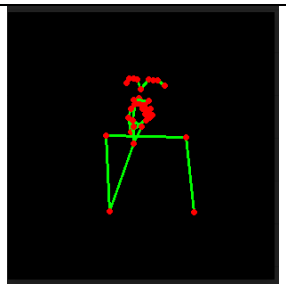
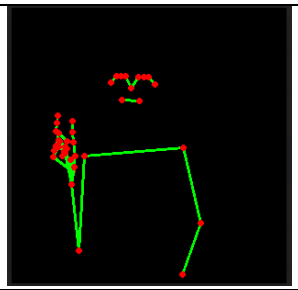
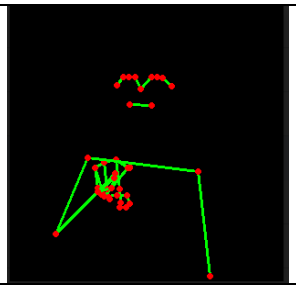
B	C	D
		
E	**F**	**G**
		
I	**K**	**L**
		
M	**N**	**O**
		
P	**Q**	**R**
		

S	T	U
		
V	**W**	**X**
		
Z	**Add**	**Afraid**
		
Bent	**Between**	**Blind**
		
Bottle	**Bowl**	**Brain**
		

Bud	Chest	Claw
		
Coolie	**Cough**	**Cow**
		
Devil	**Doctor**	**East**
		
Elbow	**Evening**	**Eye**
		
Faith	**Fat**	**Feel**
		

Fever	Few	Fist
		
Food	**Good**	**Gun**
		
Hair	**Hand**	**Head**
		
Hear	**Jain**	**King**
		
Leprosy	**Love**	**Me**
		

Nose	Nurse	Oath
		
Open	**Owl**	**Police**
		
Pray	**Promise**	**Shirt**
		
Shoulder	**Sick**	**Skin**
		
Sleep	**Soldier**	**Stand**
		

Strong	Sunday	Telephone
		
Thorn	**Thumbs Up**	**Tongue**
		
Trouble	**Water**	**Wedding**
		
West	**White**	**Word**
		
You		
		

4.2 Strengths of MediaPipe in Data Preprocessing

In comparison to alternative feature extraction methods, using MediaPipe for sign language identification can have a number of benefits, especially where gestures and visual signals are important. Using MediaPipe particularly for sign language recognition has the following benefits.

(i) Hand tracking and gesture recognition: For the purposes of hand tracking and gesture recognition, MediaPipe offers pre-built components. This is essential for accurately recording the complex hand gestures and shapes that are essential for communication in sign language.

(ii) Real-Time performance: Real-time processing tends to be necessary for sign language recognition, particularly for applications like live interpretation. The MediaPipe pipeline ensures minimal delay in identifying and interpreting signs despite having the ability to handle real-time conditions.

(iii) Multi-modal integration: Along with hand gestures, sign language also uses body language, facial expressions, and postures. One can record these non-manual signals using MediaPipe's multi-modal capabilities, which improves the precision and overall context of sign identification.

(iv) Deep Learning Integration: Deep learning models can be integrated into MediaPipe's pipelines, enabling more advanced and adaptive feature extraction. This is beneficial for capturing complex relationships between different signs and their variations.

(v) Pre-trained models: MediaPipe provides pre-trained models for hand tracking. By using these models, one can save the time necessary to create new models from scratch.

(vi) Cross Platform Compatibility: MediaPipe supports multiple platforms, which allow the deployment of sign language recognition applications on various devices, from desktop computers to smartphones and embedded systems.

(vii) Customization and adaptation: MediaPipe's modular architecture enables you to customize and adapt its components to the specific signing style or sign language dialect you're working with. This is crucial for building accurate recognition systems that cater to the diversity of sign languages.

Chapter Summary

This chapter has documented the detailed information of the models used for detecting and tracking hands and various poses. Initially, in this chapter, pre-processing has been explained for processing the collected data for static signs. The detailed description of MediaPipe hands and MediaPipe pose models is also documented. It also aims to explain the solution APIs used for the same.

In the proposed system, MediaPipe hands are used as palm detection and hand landmark model. The palm detection model used here tracks the hand and fingers together from the given image. On the other hand, the hand landmark model is used for a specific region cropped and defined by the palm detector model.

This chapter uses the MediaPipe pose as a two-step detector-tracker-based ML pipeline to detect and track human poses. This model uses a Blazepose detector and Blazepose tracker as a two-step detector-tracker. It also explains all the solution APIs used for getting the output from all the models used for pre-processing. The pre-processed sign images obtained after applying both MediaPipe hands and the MediaPipe pose model on all the 100 signs are also documented in this chapter.

CHAPTER 5

Model Training and Testing

After successfully pre-processing the data, the CNN-based model has been trained for the sign language classification system. The classifier takes the pre-processed sign images and video frames and classifies them into the corresponding sign. The classifier is trained on the dataset for static and dynamic signs containing different ISL signs. The dataset is shuffled and split into training and test sets.

This chapter describes the overall design and implementation of a SLRS to detect signs. The proposed architecture of a SLRS that identifies static and dynamic signs has been presented. We have introduced three different Convolutional Neural Network (CNN) based models. The first model is the fundamental CNN model, the second model uses MediaPipe for pre-processing and CNN to recognize static signs, and the third uses MediaPipe and CNN to recognize dynamic signs. A brief description of each of the architecture components and phase-wise working of SLRS has been explained in this chapter.

This chapter also provides a detailed description of the experimentation performed and the results produced on both datasets. The experiments have also been performed using VGG16, VGG19, and GoogleNet on static sign images and dynamic signs of video clips. Various performance metrics, like accuracy, precision, recall, and f-measure, have been used to evaluate the results generated by each CNN model.

5.1 Generalized Architecture of CNN

The objective of CNN is to learn the features present in the data by using convolutions. The CNN architecture works well for the recognition of objects which includes images. They can identify individuals, faces, street signs, and other facets of visual data. CNN architecture consists of different components with different types of layers and activation functions. The graphical representation of CNN architecture is given in Figure 5.1.

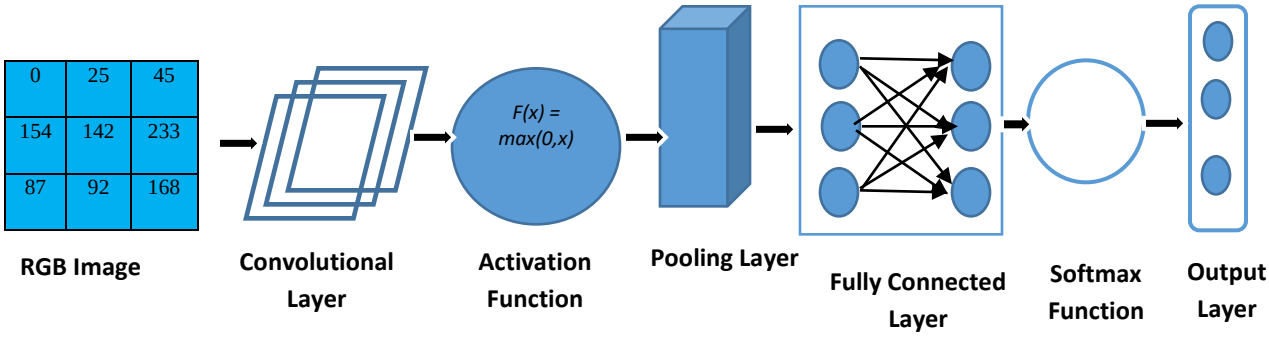


Figure 5.1: Graphical Representation of CNN Architecture

The high-level CNN architecture is shown in Figure 5.2. The listing describes the purpose and functioning of some commonly used layers, which are discussed below.

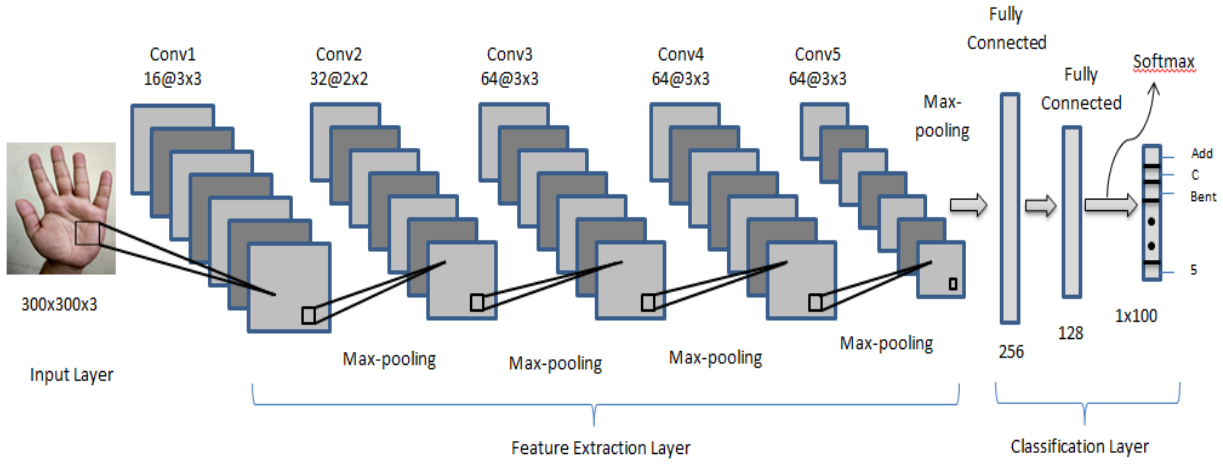


Figure 5.2: High-level generalized CNN Architecture

Convolutional Layer: The main building block of CNN architecture is the convolutional layer. This layer is used to extract different features from the image and the output retrieved is called as feature map. This feature map provides the information about edges and corners available in the input image as shown in Figure 5.3.

Convolution is a mathematical operation performed between the input image and a specific size filter with size (F x F). This operation demonstrates the sliding of the kernel across the input data, and then the dot product is computed between the input image and the filter w.r.t the filter size (F x F) [135].

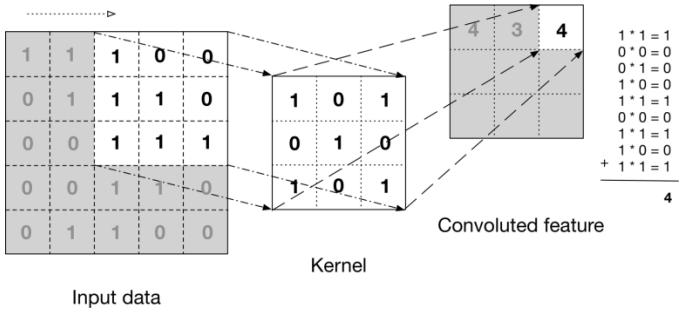


Figure 5.3: The Convolution Operation

Let us suppose the frame size of an input image $W \in R^{wxh}$. The convolutional filter with size F is used for convolution with a stride of S and P padding for the input image boundary. The size of the output of the convolution layer is presented by equation (5.1).

$$Output = \frac{W-F+2P}{S} + 1 \tag{5.1}$$

For example, there is one neuron with a receptive field size of F = 5, the input size is W = 128, and the padding of P = 1. The stride of the neuron across the input in stride of S = 1, giving output with size (128 - 5 + 2)/1+1 = 126.

Pooling Layer: Pooling layers are used for dimensionality reduction that reduces the progressive representation of data throughout the network and prevent overfitting. The pooling layer performs independently on each depth slice of the input. The main objective of this layer is to reduce the size of feature map which further reduces the computational costs. Pooling can be divided into two categories as max pooling and average pooling. Max pooling selects the largest element from the feature map whereas average pooling compute the average of the elements from an image.

Fully Connected Layer: A fully connected layer is utilized to generate scores for various classification features. The output layer has the dimensions [1 x 1 x N], where N is the number of output classes to be assessed. Each output neuron is linked to all other neurons in the preceding layer with varying weights.

As sign language recognition is a multiclassification issue, the output classification layer uses the softmax function. Finally, the class scores are computed using the final layer of 100 completely interconnected neurons. Here, the total number of classes inside the dataset is 100.

Activation Function: This layer decides whether the input of the neuron is important or not to predict the process. Several activation functions exists such as Softmax, ReLU (Rectified Linear Unit), tanH and sigmoid. Softmax function is applicable for multiclass classification. The sigmoid function and hyperbolic tangent are some other activation functions that can also be used to influence non-linearity in the network. The usage of ReLU is preferred because the derivative of the function helps backpropagation work considerably faster without making any noticeable difference to generalization accuracy [168].

Dropout: Usually, overfitting in the training dataset might result from all features being connected to the FC layer. Overfitting occurs when a specific model performs so well on training data but it is not able to give that much performance on unseen data or test data. To solve this problem, a dropout layer is used, in which few neurons are dropped off from the network during training process, which further results in reduced size of the model.

5.2 Different Optimizers used for Model Training

An optimizer is a function used in neural networks to optimize the weights and learning rate parameters to improve the model's accuracy and minimize loss value. Optimizers enable efficient training of deep neural networks, accelerate convergence, enhance robustness, and handle large datasets [190].

Although there exist several optimizers, in this study following five optimizers are considered.

i. Stochastic Gradient Descent (SGD): It works by taking small batches of the dataset from the whole set to minimize the overall computation cost. It is used to optimize the weight and loss value till it reaches local minima.

ii. RMSProp (Root Mean Square): RMSProp penalizes the parameters which oscillate the cost function. So it prevents the model from quickly adapting the changes in a single feature compared to other features. Hence, RMSProp increases the contribution of other features for making decisions.

iii. Adagrad (Adaptive Gradient Descent): It is used to adjust learning rates for each parameter during training. Adagrad is effective in settings where different

parameters have vastly different gradients or when dealing with sparse data. It can automatically adapt the learning rates to each parameter's behavior, which often results in faster convergence and improved optimization performance compared to traditional gradient descent methods.

iv. Adadelta: It is an extension of the Adagrad algorithm, designed to address some of its limitations, particularly the issue of learning rate decay. Adadelta is known for its ability to adaptively adjust learning rates during training, which can be particularly useful for complex and high-dimensional optimization problems [198].

v. Adam (Adaptive Moment Estimation): It is inherited from the Adagrad and RMSProp optimizer. In this learning rate parameter is updated for every network weight individually. The significant advantage of Adam is the fast convergence rate.

5.3 System Flow

The objective of CNN is to learn the features present in the data with higher-order using convolutions. The SLRS architecture consists of four primary phases: data collecting, image pre-processing, CNN classifier training, and testing. Figure 5.4 depicts the data flow diagram illustrating the system's operational model. The first step is the data collection phase, in which RGB data of static and dynamic signs has been collected by using the camera. The gathered sign images and video frames are then pre-processed utilizing image scaling, image normalization, and MediaPipe. The pre-processed data is subsequently saved for future use in the data storage. In the subsequent step, the proposed system is trained using a CNN classifier, and the learned model is then used for testing. The last phase is the testing phase, in which the CNN architecture parameters are fine-tuned until the results match the desired accuracy.

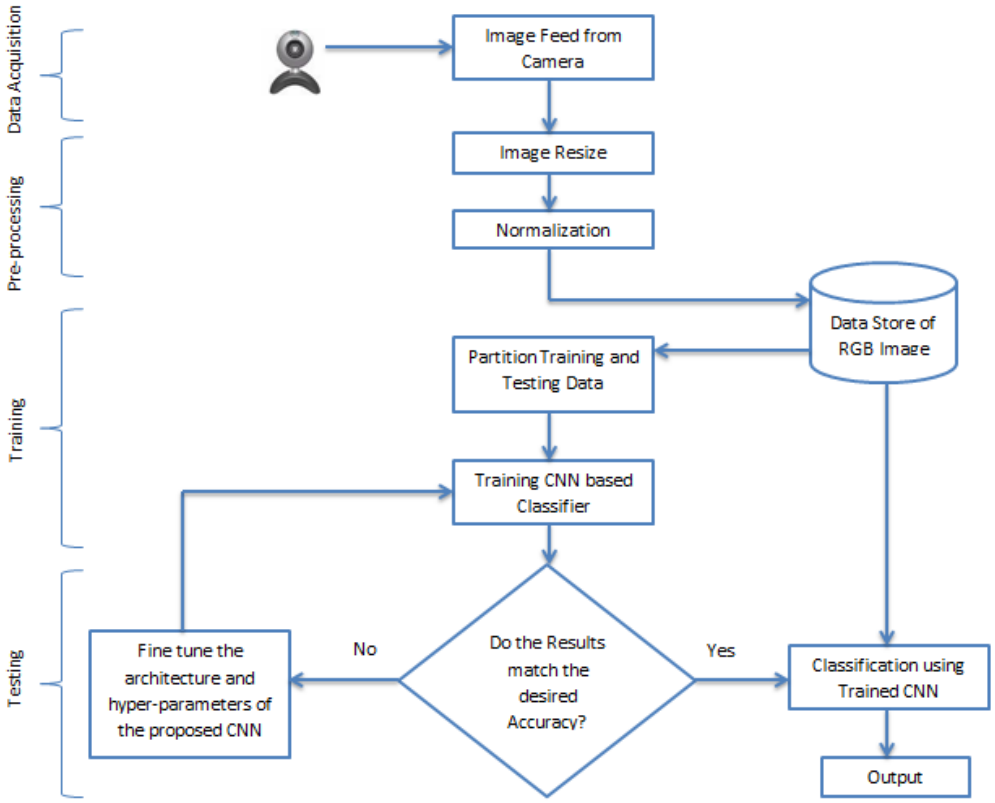


Figure 5.4: System Flow Chart

5.4 Training and Testing of CNN Architectures using Dataset of Static Signs

The dataset is broken down into three splits to perform training and testing of the developed model, as shown in Figure 5.5. The first set is the training set used to train the developed model and learn hidden patterns and features in the data. The second set is the validation set, which is part of the training set only and is used to validate the model's performance during training. The main logic behind the validation set is that it gives us the information that helps in tuning the model's hyperparameters, which further helps prevent the model's overfitting. The third set is the test set of the complete dataset used to test the developed model after training.

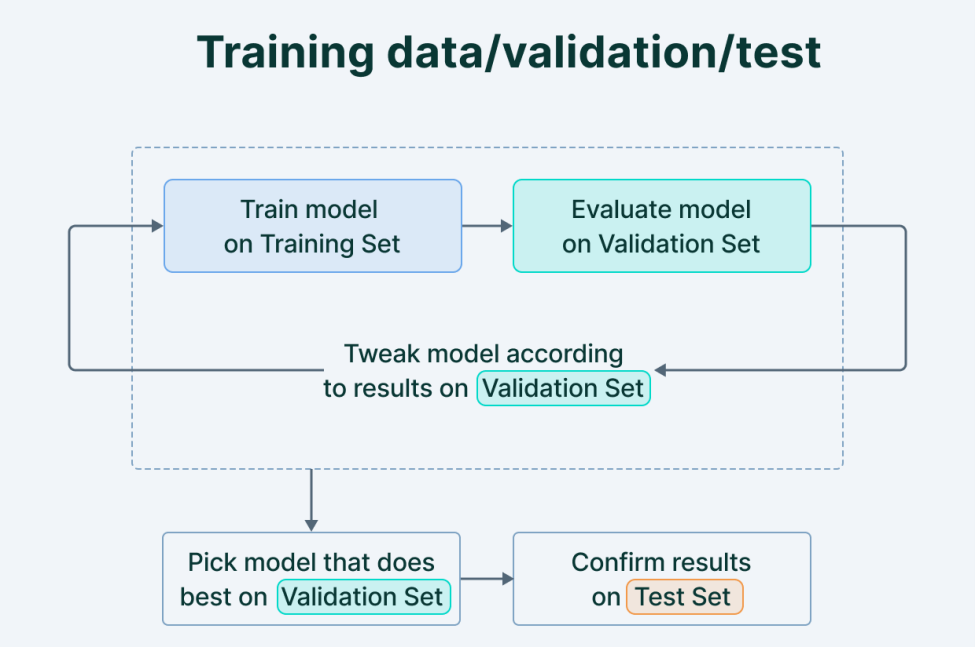


Figure 5.5: Training/ Validation/ Test Splits

5.4.1 Training and Testing using VGG for Static Signs

Visual Geometry Group (VGG) is a Convolutional Neural Network model introduced by Simonyan and A. Zisserman in 2014 [187]. It is a deep standardized network with multiple layers. The term "deep" refers to the number of layers with VGG16 and VGG 19 consisting of 16 and 19 layers, respectively. The main objective of the development of VGG is to enhance the model's performance by increasing the CNN model's depth. As training, the model using VGG16 and VGG19 was challenging, so the training started with the smaller versions of VGG with fewer layers. The VGG models employ the Adam optimization function with a learning rate of 0.001, a decay rate of 0.9, a batch size of 32, and 100 epochs. All the models are trained on Google Colaboratory with Nvidia K80 / T4 Graphical Processing Unit (GPU) and 12GB GPU Memory. VGG uses data augmentation and dropout layers to address overfitting during training. The classifier made for developing SLRS for static signs is trained on 80% of the dataset for 100 static signs.

5.4.1.1 Training using VGG16 Architecture for Static Signs

The input dimensions of the network are fixed to the image size (224 x 224) and have RGB (3 channels) images. The next step in this architecture is pre-processing, which

includes subtracting RGB values from each pixel of an image. After completion of pre-processing step, the images are transferred to convolutional layers with a small filter size of (3 x 3). After a few configurations, the filter size is reduced to (1 x 1), which defines the linear transformation of the input channels.

The convolution procedure is performed with a fixed stride of 1. Five max-pooling layers are used for spatial pooling, followed by several convolutional layers. The max-pooling operation is carried out on a pixel window of size (2 × 2) with a stride set to 2. Ultimately, completely interconnected layers with a constant structure are used. The first two layers each have 512 channels, the third layer does classification among 100 classes and has 100 channels, one for each category, and the last layer is the softmax layer. ReLu is the activation function that follows all of the hidden layers of the VGG network. VGG16 architecture is shown in Figure 5.6.

It has been noted that the total number of parameters in our gathered dataset is 18,962,852. This number represents the total size of the dataset, including all data points, features, and potentially non-trainable parameters. In a dataset, parameters could include various elements such as unique identifiers, labels, or other metadata associated with each data point. These parameters do not contribute to the trainable weights of the model but are part of the dataset's structure. Yet, the model uses just 18,961,828 trainable parameters to determine the appropriate weights to minimize the cost function of the model. These are the parameters that the model learns during the training process to minimize the cost function. Trainable parameters typically include weights and biases associated with the model's layers and neurons. These parameters are optimized through gradient descent or other optimization algorithms to make the model's predictions as accurate as possible.

The small difference in the numbers (18,962,852 - 18,961,828 = 1,024) likely accounts for the non-trainable parameters within the dataset, such as unique data point identifiers, labels, or other auxiliary information, which are not part of the model's optimization process but are essential for organizing and labeling the data.

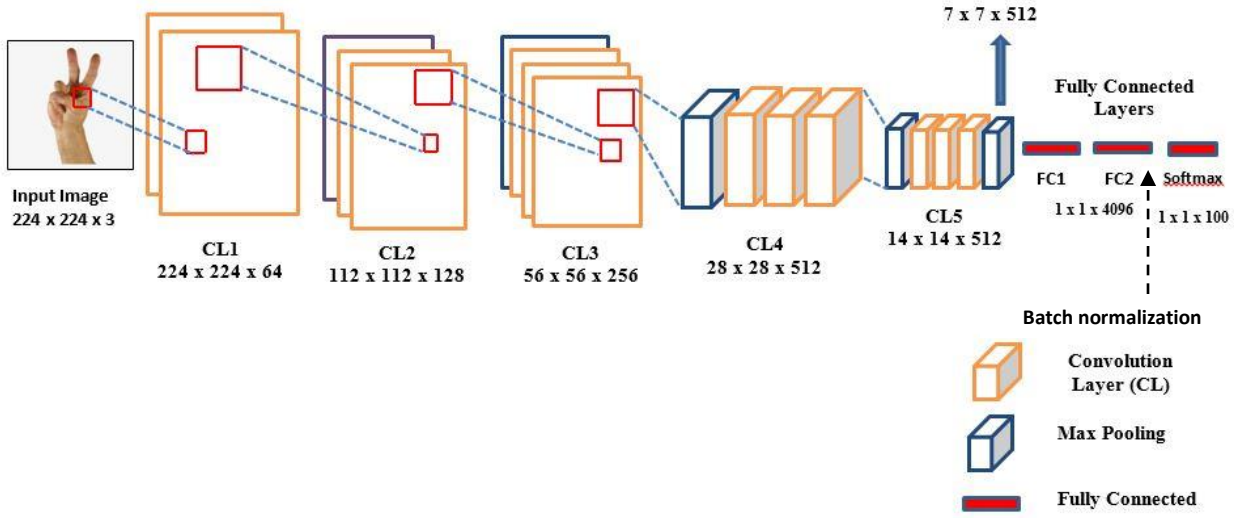


Figure 5.6: VGG 16 Architecture

5.4.1.2 Training using VGG19 Architecture for Static Signs

Another variant of the VGG model is VGG19; this model is similar to VGG16, just the difference lies in the number of layers only. VGG19 model consists of 19 layers in total. These 19 layers of VGG19 include 16 convolutional layers, three fully connected layers, and five max-pooling layers, as shown in Figure 5.7. The total number of parameters in our gathered dataset is 24,272,548; however, only 24,271,524 parameters are employed by the model to determine the best weights to minimize the cost function of the model.

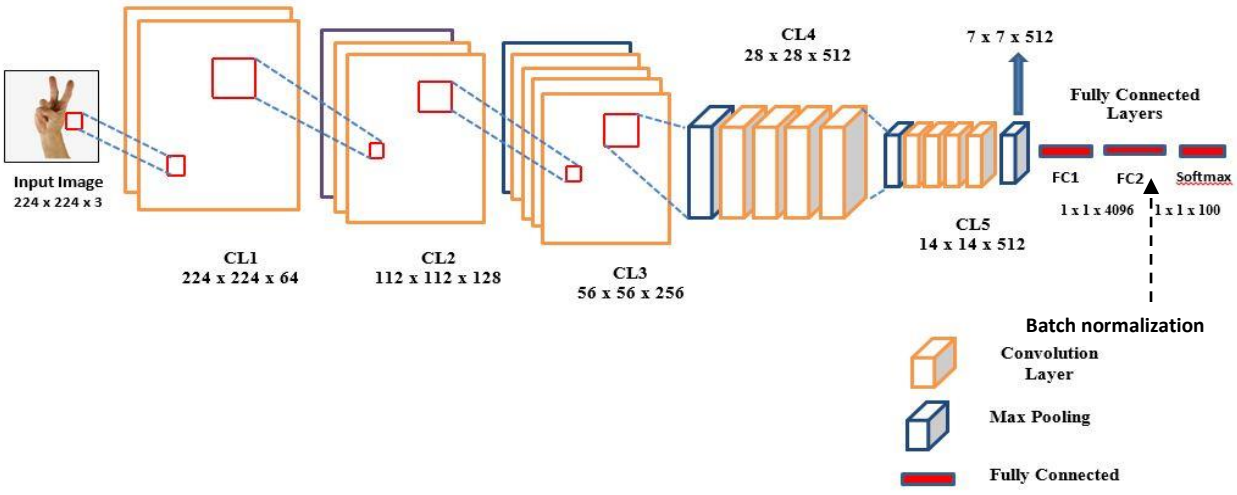


Figure 5.7: VGG19 Architecture

5.4.1.3 Testing of VGG Architectures for Static Signs

As the model is trained using VGG16 and Adam as an optimizer, the training and test accuracy of 97.76% and 96.72%, respectively. The training and test accuracy curve is shown in Figure 5.8.

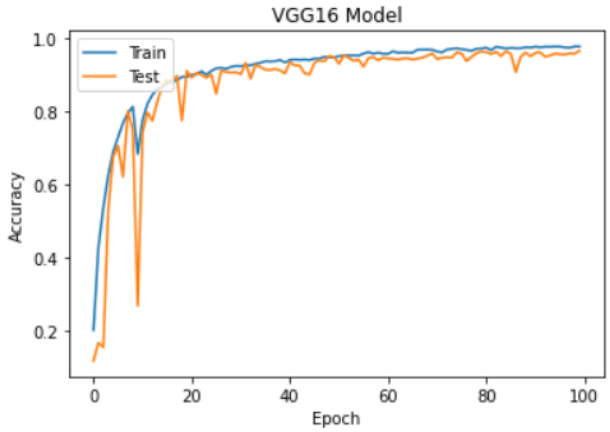


Figure 5.8: Accuracy for training and test datasets using VGG16

The classification performance of VGG16 using static sign dataset and MediaPipe showing precision, recall and F1 score is shown in Table 5.1. The evaluation parameters are taken as precision, recall and F1-score in the range of 0-1. It has been observed from the Table 5.1 that 76% of the tested signs received a flawless precision score of 1.00, and another 6% signs displayed strong precision with a score between 0.90 and 0.99. 12% of signs, on the other hand, attained precision scores between 0.80-0.89, 3%, 2% and 1% of signs are having precision score between 0.70-0.79, 0.60-0.69 and less than 0.60 respectively.

Similarly, in terms of recall scores, 79% signs attained a perfect score of 1.00, with an additional 6% signs falling within the range of 0.90 to 0.99 and another 6% in range of 0.80-0.89, while 4%, 2% and 3% signs displayed recall scores between 0.70-0.79, 0.60-0.69 and less than 0.60 respectively.

As for the F1-score, 56% signs secured a flawless F1-score of 1.00, 28% signs achieved F1-scores within the range of 0.90 to 0.99, 10% signs registered F1-scores between 0.80-

0.89, while 4% and 2% of signs attained F1 score between 0.70-0.79 and 0.60-0.69 respectively.

Table 5.1: Precision, Recall, andF1-Score for VGG16 on Static Sign Dataset using MediaPipe

S No.	Sign	Precision	Recall	F1-Score		S No.	Sign	Precision	Recall	F1-Score
1	A	1.00	0.80	0.89		**51**	M	1.00	0.83	0.91
2	Afraid	1.00	0.91	0.95		**52**	Me	1.00	0.62	0.77
3	Add	0.86	1.00	0.92		**53**	N	1.00	1.00	1.00
4	B	1.00	1.00	1.00		**54**	Nose	1.00	1.00	1.00
5	Bottle	1.00	1.00	1.00		**55**	Nine	1.00	1.00	1.00
6	Bud	1.00	1.00	1.00		**56**	Nurse	1.00	1.00	1.00
7	Bent	1.00	1.00	1.00		**57**	O	1.00	1.00	1.00
8	Between	1.00	1.00	1.00		**58**	Oath	1.00	1.00	1.00
9	Blind	1.00	1.00	1.00		**59**	One	0.90	1.00	0.95
10	Bowl	1.00	1.00	1.00		**60**	Open	1.00	1.00	1.00
11	Brain	0.90	1.00	0.95		**61**	Owl	1.00	1.00	1.00
12	C	0.91	1.00	0.95		**62**	P	1.00	1.00	1.00
13	Coolie	1.00	0.67	0.80		**63**	Policy	1.00	0.83	0.91
14	Cough	1.00	0.75	0.86		**64**	Pray	1.00	1.00	1.00
15	Cow	1.00	1.00	1.00		**65**	Promise	0.88	1.00	0.93
16	Chest	1.00	1.00	1.00		**66**	Q	0.67	1.00	0.80
17	Claw	1.00	0.88	0.93		**67**	R	1.00	1.00	1.00
18	D	1.00	1.00	1.00		**68**	S	1.00	1.00	1.00
19	Devil	0.86	1.00	0.92		**69**	Seven	1.00	0.77	0.87
20	Doctor	1.00	0.50	0.67		**70**	Soldier	0.75	1.00	0.86
21	E	1.00	0.57	0.73		**71**	Shirt	1.00	1.00	1.00
22	East	1.00	1.00	1.00		**72**	Six	1.00	0.90	0.95
23	Eight	1.00	0.90	0.95		**73**	Sick	1.00	1.00	1.00
24	Evening	1.00	1.00	1.00		**74**	Skin	1.00	1.00	1.00
25	Elbow	1.00	1.00	1.00		**75**	Shoulder	1.00	1.00	1.00
26	Eye	1.00	1.00	1.00		**76**	Stand	1.00	1.00	1.00
27	F	0.67	1.00	0.80		**77**	Strong	1.00	1.00	1.00
28	Fat	0.80	1.00	0.89		**78**	Sleep	1.00	1.00	1.00
29	Faith	0.85	1.00	0.92		**79**	Sunday	1.00	0.92	0.96
30	Fever	0.91	1.00	0.95		**80**	T	1.00	0.75	0.86
31	Feel	1.00	1.00	1.00		**81**	Ten	1.00	1.00	1.00
32	Few	1.00	0.82	0.90		**82**	Telephone	1.00	1.00	1.00

33	Food	0.82	1.00	0.90		**83**	Tongue	1.00	1.00	1.00
34	Four	1.00	1.00	1.00		**84**	Thorn	0.92	1.00	0.96
35	Fist	0.88	1.00	0.93		**85**	Three	0.57	1.00	0.73
36	Five	1.00	1.00	1.00		**86**	Trouble	1.00	1.00	1.00
37	G	1.00	0.86	0.92		**87**	Two	0.83	1.00	0.91
38	Gun	0.86	1.00	0.92		**88**	ThumbsUp	0.75	1.00	0.86
39	Good	1.00	1.00	1.00		**89**	U	1.00	0.92	0.96
40	Hair	1.00	1.00	1.00		**90**	V	1.00	0.91	0.95
41	Hand	1.00	1.00	1.00		**91**	W	1.00	1.00	1.00
42	Head	1.00	1.00	1.00		**92**	West	1.00	1.00	1.00
43	Hear	0.88	1.00	0.93		**93**	Wedding	1.00	1.00	1.00
44	I	1.00	1.00	1.00		**94**	Water	1.00	0.50	0.67
45	Jain	1.00	1.00	1.00		**95**	White	1.00	1.00	1.00
46	K	1.00	1.00	1.00		**96**	X	1.00	1.00	1.00
47	King	1.00	1.00	1.00		**97**	Y	0.94	1.00	0.97
48	L	1.00	1.00	1.00		**98**	You	0.88	1.00	0.93
49	Love	1.00	1.00	1.00		**99**	Z	1.00	1.00	1.00
50	Leprosy	0.78	0.78	0.78		**100**	Zero	0.86	1.00	0.92

As the model is trained using VGG19 and Adam as the optimizer, the training and test accuracies are 99.64% and 96%, respectively, as shown in Figure 5.9.

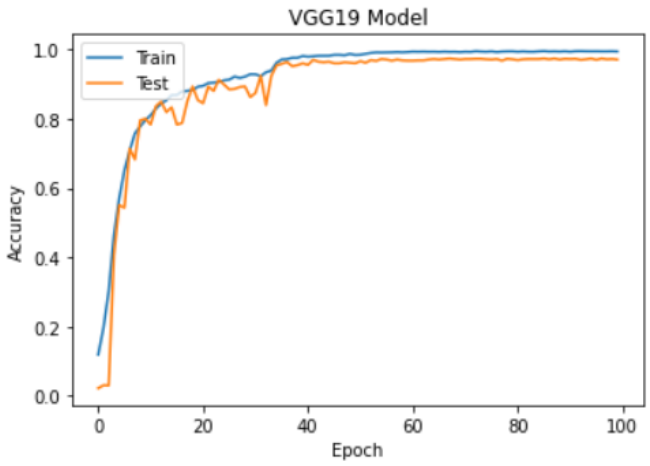


Figure 5.9: Accuracy curve for training and test datasets using VGG19

The classification performance of VGG19 using static sign dataset and MediaPipe showing precision, recall, F1 score and prediction rate is shown in Table 5.2. It has been observed from Table 5.2 that out of the signs evaluated, 83% signs achieved a perfect

precision score of 1.00, while 10% signs demonstrated strong precision scores ranging from 0.90 to 0.99. In contrast, 2%, 3% and 2% of signs exhibited precision scores between 0.80-0.89, 0.70-0.79 and 0.60-0.69 respectively.

Similarly, regarding recall scores, 84% signs achieved a perfect score of 1.00, 8% signs fell within the range of 0.90 to 0.99, 5%, 2% and 1% of signs exhibited recall scores between 0.80-0.89, 0.70-0.79 and 0.60-0.69 respectively.

For the F1-score, 71% of signs obtained a flawless score of 1.00, 24% signs achieved scores ranging from 0.90 to 0.99, and rest 2%, 1% and 2% of signs achieved F1-scores between 0.80-0.89, 0.70-0.79 and 0.60-0.69 respectively.

Table 5.2: Precision, Recall, F1-Score and Prediction rate for VGG19 on Static Sign Dataset using MediaPipe

S No.	Sign	Precision	Recall	F1-Score		S No.	Sign	Precision	Recall	F1-Score
1	A	1.00	1.00	1.00		**51**	M	1.00	1.00	1.00
2	Afraid	1.00	0.91	0.95		**52**	Me	0.71	0.62	0.66
3	Add	0.86	1.00	0.92		**53**	N	1.00	1.00	1.00
4	B	1.00	1.00	1.00		**54**	Nose	0.67	1.00	0.80
5	Bottle	1.00	1.00	1.00		**55**	Nine	1.00	1.00	1.00
6	Bud	1.00	1.00	1.00		**56**	Nurse	1.00	1.00	1.00
7	Bent	1.00	0.91	0.95		**57**	O	1.00	1.00	1.00
8	Between	1.00	1.00	1.00		**58**	Oath	1.00	1.00	1.00
9	Blind	1.00	1.00	1.00		**59**	One	1.00	1.00	1.00
10	Bowl	1.00	0.94	0.97		**60**	Open	1.00	1.00	1.00
11	Brain	1.00	0.96	0.98		**61**	Owl	1.00	1.00	1.00
12	C	1.00	1.00	1.00		**62**	P	1.00	1.00	1.00
13	Coolie	0.96	1.00	0.98		**63**	Policy	0.96	1.00	0.98
14	Cough	1.00	1.00	1.00		**64**	Pray	1.00	1.00	1.00
15	Cow	1.00	1.00	1.00		**65**	Promise	1.00	0.86	0.92
16	Chest	1.00	1.00	1.00		**66**	Q	0.94	1.00	0.97
17	Claw	1.00	1.00	1.00		**67**	R	0.91	1.00	0.95
18	D	1.00	1.00	1.00		**68**	S	1.00	0.89	0.94
19	Devil	1.00	1.00	1.00		**69**	Seven	1.00	0.85	0.92
20	Doctor	1.00	1.00	1.00		**70**	Soldier	0.75	1.00	0.86
21	E	0.86	0.95	0.90		**71**	Shirt	1.00	1.00	1.00
22	East	1.00	1.00	1.00		**72**	Six	0.91	1.00	0.95
23	Eight	1.00	1.00	1.00		**73**	Sick	1.00	1.00	1.00

24	Evening	1.00	1.00	1.00		**74**	Skin	1.00	1.00	1.00
25	Elbow	1.00	1.00	1.00		**75**	Shoulder	1.00	1.00	1.00
26	Eye	1.00	1.00	1.00		**76**	Stand	1.00	1.00	1.00
27	F	0.60	0.75	0.67		**77**	Strong	1.00	1.00	1.00
28	Fat	1.00	0.96	0.98		**78**	Sleep	1.00	1.00	1.00
29	Faith	1.00	1.00	1.00		**79**	Sunday	1.00	0.92	0.96
30	Fever	0.95	1.00	0.97		**80**	T	1.00	1.00	1.00
31	Feel	1.00	1.00	1.00		**81**	Ten	1.00	1.00	1.00
32	Few	1.00	1.00	1.00		**82**	Telephone	1.00	1.00	1.00
33	Food	1.00	1.00	1.00		**83**	Tongue	1.00	1.00	1.00
34	Four	1.00	1.00	1.00		**84**	Thorn	1.00	0.83	0.91
35	Fist	1.00	1.00	1.00		**85**	Three	1.00	1.00	1.00
36	Five	1.00	1.00	1.00		**86**	Trouble	1.00	1.00	1.00
37	G	1.00	0.86	0.92		**87**	Two	1.00	1.00	1.00
38	Gun	1.00	1.00	1.00		**88**	ThumbsUp	1.00	1.00	1.00
39	Good	0.92	1.00	0.96		**89**	U	0.92	1.00	0.96
40	Hair	1.00	0.94	0.97		**90**	V	1.00	1.00	1.00
41	Hand	1.00	1.00	1.00		**91**	W	1.00	1.00	1.00
42	Head	1.00	1.00	1.00		**92**	West	1.00	1.00	1.00
43	Hear	1.00	1.00	1.00		**93**	Wedding	1.00	1.00	1.00
44	I	1.00	1.00	1.00		**94**	Water	1.00	1.00	1.00
45	Jain	1.00	1.00	1.00		**95**	White	1.00	1.00	1.00
46	K	1.00	1.00	1.00		**96**	X	1.00	1.00	1.00
47	King	0.90	1.00	0.95		**97**	Y	1.00	1.00	1.00
48	L	1.00	1.00	1.00		**98**	You	1.00	1.00	1.00
49	Love	1.00	1.00	1.00		**99**	Z	1.00	1.00	1.00
50	Leprosy	0.78	0.78	0.78		**100**	Zero	0.96	1.00	0.98

5.4.2 Training and Testing using GoogleNet for Static Signs

GoogleNet architecture was proposed at google in 2014. This architecture has provided a significant reduction in error rate as compared to VGG architectures [108]. The complete GoogleNet architecture consists of 22 layers in total. In GoogleNet architecture, three input filter sizes (1x1), (3x3), and (5x5) are used to perform convolution operations. A max-pooling operation is also performed along with the convolution, which is further sent to the next inception module, as shown in Figure 5.10.

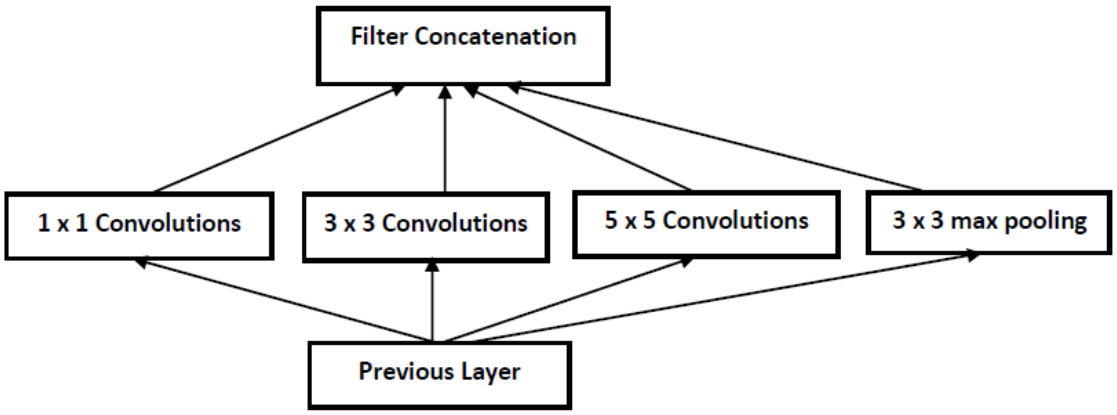


Figure 5.10: Inception Module [108]

As neural networks are expensive and time-consuming to train, each (3x3) and (5x5) convolution is added with one extra (1x1) convolution, as shown in Figure 5.11. Adding (1x1) convolution helps dimensionality reduction and faster computations.

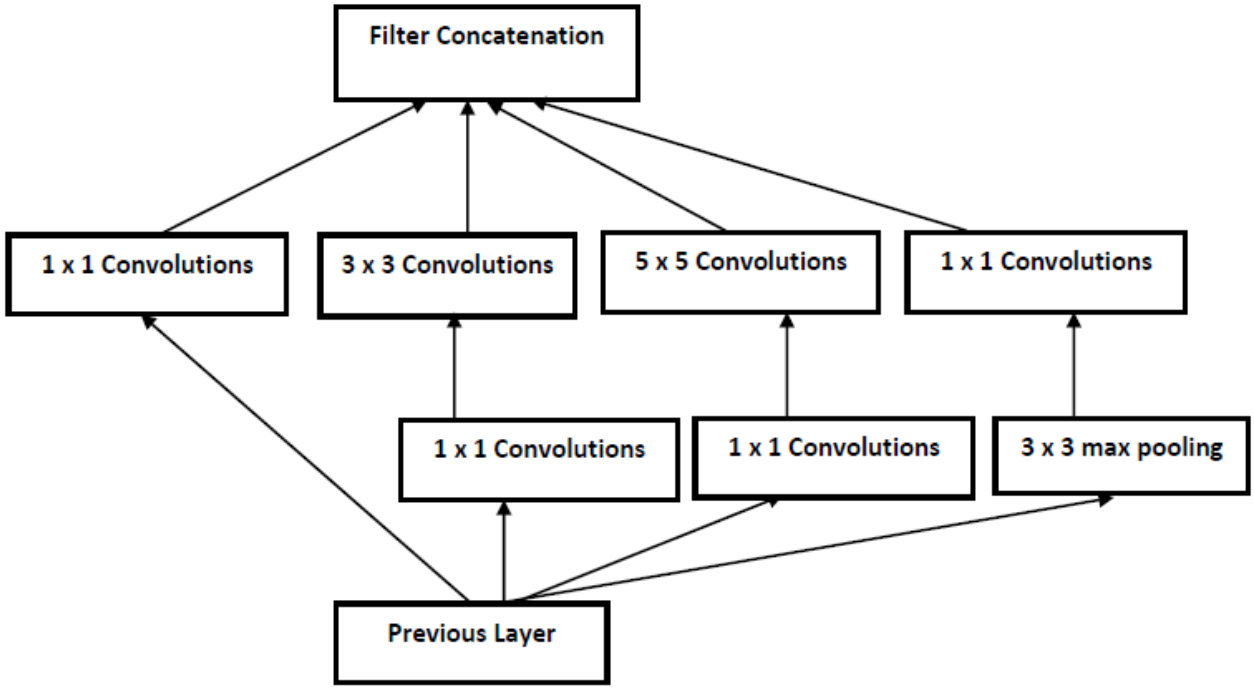


Figure 5.11: Inception Module with Dimension Reduction [108]

The classifier made for developing SLRS for static signs is also trained by using the GoogleNet model on the dataset for static signs. It has been observed that the model has been trained by using 7,465,052 parameters in total.

Testing of GoogleNet Architecture for Static Signs

As the model is trained using GoogleNet and Adam as an optimizer with a learning rate=0.001, decay rate=0.99, batch size 32 and 100 epochs, the training and test accuracy of 97.75% and 96.54%, respectively. The accuracy curve using GoogleNet architecture is shown in Figure 5.12. The classification performance of GoogleNet architecture using the dataset for static signs and MediaPipe showing precision, recall, and F1 score is shown in Table 5.3.

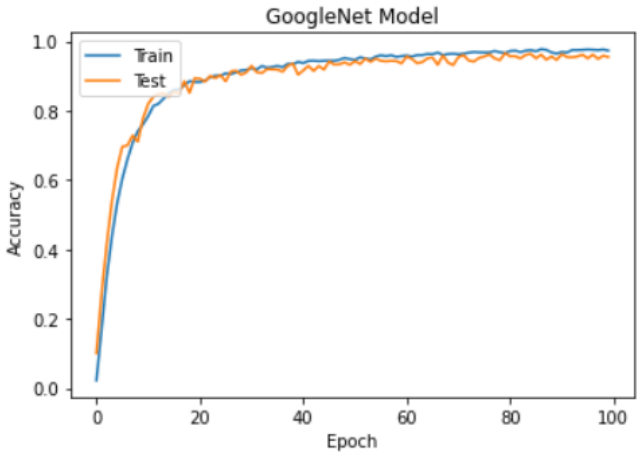


Figure 5.12: Accuracy curve using GoogleNet Architecture

Precision, recall, F1 score and prediction rate w.r.t GoogleNet architecture are shown in Table 5.3. It has been observed from Table 5.3 that out of the signs analyzed, 72% signs achieved a perfect precision score of 1.00, while 7% signs displayed precision scores ranging from 0.90 to 0.99. Conversely, 15% signs exhibited precision scores between 0.80 - 0.89, 3%, 2% and 1% achieved precision in range 0.70-0.79, 0.60-0.69 and below 0.60 respectively.

Similarly, in terms of recall scores, 70% signs obtained a perfect score of 1.00, 15% signs fell within the range of 0.90 to 0.99, and 6% signs recorded recall scores between 0.80-0.89. Conversely, 4%, 2% and 3% exhibited recall score between 0.70-0.79, 0.60-0.69 and below 0.60 respectively.

Regarding the F1-score, 54% of signs received a perfect F1-score of 1.00, 27% signs attained scores within the range 0.90 to 0.99, and 13% of signs achieved F1-scores between 0.80-0.89, and rest 4% and 2% of signs achieved F1-score between 0.70-0.79 and 0.60-0.69 respectively.

Table 5.3: Precision, Recall, F1-Score and Prediction Rate for GoogleNet on Static Sign Dataset using MediaPipe

S No.	Sign	Precision	Recall	F1-Score		S No.	Sign	Precision	Recall	F1-Score
1	A	1.00	0.80	0.89		**51**	M	1.00	0.83	0.91
2	Afraid	1.00	0.91	0.95		**52**	Me	1.00	0.62	0.77
3	Add	0.86	0.96	0.91		**53**	N	1.00	1.00	1.00
4	B	0.97	0.95	0.96		**54**	Nose	1.00	1.00	1.00
5	Bottle	1.00	1.00	1.00		**55**	Nine	1.00	1.00	1.00
6	Bud	1.00	1.00	1.00		**56**	Nurse	1.00	1.00	1.00
7	Bent	1.00	1.00	1.00		**57**	O	1.00	1.00	1.00
8	Between	1.00	1.00	1.00		**58**	Oath	1.00	1.00	1.00
9	Blind	1.00	1.00	1.00		**59**	One	0.90	1.00	0.95
10	Bowl	1.00	1.00	1.00		**60**	Open	1.00	1.00	1.00
11	Brain	0.90	0.95	0.92		**61**	Owl	1.00	1.00	1.00
12	C	0.91	0.95	0.93		**62**	P	1.00	1.00	1.00
13	Coolie	1.00	0.67	0.80		**63**	Policy	1.00	0.83	0.91
14	Cough	1.00	0.75	0.86		**64**	Pray	0.88	1.00	0.94
15	Cow	1.00	1.00	1.00		**65**	Promise	0.88	0.96	0.92
16	Chest	1.00	1.00	1.00		**66**	Q	0.67	1.00	0.80
17	Claw	0.89	0.88	0.88		**67**	R	1.00	1.00	1.00
18	D	1.00	1.00	1.00		**68**	S	1.00	1.00	1.00
19	Devil	0.86	0.94	0.90		**69**	Seven	1.00	0.77	0.87
20	Doctor	1.00	0.50	0.67		**70**	Soldier	0.75	1.00	0.86
21	E	1.00	0.57	0.73		**71**	Shirt	1.00	1.00	1.00
22	East	1.00	1.00	1.00		**72**	Six	1.00	0.90	0.95
23	Eight	1.00	0.90	0.95		**73**	Sick	1.00	1.00	1.00
24	Evening	1.00	1.00	1.00		**74**	Skin	1.00	1.00	1.00
25	Elbow	1.00	1.00	1.00		**75**	Shoulder	1.00	1.00	1.00
26	Eye	1.00	1.00	1.00		**76**	Stand	1.00	1.00	1.00
27	F	0.67	1.00	0.80		**77**	Strong	1.00	1.00	1.00
28	Fat	0.80	1.00	0.89		**78**	Sleep	1.00	1.00	1.00
29	Faith	0.85	1.00	0.92		**79**	Sunday	1.00	0.92	0.96
30	Fever	0.91	1.00	0.95		**80**	T	1.00	0.75	0.86

31	Feel	1.00	1.00	1.00		**81**	Ten	1.00	1.00	1.00
32	Few	1.00	0.82	0.90		**82**	Telephone	1.00	1.00	1.00
33	Food	0.82	1.00	0.90		**83**	Tongue	1.00	1.00	1.00
34	Four	1.00	1.00	1.00		**84**	Thorn	0.92	1.00	0.96
35	Fist	0.88	0.96	0.92		**85**	Three	0.57	1.00	0.73
36	Five	1.00	1.00	1.00		**86**	Trouble	1.00	1.00	1.00
37	G	0.88	0.86	0.87		**87**	Two	0.83	1.00	0.91
38	Gun	0.86	0.93	0.89		**88**	ThumbsUp	0.75	1.00	0.86
39	Good	1.00	1.00	1.00		**89**	U	1.00	0.92	0.96
40	Hair	1.00	1.00	1.00		**90**	V	1.00	0.91	0.95
41	Hand	1.00	1.00	1.00		**91**	W	1.00	1.00	1.00
42	Head	1.00	1.00	1.00		**92**	West	1.00	1.00	1.00
43	Hear	0.88	0.94	0.91		**93**	Wedding	1.00	1.00	1.00
44	I	1.00	1.00	1.00		**94**	Water	1.00	0.50	0.67
45	Jain	1.00	1.00	1.00		**95**	White	1.00	1.00	1.00
46	K	1.00	1.00	1.00		**96**	X	1.00	1.00	1.00
47	King	1.00	1.00	1.00		**97**	Y	0.94	1.00	0.97
48	L	1.00	1.00	1.00		**98**	You	0.88	1.00	0.94
49	Love	1.00	1.00	1.00		**99**	Z	1.00	1.00	1.00
50	Leprosy	0.78	0.78	0.78		**100**	Zero	0.86	1.00	0.92

5.4.3 CNN Architecture without using MediaPipe for Static Signs

After implementing pre-trained models of CNN, a new model based on CNN has been implemented. This model's training is based on convolutional neural networks, and no MediaPipe has been used for data pre-processing.

Training

This model is trained using Google Colaboratory, in which the classifier takes sign images and classifies them into the corresponding category. The classifier is trained on the dataset of 100 static signs. The dataset is shuffled and divided into training and test sets, with the size of the training set being 80% of the whole dataset. Shuffling the dataset significantly adds randomness to the neural network training process, which prevents the network from being biased towards specific parameters. The configuration of the CNN architecture used in the proposed system is described in Table 5.4.

Table 5.4: Parameter Configuration of Proposed CNN Architecture

Layer Type	Output Size	Parameters
Input	128x128x3	--
Conv2d_1	(128,128,16)	448
Conv2d_2	(126,126,16)	2320
Maxpooling2d_1	(63,63,16)	0
Dropout	(63,63,16)	0
Flatten	63504	0
Dense_1(FC1)	64	4064272
Dense_2(FC2)	100	6500
Total Parameters:	4073540	
Trainable Parameters:	4073540	
Non-trainable Parameters:	0	

Testing

The Indian SLRS is evaluated based on two different experiments. In the first experiment, the parameters used in training the model are fine-tuned in which the number of layers, filters, and optimizers has been changed. In the second experiment, the performance of the trained model is evaluated on color and the greyscale image dataset.

The developed SLRS has been tested on approximately 50 convolutional neural network models. The algorithms with different optimizers are used to train the network for 100 epochs with the loss function as categorical cross-entropy. Some of the other parameters used to fine-tune the network architecture are described in Table 5.5. To select an optimized CNN configuration with four layers having two convolutional layers and two fully connected layers, the different number of filters has been examined to achieve the best accuracy. It has been observed from the experimental result that the highest training and testing accuracy of 99.17% and 98.80%, respectively, have been achieved with 16 filters.

Table 5.5: Experimental Results for parameters

Number of Layers	Number of Filters	Training Accuracy	Testing Accuracy
4 (2 CL, 2FC)	16	99.17%	98.80%
4 (2 CL, 2FC)	32	98.82%	98.53%
4 (2 CL, 2FC)	64	99.05%	98.76%

The optimizers are used to tweak the model's parameters or weights, which helps minimize the loss function and predict results as accurately as possible. This tests the proposed model on optimizers like Adaptive Moment Estimation (Adam), RMSprop, Stochastic Gradient Descent (SGD), Adadelta and Adagrad. Experimental results for optimizers and colored image datasets are represented in Table 5.6. It has been observed that the SGD outperformed all other optimizers with 16 filters and four layers. The proposed model obtained the training and testing accuracy of 99.72% and 98.56%, respectively, using an SGD optimizer.

Table 5.6: Experimental results for the optimizer and colored images

Optimizer	**Training Accuracy**	**Training Loss**	**Testing Accuracy**	**Test Loss**
Adagrad	82%	2.934	79.74%	3.054
Adadelta	95.01%	1.036	93.54%	1.670
Adam	99.17%	0.0280	98.80%	0.0684
RMSProp	99.59%	0.0378	98.27%	0.1940
SGD	99.72%	0.0126	98.56%	0.0759

The proposed model is also tested on greyscale data. The results of CNN architecture without using MediaPipe are shown in Table 5.7. It has been observed that the model achieved the highest training and testing accuracy of 99.90% and 98.70%, respectively, with the SGD optimizer.

Table 5.7: Experimental results for the optimizer and greyscale images

Optimizer	**Training Accuracy**	**Training Loss**	**Testing Accuracy**	**Test Loss**
Adagrad	82.7%	2.064	80.2%	2.994
Adadelta	95.4%	0.932	93.9%	1.197
Adam	99.24%	0.0280	98.85%	0.0684
RMSProp	99.76%	0.0378	98.35%	0.1940
SGD	99.90%	0.0126	98.70%	0.0759

It has been observed from Figure 5.13 that the training and testing accuracy without using MediaPipe increases up to 15 epochs, as well as training and testing loss is also decreasing. After epoch 15, the model gets trained to the highest accuracy of 99.9%, and there is no change in accuracy and loss function. Hence, training is performed for 20 epochs to avoid overfitting.

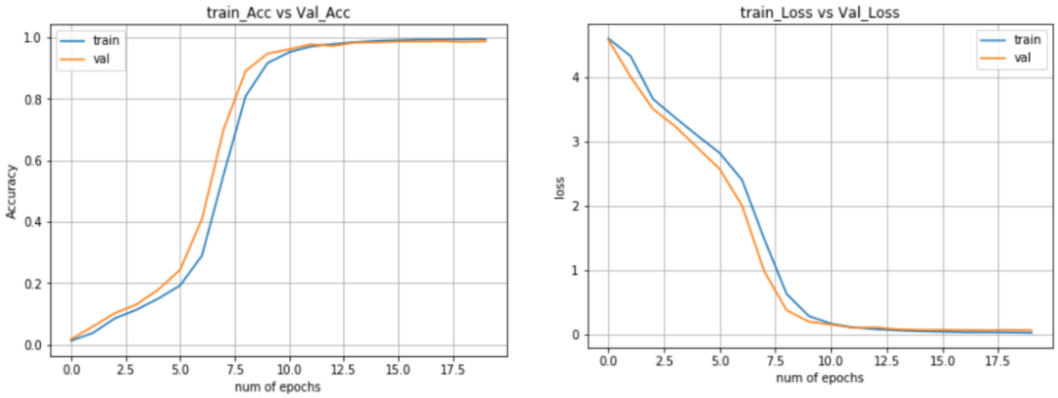


Figure 5.13: Accuracy and loss curves for training and testing datasets without using MediaPipe

This model is highly overfitted as it performs well on training and validation datasets, but its performance degrades when we have tested it on unseen data. So a new technique, MediaPipe, has been embedded in this CNN architecture to improve the results further. The classification performance of CNN architecture without using MediaPipe for static signs dataset showing precision, recall, F1 score and prediction rate is shown in Table 5.8. It has been observed from Table 5.8 that among the signs examined, 58% signs demonstrated a precision score of 1.00, indicating perfect precision, while 41% signs exhibited precision scores ranging from 0.90 to 0.99. Conversely, 1% of signs displayed precision scores between 0.70-0.79.

Similarly, in the context of recall scores, 63% signs achieved a flawless score of 1.00, while 35% signs fell within the range of 0.90 to 0.99, and rest 1% of signs recorded recall scores between 0.80-0.89 and 0.70-0.79.

As for the F1-score, 39% of signs obtained a perfect score of 1.00, 59% signs reached scores within the range 0.90 to 0.99, and 2% signs achieved F1-scores between 0.80-0.89.

Table 5.8 Precision, Recall, F1-Score and Prediction Rate for CNN Architecture without using MediaPipe

S No.	Sign	Precision	Recall	F1-Score		S No.	Sign	Precision	Recall	F1-Score
1	A	1.00	0.96	0.98		**51**	M	0.97	0.79	0.87
2	Afraid	0.97	0.97	0.97		**52**	Me	1.00	1.00	1.00
3	Add	1.00	1.00	1.00		**53**	N	1.00	1.00	1.00
4	B	1.00	1.00	1.00		**54**	Nose	0.98	1.00	0.99
5	Bottle	1.00	0.97	0.99		**55**	Nine	1.00	1.00	1.00
6	Bud	1.00	1.00	1.00		**56**	Nurse	0.99	0.98	0.98
7	Bent	0.89	1.00	0.99		**57**	O	0.96	0.97	0.96
8	Between	1.00	0.99	0.99		**58**	Oath	1.00	1.00	1.00
9	Blind	1.00	1.00	1.00		**59**	One	1.00	0.99	0.99
10	Bowl	0.97	1.00	0.98		**60**	Open	1.00	0.97	0.98
11	Brain	0.99	0.99	0.99		**61**	Owl	1.00	1.00	1.00
12	C	1.00	1.00	1.00		**62**	P	1.00	0.97	0.98
13	Coolie	0.97	0.94	0.96		**63**	Policy	0.98	1.00	0.99
14	Cough	1.00	1.00	1.00		**64**	Pray	1.00	1.00	1.00
15	Cow	0.99	0.97	0.98		**65**	Promise	1.00	1.00	1.00
16	Chest	1.00	1.00	1.00		**66**	Q	0.97	1.00	0.99
17	Claw	1.00	1.00	1.00		**67**	R	0.98	1.00	0.99
18	D	0.79	0.97	0.87		**68**	S	0.95	1.00	0.97
19	Devil	0.88	0.94	0.96		**69**	Seven	0.96	1.00	0.98
20	Doctor	0.98	1.00	0.99		**70**	Soldier	1.00	1.00	1.00
21	E	1.00	1.00	1.00		**71**	Shirt	1.00	1.00	1.00
22	East	1.00	1.00	1.00		**72**	Six	1.00	0.97	0.99
23	Eight	0.96	0.90	0.93		**73**	Sick	1.00	1.00	1.00
24	Evening	1.00	0.96	0.98		**74**	Skin	1.00	1.00	1.00
25	Elbow	0.95	1.00	0.98		**75**	Shoulder	1.00	0.98	0.99
26	Eye	1.00	1.00	1.00		**76**	Stand	1.00	1.00	1.00
27	F	0.97	0.97	0.97		**77**	Strong	0.97	1.00	0.98
28	Fat	1.00	1.00	1.00		**78**	Sleep	1.00	0.95	0.98
29	Faith	0.98	1.00	0.99		**79**	Sunday	1.00	1.00	1.00
30	Fever	0.95	1.00	0.97		**80**	T	0.99	1.00	0.99
31	Feel	0.97	1.00	0.98		**81**	Ten	1.00	0.98	0.99
32	Few	1.00	1.00	1.00		**82**	Telephone	1.00	1.00	1.00
33	Food	0.98	0.86	0.92		**83**	Tongue	0.99	1.00	0.99
34	Four	1.00	0.96	0.98		**84**	Thorn	0.96	1.00	0.98
35	Fist	0.97	0.98	0.97		**85**	Three	0.97	1.00	0.99

36	Five	1.00	1.00	1.00		**86**	Trouble	1.00	0.95	0.97
37	G	0.97	0.97	0.97		**87**	Two	1.00	1.00	1.00
38	Gun	0.97	1.00	0.99		**88**	ThumbsUp	1.00	0.95	0.97
39	Good	1.00	1.00	1.00		**89**	U	1.00	0.99	0.99
40	Hair	1.00	1.00	1.00		**90**	V	1.00	1.00	1.00
41	Hand	0.97	1.00	0.98		**91**	W	1.00	1.00	1.00
42	Head	0.90	0.99	0.94		**92**	West	1.00	0.93	0.96
43	Hear	0.97	1.00	0.98		**93**	Wedding	0.99	0.99	0.99
44	I	1.00	1.00	1.00		**94**	Water	0.87	0.98	0.95
45	Jain	0.99	1.00	0.99		**95**	White	1.00	0.98	0.99
46	K	0.98	0.98	0.98		**96**	Word	1.00	1.00	1.00
47	King	1.00	0.95	0.97		**97**	X	1.00	1.00	1.00
48	L	1.00	1.00	1.00		**98**	You	0.97	1.00	0.99
49	Love	1.00	0.98	0.99		**99**	Z	0.99	1.00	0.99
50	Leprosy	1.00	1.00	1.00		**100**	Zero	1.00	1.00	1.00

5.4.4 CNN Architecture using MediaPipe for Static Signs

In this architecture, MediaPipe is used along with CNN architecture. The collected dataset for static signs is pre-processed using the MediaPipe technique, as discussed in chapter 4.

Training

This CNN model consists of ten convolutional layers, four max-pooling layers, and two fully connected layers. The total number of parameters in our gathered dataset is 6,234,084. However, only 6,231,204 parameters are utilized by the model to determine the best weights to minimize the cost function of the model.

Testing

As the CNN architecture without MediaPipe gives a low performance, a MediaPipe technique has been introduced. The CNN architecture using MediaPipe has been tested on a static sign dataset with approximately 20 CNN models. In this, various experiments were performed by changing the number of layers and optimizers. It has been observed from Table 5.9 that the accuracy of the proposed model increases as we go on increasing the number of layers. The highest training and testing accuracy achieved is 95% and 93.56%, respectively.

Table 5.9: Experimental Results for different layers

Number of Layers	Training Accuracy	Training Loss	Testing Accuracy	Test Loss
12 (8CL,2MP,2FC)	94.56%	0.3123	93.15%	0.3620
15 (10CL,3MP,2FC)	94.89%	0.2841	93.42%	0.3475
16 (10CL,4MP,2FC)	95%	0.2640	93.56%	0.2980

CL: Convolution Layer, MP: MaxPooling, FC: Fully Connected

Table 5.10: Experimental Results for different optimizers using MediaPipe

Training Accuracy	Training Loss	Test Accuracy	Test Loss	Optimizer
80.04%	3.010	75%	4.021	Adagrad
91.2%	1.965	89.6%	1.824	Adadelta
94.10%	0.3265	93.22%	0.3712	RMSProp
94.26%	0.2999	93.80%	0.2792	SGD
95%	0.2640	93.56%	0.2980	Adam

The CNN architecture using MediaPipe with ten convolution layers, four max-pooling layers, and two fully connected layers is also tested using different optimizers. The experimental results for optimizers and colored images from the dataset of static signs are represented in Table 5.10. It has been observed that the Adam optimizer outperformed all other optimizers and obtained the highest training and testing accuracy of 95% and 93.56%, respectively.

The classification performance of CNN architecture using MediaPipe for static signs dataset showing precision, recall, and F1 score is shown in Table 5.11. It has been observed from Table 5.11 that among the signs under examination, 71% of signs achieved an impeccable precision score of 1.00, with 4% signs showcasing precision scores spanning from 0.90 to 0.99. Conversely, 19% signs exhibited precision scores between 0.80-0.89, while 3%, 1% and 2% achieved score between 0.70-0.79, 0.60-0.69 and below 0.60 respectively.

Similarly, concerning recall scores, 68% signs secured a flawless score of 1.00, while 8% signs fell within the range of 0.90 to 0.99, and an additional 19% signs achieved recall scores between 0.80-0.89. Also, 4% and 1% of signs attained recall score between 0.70-0.79 and below 0.60 respectively.

In terms of the F1-score, 52% signs attained a perfect score of 1.00, 27% signs attained scores within the 0.90 to 0.99 range, and 16% signs achieved F1-scores between 0.80-0.89. While 3% and 2% signs achieved F1-score between 0.70-0.79 and below 0.60 respectively.

Table 5.11 Precision, Recall and F1-Score for CNN Architecture using MediaPipe

S No.	Sign	Precision	Recall	F1-Score	S No.	Sign	Precision	Recall	F1-Score
1	A	1.00	1.00	1.00	**51**	M	0.92	0.92	0.92
2	Afraid	1.00	1.00	1.00	**52**	Me	1.00	1.00	1.00
3	Add	1.00	0.89	0.94	**53**	N	0.71	0.83	0.77
4	B	1.00	1.00	1.00	**54**	Nose	1.00	0.82	0.90
5	Bottle	1.00	0.80	0.89	**55**	Nine	1.00	1.00	1.00
6	Bud	0.82	1.00	0.90	**56**	Nurse	1.00	1.00	1.00
7	Bent	0.89	1.00	0.94	**57**	O	1.00	1.00	1.00
8	Between	1.00	0.89	0.94	**58**	Oath	1.00	1.00	1.00
9	Blind	0.86	0.75	0.80	**59**	One	1.00	1.00	1.00
10	Bowl	1.00	1.00	1.00	**60**	Open	1.00	1.00	1.00
11	Brain	0.89	0.80	0.84	**61**	Owl	1.00	1.00	1.00
12	C	1.00	1.00	1.00	**62**	P	1.00	1.00	1.00
13	Coolie	1.00	1.00	1.00	**63**	Police	1.00	1.00	1.00
14	Cough	0.86	0.86	0.86	**64**	Pray	1.00	0.80	0.89
15	Cow	1.00	1.00	1.00	**65**	Promise	1.00	0.90	0.95
16	Chest	1.00	1.00	1.00	**66**	Q	1.00	1.00	1.00
17	Claw	0.83	0.94	0.88	**67**	R	1.00	0.83	0.91
18	D	1.00	1.00	1.00	**68**	S	0.88	1.00	0.94
19	Devil	0.81	0.92	0.86	**69**	Seven	1.00	1.00	1.00
20	Doctor	1.00	1.00	1.00	**70**	Soldier	1.00	1.00	1.00
21	E	0.25	1.00	0.40	**71**	Shirt	1.00	0.89	0.94
22	East	1.00	1.00	1.00	**72**	Six	0.83	1.00	0.91
23	Eight	0.91	1.00	0.95	**73**	Sick	1.00	0.88	0.94
24	Evening	0.91	1.00	0.95	**74**	Skin	1.00	0.89	0.94
25	Elbow	0.83	1.00	0.91	**75**	Shoulder	1.00	0.83	0.91
26	Eye	1.00	0.80	0.89	**76**	Stand	1.00	1.00	1.00

27	F	1.00	0.77	0.87	**77**	Strong	1.00	1.00	1.00
28	Fat	1.00	1.00	1.00	**78**	Sleep	1.00	1.00	1.00
29	Faith	1.00	1.00	1.00	**79**	Sunday	1.00	1.00	1.00
30	Fever	1.00	1.00	1.00	**80**	T	1.00	1.00	1.00
31	Feel	0.88	0.86	0.87	**81**	Ten	1.00	1.00	1.00
32	Few	1.00	1.00	1.00	**82**	Telephone	0.88	1.00	0.94
33	Food	0.85	0.92	0.88	**83**	Tongue	0.75	1.00	0.86
34	Four	0.67	0.80	0.73	**84**	Thorn	1.00	1.00	1.00
35	Fist	1.00	1.00	1.00	**85**	Three	1.00	0.91	0.95
36	Five	1.00	1.00	1.00	**86**	Trouble	1.00	1.00	1.00
37	G	1.00	1.00	1.00	**87**	Thumbs Up	1.00	0.71	0.83
38	Gun	1.00	1.00	1.00	**88**	Two	1.00	0.75	0.86
39	Good	0.83	0.91	0.87	**89**	U	0.89	1.00	0.94
40	Hair	0.71	0.83	0.77	**90**	V	0.83	1.00	0.91
41	Hand	1.00	0.88	0.94	**91**	W	1.00	1.00	1.00
42	Head	0.86	1.00	0.92	**92**	West	1.00	1.00	1.00
43	Hear	1.00	0.86	0.92	**93**	Wedding	1.00	1.00	1.00
44	I	0.80	1.00	0.89	**94**	Water	1.00	0.90	0.95
45	Jain	1.00	1.00	1.00	**95**	White	0.57	0.57	0.57
46	K	1.00	1.00	1.00	**96**	Word	1.00	1.00	1.00
47	King	1.00	1.00	1.00	**97**	X	1.00	1.00	1.00
48	L	1.00	1.00	1.00	**98**	You	1.00	1.00	1.00
49	Love	1.00	1.00	1.00	**99**	Z	1.00	1.00	1.00
50	Leprosy	0.88	1.00	0.94	**100**	Zero	0.91	1.00	0.95

5.5 Comparative Analysis of different CNN Models using Dataset of Static Signs

The summarized detail of the experiments performed on a static sign dataset, using 100 epochs and MediaPipe technology, is shown in Table 5.12. It has been observed from the results that the VGG19 CNN architecture using MediaPipe outperformed all the other CNN architectures with training and testing accuracy of 99.64% and 96%, respectively.

Table 5.12: Experimental Results for different CNN Architectures and Static sign Dataset

CNN Architecture	Training Accuracy	Test Accuracy
CNN Architecture without using MediaPipe	99.17%	98.80%
CNN Architecture using MediaPipe	95%	93.56%
VGG16 using MediaPipe	97.76%	96.72%
VGG19 using MediaPipe	**99.64%**	**96%**
GoogleNet using MediaPipe	97.75%	96.54%

Table 5.13 Comparative Analysis of Signs for different CNN Architectures w.r.t Precision, Recall and F1 Score

Percentage of Signs w.r.t each model						
		CNN without MediaPipe	**CNN with MediaPipe**	**VGG16**	**VGG19**	**GoogleNet**
	1	58%	71%	76%	83%	72%
	0.90-0.99	41%	4%	6%	10%	7%
	0.80-0.89	0%	19%	12%	2%	15%
	0.70-0.79	1%	3%	3%	3%	3%
	0.60-0.69	0%	1%	2%	2%	2%
Precision	**<0.60**	0%	2%	1%	0%	1%
	1	63%	68%	79%	84%	70%
	0.90-0.99	35%	8%	6%	8%	15%
	0.80-0.89	1%	19%	6%	5%	6%
	0.70-0.79	1%	4%	4%	2%	4%
	0.60-0.69	0%	0%	2%	1%	2%
Recall	**<0.60**	0%	1%	3%	0%	3%
	1	39%	52%	56%	71%	54%
	0.90-0.99	59%	27%	28%	24%	27%
	0.80-0.89	2%	16%	10%	2%	13%
	0.70-0.79	0%	3%	4%	1%	4%
	0.60-0.69	0%	0%	2%	2%	2%
F1-Score	**<0.60**	0%	2%	0%	0%	0%

The performance of all the models is also evaluated using precision, recall and F1-score metrics in Table 5.13. These metrics serve as crucial indicators of each model's ability to classify signs effectively. It has been observed from Table 5.13 that in comparison, the VGG19 model attained a precision score of 1 for 83% signs, indicating its proficiency in precise positive classifications. It achieved a recall score of 1 for 84% of signs, signifying its effectiveness in capturing actual positive instances. The F1 score for VGG19 is 1 for 71% signs, highlighting its balanced performance.

5.6 Training and Testing of CNN Architectures using the dataset of dynamic signs

After experimentation on the static sign dataset, the model has also been developed to recognize dynamic signs. In this architecture, the experiments were performed on a dataset for dynamic signs of videos. The video associated with each sign has been divided into individual frames, and then each frame is pre-processed using the MediaPipe technique. We have collected 9,500 video clips in total for 50 dynamic signs of ISL. The experiments were performed on 25 dynamic signs with a dataset size of 4,750 video clips. The collected video clips were divided into training and test set, out of which 3,800 dynamic signs were used for training the model, and the rest 950 signs were used for testing it. The experiments were performed on different architectures of CNN, as discussed in the following subsection. The experiments are conducted on Google Colaboratory with Adam as an optimizer, learning rate=0.001, decay rate=0.99, batch size=32, and 100 epochs.

5.6.1 VGG16 for Dynamic Sign dataset

VGG16 is one of the pre-trained models of CNN used to implement SLRS. The detailed architecture of VGG16 is discussed in section 5.4.1. We have also implemented VGG16 on a dataset for dynamic signs, and training and test accuracy of 76.78% and 80% have been achieved, respectively, as shown in Figure 5.14.

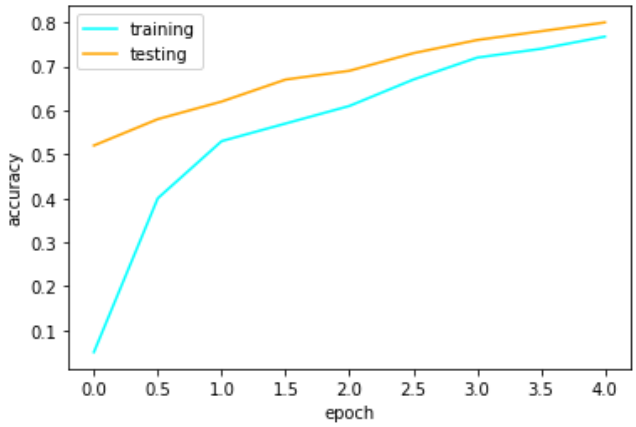


Figure 5.14: Accuracy curve for training and test datasets using VGG16 on a dataset with Dynamic Signs

5.6.2 VGG19 for Dynamic Sign dataset

VGG19 is also a CNN model that is 19 layers deep, as discussed in section 5.4.1. Due to the depth of VGG19, this model is slow in training and produces a model of immense size. This achieved the highest training and test accuracy of 78% and 82%, respectively. The training and test accuracy curve is shown in Figure 5.15.

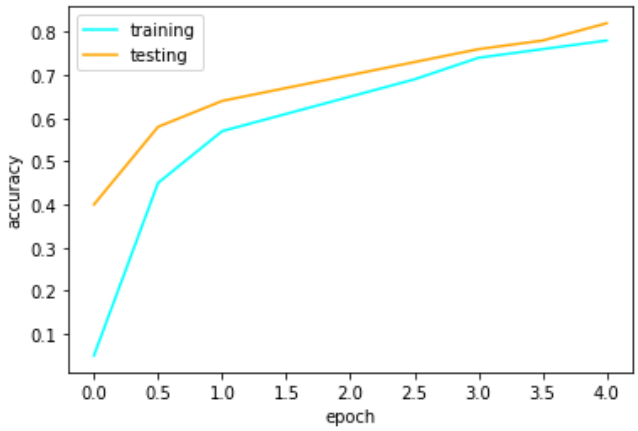


Figure 5.15: Accuracy curve for training and test datasets using VGG19 on a dataset with Dynamic Signs

5.6.3 GoogleNet for Dynamic Sign Dataset

GoogleNet model consists of 22 layers, as discussed in section 5.4.2. In this, an inception module has been introduced to consider how an optimal structure of CNN can be approximated and covered by available dense components. As the model is trained using GoogleNet and Adam as an optimizer, they achieved the highest training and test accuracy of 74% and 78%, respectively, as shown in Figure 5.16.

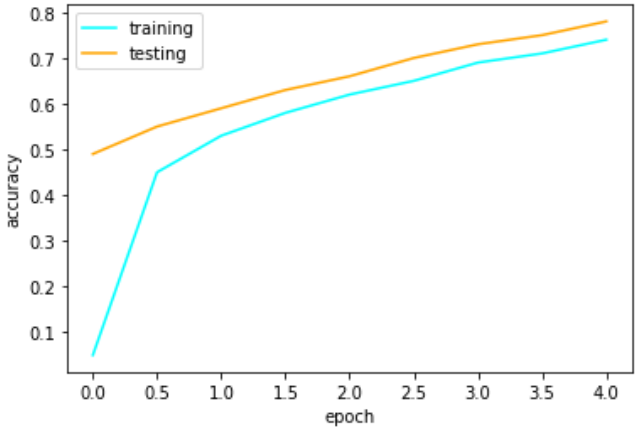


Figure 5.16: Accuracy curve for training and test datasets using GoogleNet on a dataset with Dynamic Signs

5.6.4 CNN Architecture using MediaPipe for Dynamic Sign Dataset

The proposed CNN model consists of six convolutional layers, three max-pooling layers, and three fully connected layers. The total, trainable, and non-trainable parameters taken by the model for the dataset for dynamic signs and the configuration of the CNN layers used in the proposed system are also described in Table 5.14. It has been noted that the total number of parameters in our gathered dataset is 5,050,917. However, only 5,050,213 parameters are used by the model to determine the appropriate weights for minimizing the cost function of the model.

In this architecture, experiments were performed on the model developed using the dataset for dynamic signs and the MediaPipe technique. The collected dataset for dynamic signs has been tested, and the highest training and test accuracy of 95.76% and 86% is achieved, respectively, as shown in Figure 5.17.

Table 5.14: Parameter Configuration of Proposed CNN Architecture using the dataset for dynamic signs and MediaPipe

Layer Type	Output Size	Parameters
input_1 (InputLayer)	[(None, 30, 150, 150, 3)]	0
conv3d (Conv3D)	(None, 30, 150, 150, 12)	984
batch_normalization(BatchNormalization)	(None, 30, 150, 150, 12)	48
conv3d_1 (Conv3D)	(None, 30, 150, 150, 12)	3900
batch_normalization_1(Batchnormalization)	(None, 30, 150, 150, 12)	48
max_pooling3d (MaxPooling3D)	(None, 15, 75, 75, 12)	0
conv3d_2 (Conv3D)	(None, 15, 75, 75, 16)	5200
batch_normalization_2(Batchnormalization)	(None, 15, 75, 75, 16)	64
conv3d_3 (Conv3D)	(None, 15, 75, 75, 16)	6928
batch_normalization_3(Batchnormalization)	(None, 15, 75, 75, 16)	64
max_pooling3d_1 (MaxPooling3D)	(None, 7, 37, 37, 16)	0
conv3d_4 (Conv3D)	(None, 7, 37, 37, 20)	8660
batch_normalization_4(Batchnormalization)	(None, 7, 37, 37, 20)	80
conv3d_5 (Conv3D)	(None, 7, 37, 37, 20)	10820
batch_normalization_5(Batchnormalization)	(None, 7, 37, 37, 20)	80
max_pooling3d_2 (MaxPooling3D)	(None, 3, 18, 18, 20)	0
flatten (Flatten)	(None, 19440)	0
dropout (Dropout)	(None, 19440)	0
dense (Dense)	(None, 256)	4976896
batch_normalization_6(Batchnormalization)	(None, 256)	1024
dropout_1 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 128)	32896
dense_2(Dense)	(None, 25)	3225
Total Parameters:	5,050,917	
Trainable Parameters:	5,050,213	
Non-trainable Parameters:	704	

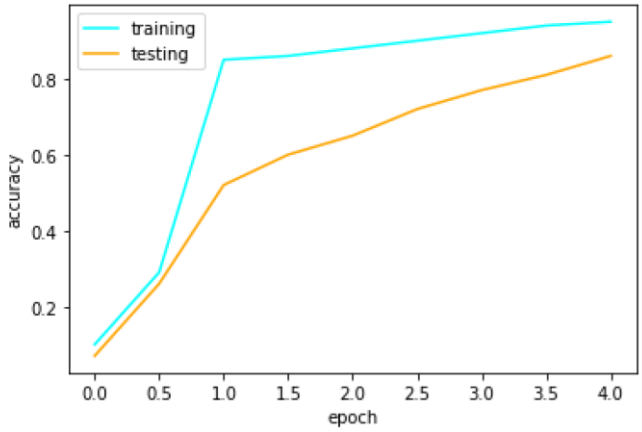


Figure 5.17: Accuracy curve for Training and Test Dataset of Dynamic Sign

The classification performance of CNN architecture using MediaPipe for dynamic signs dataset showing precision, recall, F1 score and prediction rate is shown in Table 5.15. It has been observed from Table 5.15 that among the signs under examination, 68% of signs achieved precision score of 1.00, with 28% of signs showcasing precision scores in range 0.90 to 0.99 and 4% signs exhibited precision scores between 0.70-0.79.

Similarly, concerning recall scores, 56% of signs secured a flawless score of 1.00, while 44% signs fell within the range of 0.90 to 0.99.

In terms of the F1-score, 48% of signs garnered a perfect score of 1.00, another 48% of signs attained scores within the 0.90 to 0.99 range, and 4% signs achieved F1-scores that is between 0.80-0.89.

Table 5.15: Precision, Recall and F1-Score for CNN Architecture using MediaPipe and Dynamic Sign Dataset

S No.	**Sign**	**Precision**	**Recall**	**F1-Score**	**S No.**	**Sign**	**Precision**	**Recall**	**F1-Score**
1	Above	1.00	1.00	1.00	**14**	Calm	1.00	1.00	1.00
2	Absent	0.79	0.97	0.87	**15**	Captain	1.00	1.00	1.00
3	Accept	0.98	0.94	0.96	**16**	Caption	0.99	0.98	0.98
4	Accompany	0.98	1.00	0.99	**17**	Carpet	0.96	0.97	0.96
5	Airplane	1.00	1.00	1.00	**18**	Category	1.00	1.00	1.00
6	Afternoon	1.00	1.00	1.00	**19**	Center	1.00	0.99	0.99
7	Apply	1.00	0.97	0.99	**20**	Certificate	0.97	0.97	0.97

8	Award	1.00	1.00	1.00	**21**	Change	1.00	1.00	1.00
9	Back	1.00	1.00	1.00	**22**	Charity	1.00	1.00	1.00
10	Ball	1.00	1.00	1.00	**23**	Cheque	1.00	0.99	0.99
11	Bank	0.96	0.90	0.93	**24**	Choose	1.00	0.97	0.98
12	Below	1.00	0.96	0.98	**25**	Circle	1.00	1.00	1.00
13	Big	0.95	1.00	0.98					

5.7 Comparative Analysis of different CNN Models using the dataset of dynamic signs

The summarized detail of the experiments performed on a dataset for dynamic signs, using the MediaPipe technique as data pre-processing and Adam as an optimizer, is shown in Table 5.16.

Table 5.16: Experimental Results for different CNN Architectures and Dynamic Sign Dataset

CNN Architecture	Training Accuracy	Test Accuracy
CNN Architecture using MediaPipe (12 layers)	**95.76%**	**86%**
VGG16 using MediaPipe (16 layers)	76.78%	80%
VGG19 using MediaPipe (19 layers)	78%	82%
GoogleNet using MediaPipe (22 layers)	74%	78%

It has been observed from the results that CNN architecture using MediaPipe and 12 layers outperformed all the other CNN architectures with the highest training and test accuracy of 95.76% and 86%, respectively.

From the experimental results, it has been concluded that VGG19 (19 layers) with the MediaPipe technique performs well with static sign datasets compared to all other CNN architectures. On the other hand, the proposed CNN architecture (12 layers) with the MediaPipe technique outperformed dynamic sign datasets compared to VGG and GoogleNet architectures.

Chapter Summary

In this chapter, the overall design and implementation of the SLRS have been described for the recognition of static and dynamic signs. This chapter leads to the implementation of SLRS for both datasets. A detailed description of CNN architectures has been discussed.

Further, this chapter has also compared the proposed CNN architecture with other CNN-based architectures like VGG16, VGG19, and GoogleNet for static signs. Various parameter configurations were tested to evaluate the performance of the CNN model. This chapter also describes the CNN model's parameter settings, including the number of convolutional layers, hidden layers, fully connected layers, number of filters, filter size, optimizer, and dropout.

The comparison of different CNN architectures for dynamic signs has also been explained in this chapter. After successful implementation of all the CNN models using MediaPipe technology and dynamic sign dataset, it has been observed that the proposed CNN model with 12 layers outperformed all other CNN architectures.

It has been observed that the VGG19 CNN architecture using MediaPipe for recognition of static signs outperformed all the other architectures. So, we have decided to deploy our model using VGG19 architecture for the web and mobile-based applications. The results given by the proposed SLRS have been studied using different performance measures such as accuracy, precision, recall, and F1-Score.

CHAPTER 6

User Interface

This chapter explains the Graphical User Interface (GUI) of the developed SLRS for static signs and its features.

6.1 Web/Mobile Based Application

In the proposed work, the web and mobile based application was developed for recognition of static signs in real-time. The block diagram of web based and mobile based application is shown in Figure 6.1. In this system, the user should submit an image file with supported extensions like jpg and png or he can capture the image using web/mobile camera. This image file is fed into the recognition interface for pre-processing using MediaPipe. Further, the pre-processed data is passed to the trained model for recognition. The trained deep learning model predicts the class of the input image and display its output in the form of text and voice.

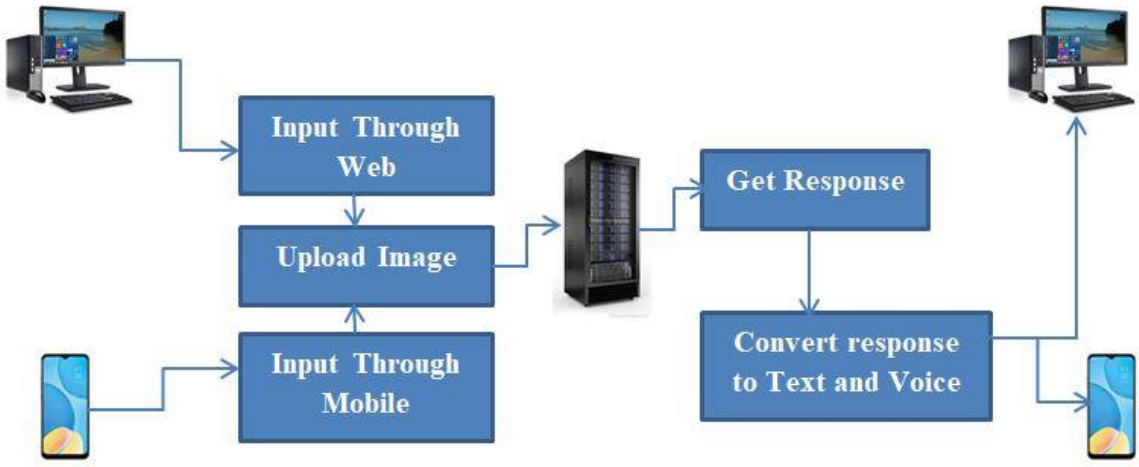


Figure 6.1: Block Diagram of Web/Mobile Application

6.2 Tools and Technology Used

Understanding neural networks were the primary focus for the development of SLRS to recognize Indian signs. Various technologies used for the implementation of SLRS are described as follows.

6.2.1 TensorFlow

TensorFlow is a free and open-source software framework for data flow and differentiable programming in various applications.

6.2.2 OpenCV

OpenCV (Open-Source Computer Vision Library) consists of programming functions oriented primarily toward real-time computer vision. The library is platform-independent and free to use.

6.2.3 Keras

Python-based Keras is an open-source neural network library. It prioritizes user-friendliness, modularity, and extensibility to facilitate rapid experimentation with deep neural networks.

6.2.4 NumPy

NumPy is a library of python used to provide a support for multidimensional arrays, matrices and a number of high-level mathematical functions.

6.3 Data Flow Diagrams

Data Flow Diagrams (DFD) describe the flow of the implemented system. Different levels of DFDs exist starting from level0 level 1, level 2, ….., and level n. The level 0 DFD is shown in Figure 6.2, level 1 in Figure 6.3, level 2 in Figure 6.4.

Level 0 DFD is used to represent the complete system for sign language recognition of static signs. In the developed SLRS the image will be captured by using web or mobile camera and is forwarded to the application for matching. Further, the system will send the response to the user in the form of text and speech.

Level 1 DFD is shown in Figure 6.3, in which deep learning technique has been implemented to predict the result of the uploaded sign image. The uploaded image will be recognized by matching the static sign symbol with the dataset stored on Microsoft Azure server.

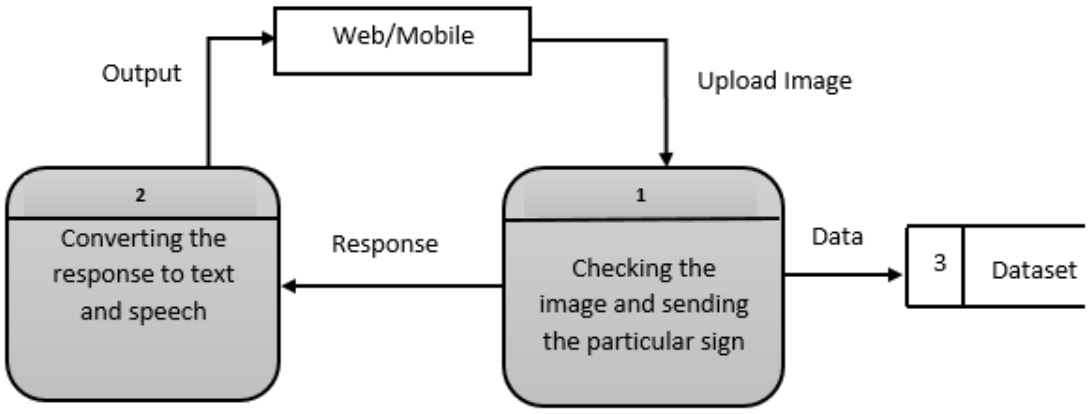


Figure 6.2: Level0 DFD

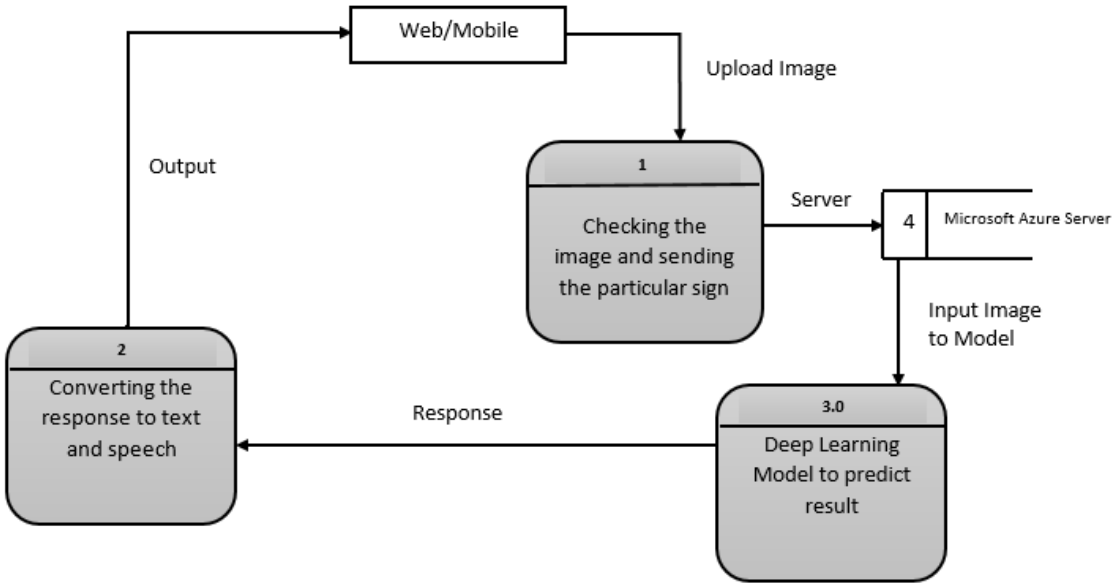


Figure 6.3: Level 1 DFD

Level 2 DFD is shown in Figure 6.4, in which the detailed levels of all the steps are described. This DFD represents the detailed description starting from sign uploading, feature extraction, matching of sign with the trained model, sign recognition and output in the form of text and voice.

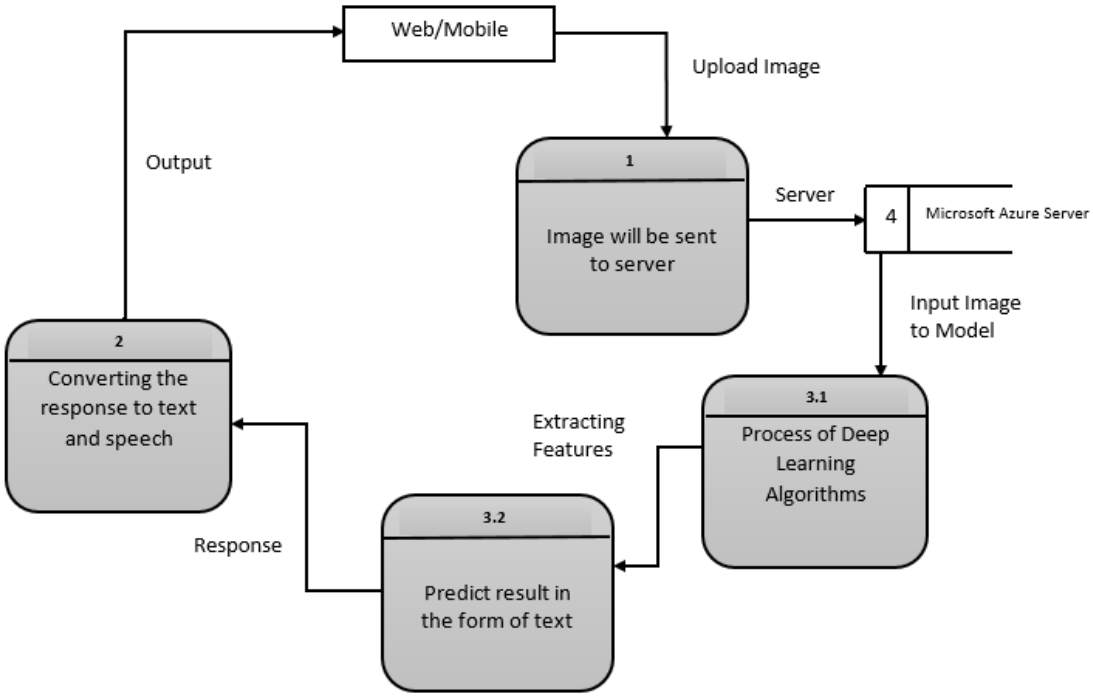


Figure 6.4: Level 2 DFD

6.4 Web-Based Graphical User Interface

The web-based SLRS runs on any device containing a browser. It is a progressive web-based SLRS used to identify static, manual signs. The landing page of our developed platform is shown in Figure 6.5 and the footer of a landing page is shown in Figure 6.6, in which the markers used by body posture, hands, and face are represented.

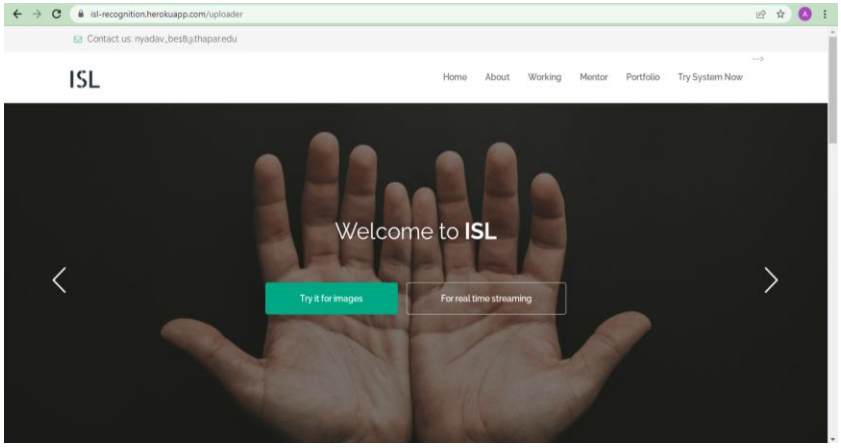


Figure 6.5: Landing Page of Web-based SLRS

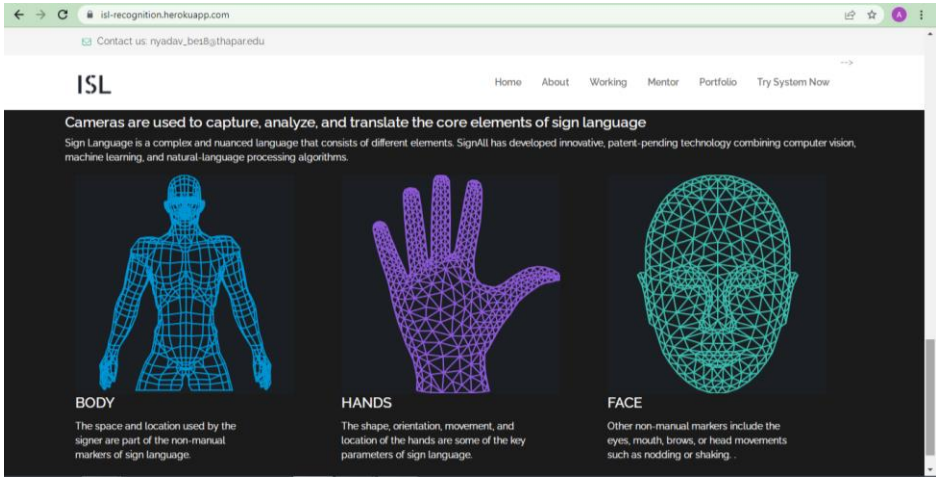


Figure 6.6: Footer of the Landing Page

The brief description of the system's working is shown in Figure 6.7. The complete interface used by SLRS from uploading of an image to the recognition process is shown in Figure 6.8.

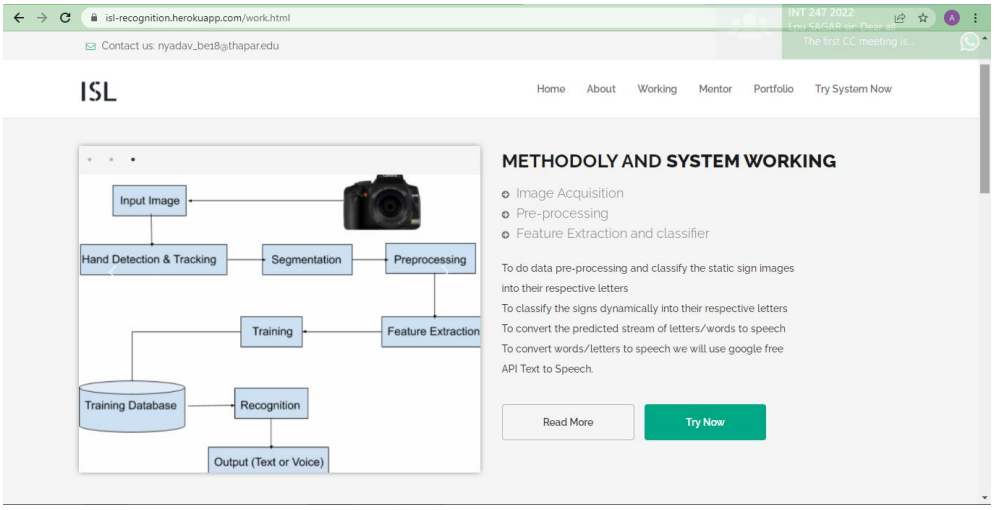
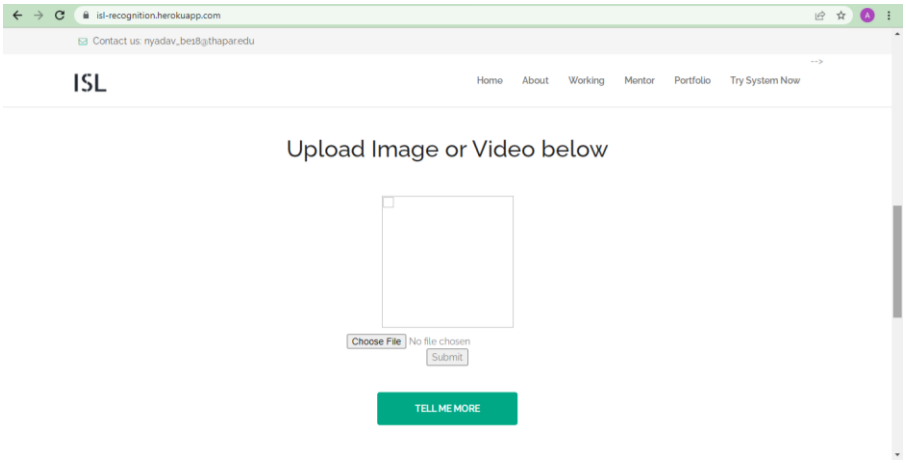
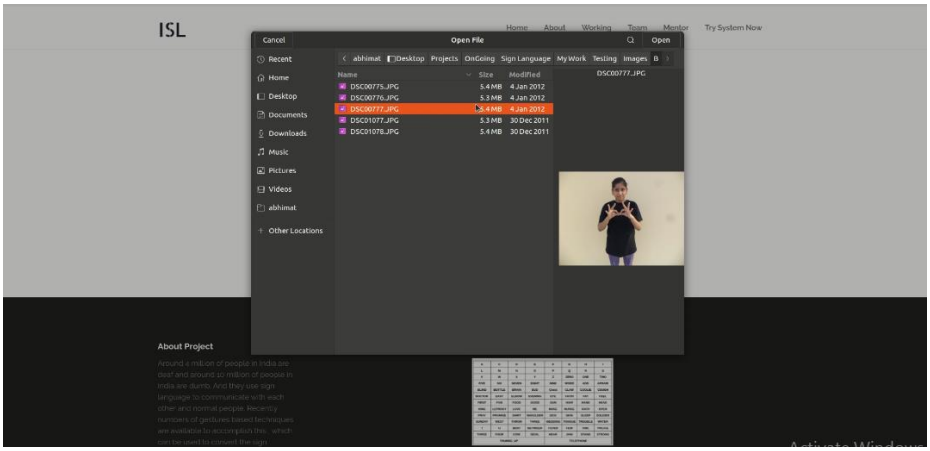


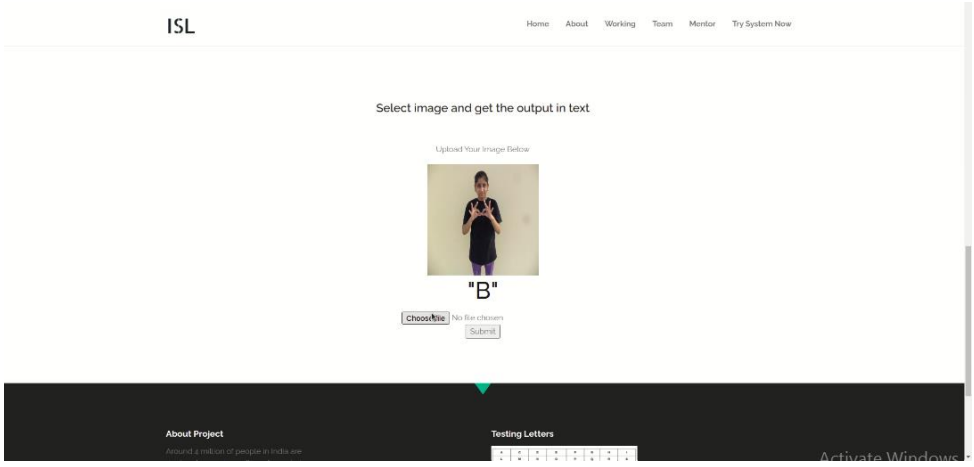
Figure 6.7: The 'Working' Page



(a) Choose an image of a sign for recognition.



(b) Select Image for Input

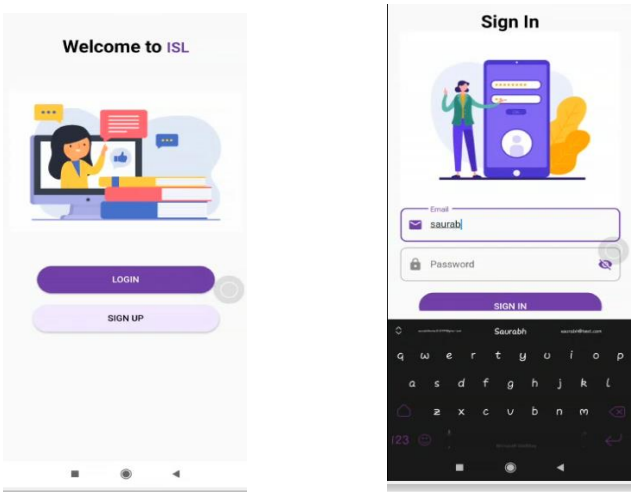


(c) Recognized sign displayed as text

Figure 6.8: Complete SLRS

6.5 Mobile Based Graphical User Interface

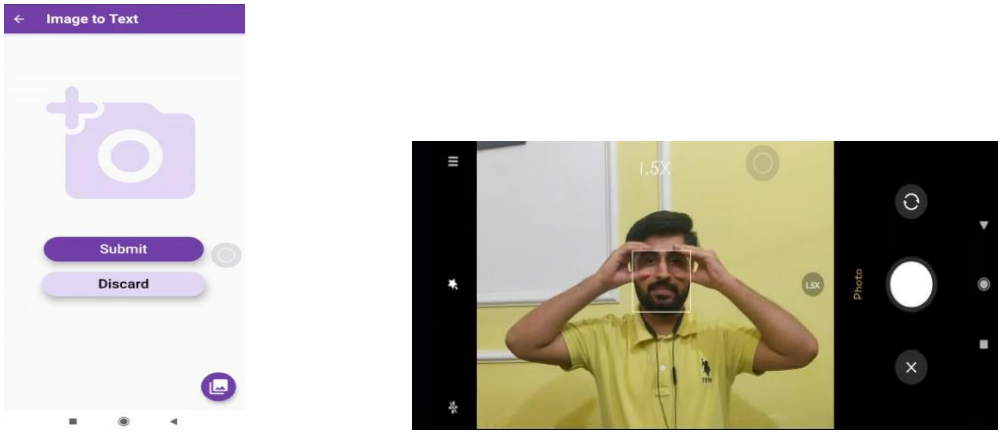
The mobile-based SLRS can run on any android phone. To make this application run, we need to install the application from the play store. This interface has been developed using the Azure platform. The end users have to install this application to recognize signs on their android phones. The landing page and the login page for SLRS are shown in Figure 6.9 below.



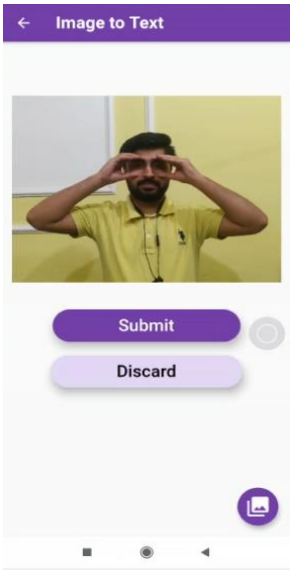
(a) Landing Page (b) Login Page

Figure 6.9: Screenshots of Mobile Application

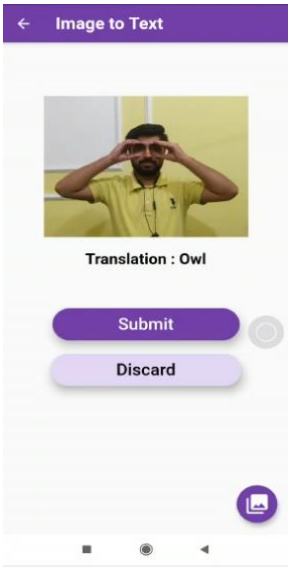
The interface used for the recognition of static, manual signs of Indian sign language is shown in Figure 6.10 below.



(a) Select/Capture Image (b) Capture Image



(c) Submit Image



(d) Recognized Image

Figure 6.10: Screenshots of Mobile Application for Recognition of Static Signs

6.6 Strengths of developed GUI

GUI itself is a standard component of user-facing applications, its design and functionality play a pivotal role in enhancing the overall user experience and, consequently, the efficacy of our sign language recognition system. The research contributions of the GUI are outlined as follows.

i. Accessibility and Inclusivity: The work has focused on user-centric design principles, conducting usability studies to create an interface that caters to the specific needs of sign language users.

ii. Integration and Interoperability: The GUI facilitates seamless integration with the core sign language recognition system, enabling users to access and utilize the technology effortlessly. Research contributions include the development of effective communication protocols and APIs to ensure the GUI's interoperability with the recognition engine.

Although, the GUI may appear as a conventional component of a web application or mobile app, its design, features, and user-centric research contributions are integral to the success of our sign language recognition system. By focusing on accessibility, inclusivity, and integration, the work aims to provide an inclusive and effective tool that significantly enhances communication for sign language users.

6.7 Comparison of the proposed system with existing systems

The proposed system offers several distinctive advantages that set it apart from the current state of the art in sign language recognition. Various distinct features of the proposed model are described as follows.

i. Higher Accuracy and Precision: The proposed system has been meticulously trained on a vast dataset of sign language gestures, which has resulted in significantly higher accuracy and precision in recognizing and translating sign language into text or speech. This improved accuracy directly benefits the deaf and hard-of-hearing community.

ii. Hand Tracking and Posture Recognition: The proposed system makes use of MediaPipe technique for tracking hands and postures. MediaPipe's hand tracking is optimized for real-time performance, making it suitable for sign language recognition.

iii. Real-time Recognition: Unlike some existing systems that may have latency issues, our system is optimized for real-time recognition. It can quickly interpret signs and provide immediate feedback, facilitating seamless and natural communication.

iv. User-Friendly Interface: In this work an intuitive and user-friendly interface has been developed, which encourages wider adoption among both signers and non-signers.

v. Robustness to Environmental Factors: The proposed system has been trained to recognize signs in various lighting conditions, backgrounds, and distances, making it more robust for real-world applications.

Table 6.1 presents the comparison of state-of-the-art works with the proposed model. It has been observed that most of the earlier researchers worked on alphabets only [165] [174]. In the proposed work, total 100 number of signs are used including 23 alphabets, 10 digits and 67 words. Although the accuracy on small datasets [166] used by other researches was approximately 99% using machine learning models, the proposed work has achieved 96% of accuracy on 100 signs using VGG19 which is a deep learning model. Furthermore, the proposed strategy performs effectively in a variety of lighting situations and it is also free of feature occlusion due to the existence of both one-hand and two-hand signs.

Table: 6.1 Comparative Analysis of the proposed ISL recognition system with existing work

Author(s)	Number of Signs	Single/Double Handed Signs	Methodology Used	Average Accuracy (%)
Rao and Kishore (2018) [148]	10 Sentences	Both	Adaboost	90%
Shenoy *et al*.(2018) [166]	45 signs (23 Alphabets, 10 digits and 12 gestures)	Both	KNN (Hand Pose) and HMM (Gestures)	99.7%(Hand Pose), 97.23%(Gesture)
Sharma *et al.*(2021) [164]	26 Alphabets	Both	SVM, VGG16	98.52% (Single handed), 97% (Double handed)
Srivastava *et al.* (2021) [210]	26 Alphabets	Both	MobileNet v2	85.45%
Proposed	100 Signs (23 Alphabets, 10 digits and 67 words)	Both	VGG19	96%

The proposed sign language recognition system represents a significant advancement in technology aimed at enhancing communication for the deaf and hard-of-hearing community. In this, a person must cope with words and numbers in addition to alphabets, which makes communication more effective for real-world scenarios.

Chapter Summary

This chapter presents the detailed online web and mobile based Graphical User Interface (GUI) of the SLRS for Indian Sign Language. These systems perform the recognition of static, manual signs. It takes the image as input, pre-processes it, and matches it with the dataset available using the trained model to recognize static signs. The usage of tools and technologies, flow diagrams, and class diagrams used by the SLRS are also presented in this chapter.

CHAPTER 7

Conclusion and Future Scope

The main objective of this thesis is to develop a SLRS to recognize static and dynamic signs of Indian sign language in which, our major contribution is to create the dataset for sign language recognition and develop the system for recognition of static signs and dynamic signs in real-time. This chapter concludes the research work presented in this thesis and presents the future scope of the work.

7.1 Conclusion

As no public dataset is available for the development of Indian SLRS, it is required to create the dataset for the same. The developed SLRS has the capability to classify static and dynamic signs too. After this, a systematic literature review of different sign languages belonging to different countries has been performed. A proper methodology, including planning, conducting, and documenting the review, was followed to perform the detailed literature review. The review was conducted by finding relevant research articles from prestigious electronic databases and leading conferences and publications in this field. To further limit the number of chosen studies, inclusion/exclusion criteria were used, and retrieved research papers were organized according to their publication years. Different sources for journals, conferences, and workshops were used for downloading the research articles. The systematic literature review has been categorized based mainly on six parameters: data acquisition, single/double-handed signs, static/dynamic signs, isolated/continuous signs, a technique used for sign classification, and the model's accuracy. In this review, each of these parameters has been discussed for different sign languages worldwide. The analysis of the review is graphically represented by using pie charts.

The main problem encountered in developing SLRS is the unavailability of the sign language dataset. Some datasets are available publically for sign language, but none of them belong to Indian sign language. So, to implement a SLRS, we must collect and develop the dataset of our own for sign images and video clips. The SLRS in this thesis is

proposed, and two datasets were collected and developed for recognizing signs of Indian sign language. The dataset for static signs consists of 35,000 static signs, and the dataset for dynamic signs consists of 9,500 videos of the Indian sign language. Both the datasets were collected in different environmental conditions with varying distances of a person and a camera. The description of how signs are categorized as single-handed signs, double-handed signs, facial expressions, single-handed signs with a face, and double-handed signs with the face is also mentioned. To collect the signs, all the subjects were appropriately trained. Since none of the participants were used to sign language and the signs are too distinct from one another, this training and recording procedure requires substantial work. The collected sign images and videos are then pre-processed using the MediaPipe technique. Two MediaPipe techniques, which are MediaPipe hands and MediaPipe pose, were described in detail.

Three approaches have been proposed in this thesis work. To detect static signs, a simple CNN model has been proposed. In this approach, approximately 50 experiments were performed by changing the number of layers, filter size, epochs, and optimizers. The experiments were also performed on colored and greyscale images, and it has been found that the model outperformed with greyscale image data and SGD as an optimizer. The proposed model excelled on training and validation datasets, but it became highly overfitted when we tested it for the unseen dataset. To overcome the drawback of the first proposed approach, a second CNN-based model has been proposed. This model was based on a static sign dataset and the MediaPipe technique. MediaPipe hand and MediaPipe pose detection techniques were used to pre-process collected static sign images. After this, the third approach that has been proposed for the recognition of dynamic signs using video clips is discussed. The dataset for dynamic signs has been collected and developed for the recognition of dynamic signs. The media pipe pose detection technique has been embedded with the CNN architecture for the recognition of dynamic sign video clips is also explained. Finally, the proposed models were tested on the collected dataset and unseen data based on performance metrics and compared to the state-of-the-art models.

To conclude the thesis, an advanced real-time web-based and mobile-based application is developed to recognize Indian signs that help hearing-impaired people communicate with others in society. The results show that the system has achieved satisfactory performance with a high classification accuracy of Indian signs under different illumination conditions and at different distances.

7.2 Future Scope

The proposed system recognizes the signs of Indian sign language using web-based and mobile-based applications in real-time. For futuristic work, it is required to extend the dataset for the recognition of static and dynamic signs. In addition to this, the developed SLRS is responsible for recognizing static signs in web and mobile based application and it is also required to develop a system to recognize dynamic signs in real-time.

The proposed system can be used as a basis for commercial SLRSs. We can extend this system for domain-specific applications like railway stations, restaurants, hotels, airports, etc. This system helps hearing-impaired persons learn new facts and allow them to come forward and work in one place to become independent people in society. Also, we plan to study more recognition techniques and build an in-depth solution for sign language recognition.

In future, the proposed SLRS can also be extended to remove real-time challenges like occlusion by other body parts and between signer variations.

List of Publications

1. Ankita Wadhawan and Parteek Kumar, “Sign Language Recognition Systems: A Decade Systematic Literature Review”, Archives of Computational Methods in Engineering (IF: 7.242), pp. 1-29, 17 December 2019. (https://doi.org/10.1007/s11831-019-09384-2).
2. Ankita Wadhawan and Parteek Kumar, "Deep Learning-based Sign Language Recognition System for Static Signs", Neural Computing and Applications (IF:4.664), pp. 1-12, 01, January 2020. (https://doi.org/10.1007/s00521-019-04691-y).

References

[1] Abdel-Samie, A.G.A.R., Elmisery, F.A., Brisha, A.M. and Khalil, A.H. (2018) Arabic sign language recognition using kinect sensor. *Research Journal of Applied Sciences, Engineering and Technology*, 15(2), pp.57-67.

[2] Abraham, E., Nayak, A. and Iqbal, A. (2019) October. Real-Time Translation of Indian Sign Language using LSTM. In *2019 Global Conference for Advancement in Technology (GCAT)* (pp. 1-5). IEEE.

[3] Abreu J. G, Teixeira J. M, Figueiredo L. S and Teichrieb V (2016) Evaluating Sign Language Recognition Using the Myo Armband. *In: XVIII IEEE Symposium on Virtual and Augmented Reality (SVR)*, pp. 64-70.

[4] Adhan S and Pintavirooj C (2016) Thai sign language recognition by using geometric invariant feature and ANN classification. *In: 9th IEEE International Conference on Biomedical Engineering (BMEiCON)*, pp. 1-4.

[5] Adithya V, Vinod P. R, and Gopalakrishnan U (2013) Artificial neural network based method for Indian sign language recognition. *In: IEEE Conference on Information & Communication Technologies (ICT)*, pp. 1080-1085.

[6] Admasu Y. F and Raimond K (2010) Ethiopian sign language recognition using Artificial Neural Network. *In: 10th International Conference on Intelligent Systems Design and Applications (ISDA)*, pp. 995-1000.

[7] Agarwal A and Thakur M. K (2013) Sign language recognition using Microsoft Kinect. *In: Sixth IEEE International Conference on Contemporary Computing (IC3)*, pp. 181-185.

[8] Agrawal S. C, Jalal A. S and Bhatnagar C (2012) Recognition of Indian Sign Language using feature fusion. *In: 4th IEEE International Conference on Intelligent Human Computer Interaction (IHCI)*, pp. 1-5.

[9] Ahmed A. A and Aly S (2014) Appearance-based arabic sign language recognition using hidden markov models. *In: IEEE International Conference on Engineering and Technology (ICET)*, pp. 1-6.

[10] Ahmed S. T and Akhand M. A. H (2016) Bangladeshi Sign Language Recognition using fingertip position. *In: IEEE International Conference on Medical Engineering, Health Informatics and Technology (MediTec)*, pp. 1-5. IEEE.

[11] Ahmed W, Chanda K and Mitra S (2016) Vision based Hand Gesture Recognition using Dynamic Time Warping for Indian Sign Language. *In: IEEE International Conference on Information Science (ICIS)*, pp. 120-125.

[12] Akmeliawati R, Ooi M. P. L and Kuang Y. C (2007) Real-time Malaysian sign language translation using colour segmentation and neural network. *In: IEEE Conference Proceedings on Instrumentation and Measurement Technology, IMTC*, pp. 1-6.

[13] Almasre, M.A. and Al-Nuaim, H., 2020. A comparison of Arabic sign language dynamic gesture recognition models. *Heliyon*, 6(3), p.e03554.

[14] AlQattan D, and Sepulveda F (2017) Towards sign language recognition using EEG-based motor imagery brain computer interface. *In: 5th IEEE International Winter Conference on Brain-Computer Interface (BCI)*, pp. 5-8.

[15] 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.

[16] Aly, S. and Aly, W. (2020). DeepArSLR: A novel signer-independent deep learning framework for isolated arabic sign language gestures recognition. *IEEE Access*, 8, pp.83199-83212.

[17] Alzohairi, R., Alghonaim, R., Alshehri, W., Aloqeely, S., Alzaidan, M. and Bchir, O. (2018). Image based Arabic sign language recognition system. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 9(3).

[18] Amrutha CU, Davis N, Samrutha KS, Shilpa NS, Chunkath J (2016) Improving language acquisition in sensory deficit individuals with mobile application. *Procedia Technol* 24:1068–1073.

[19] Arsalan, M., Kim, D. S., Owais, M., & Park, K. R. (2020) OR-Skip-Net: Outer residual skip network for skin segmentation in non-ideal situations. *Expert Systems with Applications, 141*, 112922.

[20] Aryanie D and Heryadi Y (2015). American sign language-based finger-spelling recognition using k-Nearest Neighbors classifier. *In: 3rd IEEE International Conference on Information and Communication Technology (ICoICT)*, pp. 533-536.

[21] Assaleh K, Shanableh T, Fanaswala M, Bajaj H and Amin F (2008) Vision-based system for continuous Arabic Sign Language recognition in user dependent mode. *In: 5th IEEE International Symposium on Mechatronics and Its Applications, ISMA*, pp. 1-5.

[22] Athira, P.K., Sruthi, C.J. and Lijiya, A. (2019). A signer independent sign language recognition with co-articulation elimination from live videos: an Indian scenario. *Journal of King Saud University-Computer and Information Sciences*, Vol. 34(3), pp. 771-781.

[23] Azar S. G and Seyedarabi H (2016) Continuous Hidden Markov Model based dynamic Persian Sign Language recognition. *In: 24th IEEE Iranian Conference on Electrical Engineering (ICEE)*, pp. 1107-1112.

[24] Badhe PC, Kulkarni V (2015) Indian Sign Language translator using gesture recognition algorithm. *In: Proceedings of IEEE international conference on computer graphics on vision and inform* pp. 195-200. IEEE.

[25] Badhe, Purva C., and Vaishali Kulkarni. (2015) Indian sign language translator using gesture recognition algorithm. In *2015 IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS)*, pp. 195-200. IEEE.

[26] Bantupalli, Kshitij, and Ying Xie (2018) American sign language recognition using deep learning and computer vision. In *2018 IEEE International Conference on Big Data (Big Data)*, pp. 4896-4899. IEEE.

[27] Basiri, Salar, Alireza Taheri, Ali F. Meghdari, Mehrdad Boroushaki, and Minoo Alemi (2021) Dynamic iranian sign language recognition using an optimized deep neural network: an implementation via a robotic-based architecture. *International Journal of Social Robotics (2021*): 1-21.

[28] Bhagat, N.K., Vishnusai, Y. and Rathna, G.N. (2019), December. Indian sign language gesture recognition using image processing and deep learning. In *2019 Digital Image Computing: Techniques and Applications (DICTA)* (pp. 1-8). IEEE.

[29] Boostlingo Blog, Available: https://boostlingo.com/2021/04/01/6-sign-language-families-and-where-theyre-used/, [Accessed: 01, April 2021].

[30] Chikkanna M., and Guddeti R. M. R. (2013) Kinect based real-time gesture spotting using HCRF," in *Proc. of IEEE Int. Conf. on Advances in Computing, Communications and Informatics (ICACCI)*, Mysore, India, pp. 925-928.

[31] Chinese Sign Language. [Accessed Online: 16, March 2018], https://en.wikipedia.org/wiki/Chinese_Sign_Language.

[32] Chong, T.W. and Lee, B.G. (2018) American sign language recognition using leap motion controller with machine learning approach. *Sensors*, *18*(10), p.3554.

[33] Chuan C. H, Regina E, and Guardino C (2014). American sign language recognition using leap motion sensor. *In: 13th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 541-544.

[34] Dahmani D and Larabi S (2014) User-independent system for sign language finger spelling recognition. *Journal of Visual Communication and Image Representation*, 25(5), 1240-1250.

[35] Darwish S. M (2017) Man-Machine Interaction System for Subject Independent Sign Language Recognition. *In: Proceedings of the 9th International Conference on Computer and Automation Engineering*, *ACM*, pp. 121-125.

[36] Dasgupta T, Shukla S, Kumar S, Diwakar S, Basu A (2008) A multilingual multimedia Indian Sign Language dictionary tool, *In: Proceedings of international joint conference on natural language processing, Hyderabad, India*, pp 57–64.

[37] Davydov M. V, Nikolski I. V and Pasichnyk V. V (2010) Real-time Ukrainian sign language recognition system. *In: IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS)*, 1, pp. 875-879.

[38] De Paula Neto F. M, Cambuim L. F, Macieira R. M, Ludermir T. B, Zanchettin C and Barros E. N (2015) Extreme Learning Machine for Real Time Recognition of Brazilian Sign Language. *In: IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1464-1469.

[39] Deriche, M., Aliyu, S.O. and Mohandes, M. (2019) An intelligent arabic sign language recognition system using a pair of LMCs with GMM based classification. *IEEE Sensors Journal*, 19(18), pp.8067-8078.

[40] Dias D. B, Madeo R. C, Rocha T, Bíscaro H. H and Peres S. M (2009) Hand movement recognition for brazilian sign language: a study using distance-based neural networks. *In: IEEE International Joint Conference on Neural Networks, IJCNN*. pp. 697-704.

[41] Dipietro L., Sabatini A. M., and Dario P. (2003) Evaluation of an instrumented glove for hand-movement acquisition. *Journal of rehabilitation research and development*, 40 (2), pp. 179–190.

[42] Dour S, Kundargi M (2013) Design of ANFIS system for recognition of single hand and two hand signs for Indian SignLanguage. *Int J Appl Inf Syst*, pp 18–25.

[43] Eberhard, David M.; Simons, Gary F.; Fennig, Charles D., eds. (2021), "Sign language", *Ethnologue: Languages of the World (24th ed.), SIL International*, retrieved 2021-05-15.

[44] Elons A. S, Ahmed M, Shedid H and Tolba M. F (2014) Arabic sign language recognition using leap motion sensor. *In: 9th IEEE International Conference on Computer Engineering & Systems (ICCES)*, pp. 368-373.

[45] Fahn C.S., and Sun H. (2005) Development of a data glove with reducing sensors based on magnetic induction, *IEEE Trans. on Industrial Electronics*, 25(2), pp. 585–594.

[46] Ferreira P. M, Cardoso J. S and Rebelo A (2017). Multimodal Learning for Sign Language Recognition. *In: Iberian Conference on Pattern Recognition and Image Analysis, Springer, Cham*, pp. 313-321.

[47] Galicia R, Carranza O, Jiménez E. D and Rivera G. E (2015) Mexican sign language recognition using movement sensor. *In: 24th IEEE International Symposium on Industrial Electronics (ISIE)*, pp. 573-578.

[48] Gangrade, J., Bharti, J. and Mulye, A. (2020). Recognition of Indian sign language using ORB with bag of visual words by Kinect sensor. *IETE Journal of Research*, pp.1-15.

[49] García-Bautista G, Trujillo-Romero F and Caballero-Morales S. O (2017) Mexican sign language recognition using kinect and data time warping algorithm. *In: International Conference on Electronics, Communications and Computers (CONIELECOMP),* pp. 1-5.

[50] Garg, P., Aggarwal, N. and Sofat, S. (2009) Vision based hand gesture recognition. *International Journal of Computer and Information Engineering*, *3*(1), pp.186-191.

[51] Geng L, Ma X, Wang H, Gu J and Li Y (2014) Chinese sign language recognition with 3D hand motion trajectories and depth images. *In: 11th IEEE World Congress on Intelligent Control and Automation (WCICA),* pp. 1457-1461.

[52] Gruber, Ivan, Zdenek Krnoul, Marek Hruz, Jakub Kanis, and Matyas Bohacek (2021) Mutual Support of Data Modalities in the Task of Sign Language Recognition. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* pp. 3424-3433.

[53] Guo D, Zhou W, Li H and Wang M (2017) Online early-late fusion based on adaptive HMM for sign language recognition. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 14(1), 8.

[54] Gurbuz, S.Z., Gurbuz, A.C., Malaia, E.A., Griffin, D.J., Crawford, C.S., Rahman, M.M., Kurtoglu, E., Aksu, R., Macks, T. and Mdrafi, R. (2020). American sign language recognition using rf sensing. *IEEE Sensors Journal*, *21*(3), pp.3763-3775.

[55] Hamed A, Belal N. A and Mahar K. M (2016) Arabic Sign Language Alphabet Recognition Based on HOG-PCA Using Microsoft Kinect in Complex Backgrounds. *In: IEEE 6th International Conference on Advanced Computing (IACC),* pp. 451-458.

[56] Hassan M, Assaleh K and Shanableh T (2016) User-Dependent Sign Language Recognition Using Motion Detection. *In: IEEE International Conference on Computational Science and Computational Intelligence (CSCI),* pp. 852-856.

[57] Hasan M, Sajib T. H and Dey M (2016) A machine learning based approach for the detection and recognition of Bangla sign language. *In: IEEE International Conference on Medical Engineering, Health Informatics and Technology (MediTec),* pp. 1-5.

[58] Hayani, S., Benaddy, M., El Meslouhi, O. and Kardouchi, M. (2019), July. Arab sign language recognition with convolutional neural networks. In 2019 *International Conference of Computer Science and Renewable Energies (ICCSRE*) (pp. 1-4). IEEE.

[59] "Hearing Loss: Determining Eligibility for Social Security Benefits." Available: https://www.ncbi.nlm.nih.gov/books/NBK207836/.

[60] Hisham, B. and Hamouda, A. (2021). Arabic sign language recognition using Ada-Boosting based on a leap motion controller. *International Journal of Information Technology*, 13(3), pp.1221-1234.

[61] Hosoe H, Sako S and Kwolek B (2017) Recognition of JSL finger spelling using convolutional neural networks. *In: IEEE Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*, pp. 85-88.

62] Hossen, M.A., Govindaiah, A., Sultana, S. and Bhuiyan, A. (2018) June. Bengali sign language recognition using deep convolutional neural network. In *2018 joint 7th international conference on informatics, electronics & vision (iciev) and 2018 2nd international conference on imaging, vision & pattern recognition (icIVPR)*, pp. 369-373. IEEE.

[63] Huang C. L and Tsai B. L (2010) A vision-based Taiwanese sign language Recognition. *In: 20th IEEE International Conference on Pattern Recognition (ICPR)*, pp. 3683-3686.

[64] Huang, J., Zhou, W., Li, H. and Li, W. (2018). Attention-based 3D-CNNs for large-vocabulary sign language recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(9), pp.2822-2832.

[65] Infantino I, Rizzo R and Gaglio S (2007) A framework for sign language sentence recognition by commonsense context. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(5), 1034-1039.

[66] Islam M. M, Siddiqua S and Afnan J (2017) Real time Hand Gesture Recognition using different algorithms based on American Sign Language. *In: IEEE International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, pp. 1-6.

[67] Jangyodsuk P, Conly C and Athitsos V (2014). Sign language recognition using dynamic time warping and hand shape distance based on histogram of oriented gradient features. *In: Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments*, ACM, pp. 50.

[68] Jiang, Songyao, Bin Sun, Lichen Wang, Yue Bai, Kunpeng Li, and Yun Fu. (2021) Skeleton aware multi-modal sign language recognition. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3413-3423. 2021.

[69] Jiang, X., Lu, M. and Wang, S.H. (2020). An eight-layer convolutional neural network with stochastic pooling, batch normalization and dropout for fingerspelling recognition of Chinese sign language. *Multimedia Tools and Applications*, 79(21), pp.15697-15715.

[70] Jiménez L. A. E, Benalcázar M. E and Sotomayor N (2017) Gesture Recognition and Machine Learning Applied to Sign Language Translation. *In: VII Latin American Congress on Biomedical Engineering CLAIB, Springer, Singapore, Bucaramanga, Santander, Colombia*, October 26th-28th, 2016, pp. 233-236.

[71] Joze, H.R.V. and Koller, O. (2018). Ms-asl: A large-scale data set and benchmark for understanding American sign language. *arXiv preprint arXiv:1812.01053*.

[72] Kamruzzaman, M.M. (2020). Arabic sign language recognition and generating Arabic speech using convolutional neural network. *Wireless Communications and Mobile Computing*.

[73] 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.

[74] Karayılan T and Kılıç Ö (2017) Sign language recognition. *In: IEEE International Conference on Computer Science and Engineering (UBMK)*, pp. 1122-1126.

[75] Kelly D, Delannoy J. R, Mc Donald J and Markham C (2009a) Incorporating facial features into a multi-channel gesture recognition system for the interpretation of irish sign language sequences. *In: IEEE 12th International Conference on Computer Vision Workshops (ICCV)*, pp. 1977-1984.

[76] Kelly D, McDonald J and Markham C (2010) A person independent system for recognition of hand postures used in sign language. *Pattern Recognition Letters*, 31(11), 1359-1368.

[77] Kelly D, Reilly Delannoy J, Mc Donald J and Markham C (2009b) A framework for continuous multimodal sign language recognition. *In: Proceedings of the 2009 international conference on Multimodal interfaces, ACM*, pp. 351-358.

[78] Kessler G. D., Hodges L., and Walker N. (1995) Evaluation of the Cyber glove as a whole hand input device," *ACM Trans. Computer-Human Interaction (TOCHI)*, 2(4), pp. 263–283.

[79] Khomami, S.A. and Shamekhi, S. (2021). Persian sign language recognition using IMU and surface EMG sensors. *Measurement*, 168, p.108471.

[80] Kim J, Wagner J, Rehm M and André E (2008) Bi-channel sensor fusion for automatic sign language recognition. *In: 8th IEEE International Conference on Automatic Face & Gesture Recognition*, pp. 1-6.

[81] Kim S. Y, Han H. G, Kim J. W, Lee S and Kim T. W (2017). A Hand Gesture Recognition Sensor Using Reflected Impulses. *IEEE Sensors Journal*, 17(10), 2975-2976.

[82] Kishore P. V. V, Prasad M. V. D, Kumar D. A and Sastry A. S. C. S (2016) Optical flow hand tracking and active contour hand shape features for continuous sign language recognition with artificial neural networks. *In: IEEE 6th International Conference on Advanced Computing (IACC)*, pp. 346-351.

[83] Kishore, P.V.V., Rao, G.A., Kumar, E.K., Kumar, M.T.K. and Kumar, D.A. (2018). Selfie sign language recognition with convolutional neural networks. *International Journal of Intelligent Systems and Applications*, *11*(10), p.63.

[84] Kitchenham B and ChartersS (2007) Guidelines for performing systematic literature reviews in software engineering. *Technical report, EBSE Technical Report EBSE*-2007-01, 2.3.

[85] Kosmidou V. E and Hadjileontiadis L. J (2008) Intrinsic mode entropy: An enhanced classification means for automated Greek sign language gesture recognition.

In: 30th IEEE Annual International Conference Engineering in Medicine and Biology Society, EMBS, pp. 5057-5060.

[86] Kumar A, Thankachan K and Dominic M.M (2016) Sign language recognition. *In: 3rd IEEE International Conference on Recent Advances in Information Technology (RAIT)*, pp. 422-428.

[87] Kumar D. A, Kishore P. V. V, Sastry A. S. C. S and Swamy P. R. G (2016) Selfie continuous sign language recognition using neural network. *In: IEEE Annual India Conference (INDICON)*, pp. 1-6.

[88] Kumar P, Gauba H, Roy P. P and Dogra D. P (2017a). A multimodal framework for sensor based sign language recognition, *Neurocomputing*, *259*, pp.21-38.

[89] Kumar P, Gauba H, Roy P. P and Dogra D. P (2017b) Coupled HMM-based multi-sensor data fusion for sign language recognition. *Pattern Recognition Letters*, 86, 1-8.

[90] Kumar P, Saini R, Behera S. K, Dogra D. P and Roy P. P (2017c) Real-time recognition of sign language gestures and air-writing using leap motion. *In: Fifteenth IEEE International Conference on Machine Vision Applications (MVA)*, pp. 157-160.

[91] Kumar P, Saini R, Roy P. P and Dogra D. P (2017d) A position and rotation invariant framework for sign language recognition (SLR) using Kinect. *Multimedia Tools and Applications*, 1-24.

[92] Lahoti, S., Kayal, S., Kumbhare, S., Suradkar, I. and Pawar, V. (2018) July. Android based american sign language recognition system with skin segmentation and SVM. In *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (pp. 1-6). IEEE.

[93] Lang S, Block M and Rojas R (2012) Sign language recognition using Kinect. *In Artificial intelligence and soft computing, Springer Berlin/Heidelberg* pp. 394-402.

[94] Latif, G., Mohammad, N., AlKhalaf, R., AlKhalaf, R., Alghazo, J. and Khan, M., (2020). An automatic Arabic sign language recognition system based on deep CNN: An assistive system for the deaf and hard of hearing. *International Journal of Computing and Digital Systems*, 9(4), pp.715-724.

[95] Lee, Carman KM, Kam KH Ng, Chun-Hsien Chen, Henry CW Lau, S. Y. Chung, and Tiffany Tsoi. (2021) American sign language recognition and training method with recurrent neural network. *Expert Systems with Applications* 167: 114403.

[96] Lee Y. H and Tsai C. Y (2009) Taiwan sign language (TSL) recognition based on 3D data and neural networks. *Expert systems with applications*, 36(2), 1123-1128.

[97] Lewis, M. Paul, Gary F. Simons, and Charles D. Fennig (eds.) 2014 *Ethnologue: Languages of the World, Seventeenth edition.* Dallas, Texas: SIL International. Online version: http://www.ethnologue.com/17/.

[98] Li, D., Rodriguez, C., Yu, X. and Li, H. (2020). Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 1459-1469).

[99] Lima, D.F., Neto, A.S.S., Santos, E.N., Araujo, T.M.U. and Rêgo, T.G.D. (2019), October. Using convolutional neural networks for fingerspelling sign recognition in brazilian sign language. *In Proceedings of the 25th Brazillian Symposium on Multimedia and the Web* pp. 109-115.

[100] Li Y, Chen X, Tian J, Zhang X, Wang K and Yang J (2010) Automatic recognition of sign language subwords based on portable accelerometer and EMG sensors. *In International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction, ACM*, p. 17.

[101] Luis-Pérez F. E, Trujillo-Romero F and Martínez-Velazco W (2011) Control of a service robot using the mexican sign language. *In: Mexican International Conference on Artificial Intelligence, Springer, Berlin, Heidelberg*, pp. 419-430.

[102] Luqman, H. and El-Alfy, E.S.M., 2021. Towards Hybrid Multimodal Manual and Non-Manual Arabic Sign Language Recognition: mArSL *Database and Pilot Study. Electronics*, 10(14), p.1739.

[103] Madani H and Nahvi M (2013) Isolated dynamic Persian sign language recognition based on camshift algorithm and radon transform. *In: First IEEE Iranian Conference on Pattern Recognition and Image Analysis (PRIA)*, pp. 1-5.

[104] Madushanka A. L. P, Senevirathne R. G. D. C, Wijesekara L. M. H, Arunatilake S. M. K. D and Sandaruwan K. D (2016) Framework for Sinhala Sign Language recognition and translation using a wearable armband. *In: Sixteenth IEEE International Conference on Advances in ICT for Emerging Regions (ICTer)*, pp. 49-57. IEEE.

[105] Majid M. B. A, Zain J. B. M and Hermawan A (2015) Recognition of Malaysian sign language using skeleton data with neural network. *In: IEEE International Conference on Science in Information Technology (ICSITech)*, pp. 231-236.

[106] Maraqa M and Abu-Zaiter R (2008) Recognition of Arabic Sign Language (ArSL) using recurrent neural networks. *In: First IEEE International Conference on the Applications of Digital Information and Web Technologies. ICADIWT*, pp. 478-481.

[107] Mariappan, H.M. and Gomathi, V. (2019), February. Real-time recognition of Indian sign language. In *2019 International Conference on Computational Intelligence in Data Science (ICCIDS)* pp. 1-6. IEEE.

[108] Maw, Jonathan, Kai Yuen Wong, and Patrick Gillespie. "Hand anatomy." *British Journal of Hospital Medicine* 77.

[109] “MediaPipe Hands”, Available: https://developers.google.com/mediapipe/solutions/vision/hand_landmarker, [Accessed: 2020].

[110] “MediaPipe Pose”, Available: https://developers.google.com/mediapipe/solutions/vision/pose_landmarker, [Accessed: 2020].

[111] Mehrotra K., Godbole A and Belhe S (2015) Indian Sign Language Recognition Using Kinect Sensor. In: International Conference on Image Analysis and Recognition, Springer, Cham, pp. 528-535.

[112] Mendes Junior, José Jair Alves, Melissa La Banca Freitas, Daniel Prado Campos, Felipe Adalberto Farinelli, Sergio Luiz Stevan, and Sérgio Francisco Pichorim. (2020) Analysis of Influence of Segmentation, Features, and Classification in sEMG Processing: A Case Study of Recognition of Brazilian Sign Language Alphabet." *Sensors* 20, no. 16 : 4359.

[113] *Mitchell R, Young T, Bachleda B, Karchmer M (2006)* How Many People Use ASL in the United States?: Why Estimates Need Updating *(PDF). Sign Language Studies. Gallaudet University Press.* ***6*** *(3).* ISSN 0302-1475. Retrieved November 27, 2012.

[114] Mittal, A., Kumar, P., Roy, P.P., Balasubramanian, R. and Chaudhuri, B.B. (2019). A modified LSTM model for continuous sign language recognition using leap motion. *IEEE Sensors Journal*, *19*(16), pp.7056-7063.

[115] Moghaddam M, Nahvi M and Pak R. H (2011) Static persian sign language recognition using kernel-based feature extraction. *In: 7th IEEE Iranian on Machine Vision and Image Processing (MVIP),* pp. 1-5.

[116] Mohandes M, Aliyu S and Deriche M (2014) Arabic sign language recognition using the leap motion controller. *In: 23rd IEEE International Symposium on Industrial Electronics (ISIE),* pp. 960-965.

[117] Mohandes M, Deriche M, Johar U and Ilyas S (2012) A signer-independent Arabic Sign Language recognition system using face detection, geometric features, and a Hidden Markov Model. *Computers & Electrical Engineering*, 38(2), 422-433.

[118] Mohandes M, Quadri S. I and Deriche M (2007) Arabic sign language recognition an image-based approach. *In: 21st IEEE International Conference on Advanced Information Networking and Applications Workshops, AINAW'07*, 1, pp. 272-276.

[119] Moryossef, Amit, Ioannis Tsochantaridis, Joe Dinn, Necati Cihan Camgoz, Richard Bowden, Tao Jiang, Annette Rios, Mathias Muller, and Sarah Ebling.(2021) Evaluating the Immediate Applicability of Pose Estimation for Sign Language Recognition. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3434-3440.

[120] Mukai N, Harada N and Chang Y (2017) Japanese Fingerspelling Recognition Based on Classification Tree and Machine Learning. *In: Nicograph International (NicoInt),* pp. 19-24.

[121] 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.

[122] Naglot D and Kulkarni M (2016) ANN based Indian Sign Language numerals recognition using the leap motion controller. *In: IEEE International Conference on Inventive Computation Technologies (ICICT)*, 2, pp. 1-6.

[123] Nakjai, P. and Katanyukul, T. (2019). Hand sign recognition for Thai finger spelling: An application of convolution neural network. *Journal of Signal Processing Systems*, 91(2), pp.131-146.

[124] Nel W, Ghaziasgar M and Connan J (2013) An integrated sign language recognition system. *In: Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference, ACM*, pp. 179-185.

[125] On-Device, Real Time Hand Tracking with MediaPipe, Available: https://ai.googleblog.com/2019/08/on-device-real-time-hand-tracking-with.html, Accessed: *August 19, 2019*.

[126] Oszust M and Wysocki M (2013) Polish sign language words recognition with kinect. *In: 6th IEEE International Conference on Human System Interaction (HSI)*, pp. 219-226.

[127] Oyedotun O. K and Khashman A (2017) Deep learning in vision-based static hand gesture recognition. *Neural Computing and Applications*, 28(12), 3941-3951.

[128] Oz C and Leu M C (2007) Linguistic properties based on American Sign Language isolated word recognition with artificial neural networks using a sensory glove and motion tracker. *Neurocomputing*, 70(16), 2891-2901.

[129] 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.

[130] Parcheta Z and Martínez-Hinarejos C. D (2017) Sign language gesture recognition using hmm. *In: Iberian Conference on Pattern Recognition and Image Analysis, Springer, Cham*, pp. 419-426.

[131] Pariwat T and Seresangtakul P (2017) Thai finger-spelling sign language recognition using global and local features with SVM. *In: 9th IEEE International Conference on Knowledge and Smart Technology (KST)*, pp. 116-120.

[132] Pariwat, T. and Seresangtakul, P. (2019). Thai Finger-Spelling Sign Language Recognition Employing PHOG and Local Features with KNN. *Int. J. Adv. Soft Comput. Appl*, 11, pp.94-107.

[133] Pariwat, T. and Seresangtakul, P. (2021). Multi-Stroke Thai Finger-Spelling Sign Language Recognition System with Deep Learning. *Symmetry*, 13, 262.

[134] Paulraj M. P, Yaacob S, Desa H, Hema C. R, Ridzuan W. M and Ab Majid W (2008) Extraction of head and hand gesture features for recognition of sign language. *In: IEEE International Conference on Electronic Design, ICED*, pp. 1-6.

[135] Prabhu (2018) Understanding of Convolutional Neural Network (CNN)— Deep Learning *https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148*. Accessed 4 March 2018.

[136] Pu J, Zhou W, Zhang J and Li H (2016) Sign language recognition based on trajectory modeling with hmms. *In: International Conference on Multimedia Modeling, Springer, Cham*, pp. 686-697.

[137] Pu, J., Zhou, W. and Li, H. (2019). Iterative alignment network for continuous sign language recognition. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* pp. 4165-4174.

[138] Quan Y and Jinye P (2008) Chinese Sign Language Recognition for a Vision-Based Multi-features Classifier. *In: IEEE International Symposium on Computer Science and Computational Technology, ISCSCT'08*, 2, pp. 194-197.

[139] Quan Yang Peng J and Yulong L (2009) Chinese sign language recognition based on gray-level co-occurrence matrix and other multi-features fusion. *In: 4th IEEE Conference on Industrial Electronics and Applications, ICIEA*, pp. 1569-1572.

[140] Ragab A, Ahmed M, and Chau S C (2013). Sign language recognition using Hilbert curve features. *In: International Conference Image Analysis and Recognition, Springer, Berlin, Heidelberg*, pp. 143-151.

[141] Raghuveera, T., Deepthi, R., Mangalashri, R. and Akshaya, R (2020) A depth-based Indian sign language recognition using microsoft kinect. *Sadhana*, 45(1), pp.1-13.

[142] Rahaman M. A, Jasim M, Ali M. H and Hasanuzzaman M (2014) Real-time computer vision-based Bengali Sign Language recognition. *In: 17th IEEE International Conference on Computer and Information Technology (ICCIT)*, pp. 192-197.

[143] Rastgoo, Razieh, Kourosh Kiani, and Sergio Escalera (2018) Multi-modal deep hand sign language recognition in still images using restricted Boltzmann machine. *Entropy* 20, no. 11: 809.

[144] Rastgoo, R., Kiani, K. and Escalera (2020a). Hand sign language recognition using multi-view hand skeleton. *Expert Systems with Applications*, 150, p.113336.

[145] Rastgoo, R., Kiani, K. and Escalera. (2020b). Video-based isolated hand sign language recognition using a deep cascaded model. *Multimedia Tools and Applications*, 79, pp.22965-22987.

[146] Rastgoo, R., Kiani, K. and Escalera 2021. Real-time isolated hand sign language recognition using deep networks and SVD. *Journal of Ambient Intelligence and Humanized Computing*, pp.1-21.

[147] Rao G. A and Kishore, P. V. V (2017). Selfie video based continuous Indian sign language recognition system. *Ain Shams Engineering Journal*, *9*(4), pp.1929-1939.

[148] Rao, G.A. and Kishore, P.V.V. (2018) Selfie sign language recognition with multiple features on adaboost multilabel multiclass classifier. *Journal of Engineering Science and Technology*, 13(8), pp.2352-2368.

[149] Rao G. A, Kishore P. V. V, Sastry A. S. C. S, Kumar D. A and Kumar E. K (2018) Selfie Continuous Sign Language Recognition with Neural Network Classifier. *In: Proceedings of 2nd International Conference on Micro-Electronics, Electromagnetics and Telecommunications, Springer, Singapore*, pp. 31-40.

[150] Reilly, Charles and Suvannus, Sathaporn (1999) Education of deaf people in the kingdom of Thailand.

[151] Rekha J, Bhattacharya J and Majumder S (2011). Shape, texture and local movement hand gesture features for indian sign language recognition. *In: 3rd IEEE*

International Conference on Trendz in Information Sciences and Computing (TISC), pp. 30-35.

[152] Rezende, T.M., Almeida, S.G.M. and Guimarães, F.G. (2021) Development and validation of a brazilian sign language database for human gesture recognition. *Neural Computing and Applications*, pp.1-19.

[153] Saengsri S, Niennattrakul V and Ratanamahatana C. A (2012) TFRS: Thai finger-spelling sign language recognition system. *In: Second International Conference on Digital Information and Communication Technology and it's Applications (DICTAP)*, pp. 457-462.

[154] Saha S, Lahiri R, Konar A and Nagar A. K (2016) A novel approach to American sign language recognition using MAdaline neural network. *In: IEEE Symposium Series on Computational Intelligence (SSCI)*, (pp. 1-6).

[155] Sajanraj, T.D. and Beena, M.V. (2018), April. Indian sign language numeral recognition using region of interest convolutional neural network. In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)* (pp. 636-640). IEEE.

[156] Sako S and Kitamura T (2013) Subunit modeling for Japanese sign language recognition based on phonetically depend multi-stream hidden markov models. *In: International Conference on Universal Access in Human-Computer Interaction, Springer, Berlin, Heidelberg*, pp. 548-555.

[157] Saleh, Y. and Issa, G. (2020) Arabic sign language recognition through deep neural networks fine-tuning, *International Association of Online Engineering. Retrieved August 24, 2022 from https://www.learntechlib.org/p/217934/.*

[158] Sarhan N. A, El-Sonbaty Y and Youssef S. M (2015) HMM-Based Arabic sign language recognition using Kinect. *In: Tenth IEEE International Conference on Digital Information Management (ICDIM)*, pp. 169-174.

[159] Sarkaleh A. K, Poorahangaryan F, Zanj B and Karami A (2009) A neural network based system for persian sign language recognition. *In: IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pp. 145-149.

[160] Savur C and Sahin F (2015) Real-time american sign language recognition system using surface EMG signal. *In: IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pp. 497-502.

[161] Savur C and Sahin F (2016) American Sign Language Recognition system by using surface EMG signal. *In: IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 002872-00287.

[162] Seymour M and Tšoeu M (2015) A mobile application for South African Sign Language (SASL) recognition. *In: AFRICON, IEEE*, pp. 1-5.

[163] Shamrat, F.J.M., Chakraborty, S., Billah, M.M., Kabir, M., Shadin, N.S. and Sanjana, S (2021) Bangla numerical sign language recognition using convolutional neural networks. *Indonesian Journal of Electrical Engineering and Computer Science* 23, no. 1, 405-413.

[164] Shanableh T and Assaleh K (2011) User-independent recognition of Arabic sign language for facilitating communication with the deaf community. *Digital Signal Processing*, 21(4), 535-542.

[165] Sharma A, Sharma N, Saxena Y, Singh A, Sadhya D (2021) Benchmarking deep neural network approaches for Indian Sign Language recognition. *Neural Computing and Applications*, 33(12), pp.6685-6696.

[166] Shenoy, K., Dastane, T., Rao, V. and Vyavaharkar, D. (2018), July. Real-time Indian sign language (ISL) recognition. In *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (pp. 1-9). IEEE.

[167] Sign Language Ethnologue, Available: *https://www.ethnologue.com/browse/names/*.

[168] Simonyan, Karen, and Andrew Zisserman (2014) Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[169] Simos M and Nikolaidis N (2016) Greek sign language alphabet recognition using the leap motion device. *In: Proceedings of the 9th Hellenic Conference on Artificial Intelligence, ACM* pp. 34.

[170] Sincan, Ozge Mercanoglu, Julio Junior, C. S. Jacques, Sergio Escalera, and Hacer Yalim Keles (2021) Chalearn LAP large scale signer independent isolated sign language recognition challenge: Design, results and future research. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3472-3481.

[171] Sripairojthikoon, N. and Harnsomburana, J. (2019), July. Thai Sign Language Recognition Using 3D Convolutional Neural Networks. *In Proceedings of the 2019 7th International Conference on Computer and Communications Management* (pp. 186-189).

[172] Srivastava, Sharvani, Amisha Gangwar, Richa Mishra, and Sudhakar Singh. (2021) "Sign language recognition system using TensorFlow object detection API." In *International conference on advanced network technologies and intelligent computing*, pp. 634-646.

[173] Sruthi, C. J., and A. Lijiya (2019) Signet: A deep learning based indian sign language recognition system. In *2019 International conference on communication and signal processing (ICCSP)*, pp. 0596-0600. IEEE.

[174] Suri, K. and Gupta, R. (2019) Continuous sign language recognition from wearable IMUs using deep capsule networks and game theory. *Computers & Electrical Engineering*, 78, pp.493-503.

[175] Sun C, Zhang T, Bao B. K and Xu C (2013a) Latent support vector machine for sign language recognition with Kinect. *In: 20th IEEE International Conference on Image Processing (ICIP)*, pp. 4190-4194.

[176] Sun C, Zhang T, Bao B. K, Xu C, and Mei T (2013b). Discriminative exemplar coding for sign language recognition with Kinect. *IEEE Transactions on Cybernetics*, 43(5), 1418-1428.

[177] Sun C, Zhang T and Xu C (2015) Latent support vector machine modeling for sign language recognition with Kinect. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(2), 20.

[178] Tangsuksant W, Adhan S, and Pintavirooj C (2014). American Sign Language recognition by using 3D geometric invariant feature and ANN classification. *In: 7th International Conference on Biomedical Engineering (BMEiCON)*, pp. 1-5.

[179] Taskiran, M., Killioglu, M. and Kahraman, N. (2018), July. A real-time system for recognition of American sign language by using deep learning. In *2018 41st International Conference on Telecommunications and Signal Processing (TSP)* (pp. 1-5). IEEE.

[180] Thangali A., and Sclaroff S. (2009) An alignment based similarity measure for hand detection in cluttered sign language video, in *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Florida, USA, pp. 89-96, 2009.

[181] Thang P. Q, Thuy, N. T and Lam H. T (2017a) The SVM, SimpSVM and RVM on sign language recognition problem. *In: Seventh IEEE International Conference on Information Science and Technology (ICIST)*, pp. 398-403.

[182] Thang P. Q, Dung N. D and Thuy N. T (2017b) A comparison of simpsvm and rvm for sign language recognition. *In: Proceedings of the 2017 International Conference on Machine Learning and Soft Computing, ACM*, pp. 98-104.

[183] Theodorakis S, Katsamanis A and Maragos P (2009) Product-HMMs for automatic sign language recognition. *In: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pp. 1601-1604.

[184] Tripathi K, Baranwal N and Nand G. C (2015) Continuous dynamic Indian Sign Language gesture recognition with invariant backgrounds. *In: IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2211-2216.

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

[186] Uddin M A and Chowdhury S. A (2016). Hand sign language recognition for Bangla alphabet using Support Vector Machine. *In: IEEE International Conference on Innovations in Science, Engineering and Technology (ICISET)*, pp. 1-4.

[187] “Understanding GoogLeNet Model – CNN Architecture”, Available: *https://www.geeksforgeeks.org/understanding-googlenet-model-cnn-architecture/*, [Accessed: 18, Nov, 2021].

[188] Usachokcharoen P, Washizawa Y and Pasupa K (2015) Sign language recognition with microsoft Kinect's depth and colour sensors. *In: IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pp. 186-190.

[189] Vanjikumaran S and Balachandran G (2011) An automated vision based recognition system for Sri Lankan Tamil sign language finger spelling. *In: IEEE International Conference on Advances in ICT for Emerging Regions (ICTer)*, pp. 39-44.

[190] Wadhawan, A. and Kumar, P. (2020) Deep learning-based sign language recognition system for static signs. *Neural Computing and Applications*, 32(12), pp.7957-7968.

[191] Wang X, Jiang F and Yao H (2008) DTW/ISODATA algorithm and Multilayer architecture in Sign Language Recognition with large vocabulary. *In: IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIHMSP*, pp. 1399-1402.

[192] Wen, Feng, Zixuan Zhang, Tianyiyi He, and Chengkuo Lee (2021) AI enabled sign language recognition and VR space bidirectional communication using triboelectric smart glove." *Nature communications* 12, no. 1 :1-13.

[193] Wise S., Gardner W., Sabelman E., Valainis E., Wong Y., Glassand K., Drace J., and Rosen J (1990) Evaluation of a fiber optic glove for semi-automated goniometric measurements. *Journal of rehabilitation research and development*, 27(4), pp. 411–424.

[194] Wu J, Tian Z, Sun L, Estevez L and Jafari R (2015) Real-time American sign language recognition using wrist-worn motion and surface EMG sensors. *In: IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pp. 1-6.

[195] Xiao, Q., Qin, M., Guo, P. and Zhao, Y. (2019) Multimodal fusion based on LSTM and a couple conditional hidden Markov model for Chinese sign language recognition. *IEEE Access*, 7, pp.112258-112268.

[196] Xiao, Q., Qin, M. and Yin, Y. (2020) Skeleton-based Chinese sign language recognition and generation for bidirectional communication between deaf and hearing people. *Neural networks*, 125, pp.41-55.

[197] Yang Q (2010) Chinese sign language recognition based on video sequence appearance modeling. *In: 5th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 1537-1542.

[198] Yang S and Zhu Q. (2017, May). Video-based Chinese sign language recognition using convolutional neural network. *In IEEE 9th International Conference on Communication Software and Networks (ICCSN),* pp 929-934.

[199] Yang W, Tao J and Ye Z (2016) Continuous sign language recognition using level building based on fast hidden Markov model. *Pattern Recognition Letters*, 78, 28-35.

[200] Yang W, Tao J, Xi C and Ye Z (2015) Sign language recognition system based on weighted hidden Markov model. *In: 8th IEEE International Symposium on Computational Intelligence and Design (ISCID)*, 2, pp. 449-452.

[201] Yasir F, Prasad P. C, Alsadoon A and Elchouemi A (2015) Sift based approach on bangla sign language recognition. *In: IEEE 8th International Workshop on Computational Intelligence and Applications (IWCIA),* pp. 35-39.

[202] Yu S. H, Huang C. L, Hsu S. C, Lin H. W and Wang H. W (2011) Vision-based continuous sign language recognition using product HMM. *In: IEEE First Asian Conference on Pattern Recognition (ACPR)*, pp. 510-514.

[203] Zadghorban, M. and Nahvi, M., 2018. An algorithm on sign words extraction and recognition of continuous Persian sign language based on motion and shape features of hands. *Pattern Analysis and Applications*, 21(2), pp.323-335.

[204] Zakariya, A.M. and Jindal, R. (2019) July. Arabic Sign Language Recognition System on Smartphone. *In 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (pp. 1-5). IEEE.

[205] Zamani M, and Kanan H. R (2014). Saliency based alphabet and numbers of American sign language recognition using linear feature extraction. *In: 4th IEEE*

International eConference on Computer and Knowledge Engineering (ICCKE), pp. 398-403.

[206] Zare A. A and Zahiri S. H (2016) Recognition of a real-time signer-independent static Farsi sign language based on Fourier coefficients amplitude. *International Journal of Machine Learning and Cybernetics*, 1-15.

[207] Zhang J, Zhou W and Li H (2014) A threshold-based hmm-dtw approach for continuous sign language recognition. *In: Proceedings of International Conference on Internet Multimedia Computing and Service, ACM*, p. 237.

[208] Zhang J, Zhou W and Li H (2015) A new system for chinese sign language recognition. *In: 2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)*, pp. 534-538.

[209] Zhang J, Zhou W, Xie C, Pu J and Li H (2016) Chinese sign language recognition with adaptive hmm. *In: IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1-6.

[210] Zhao, K., Zhang, K., Zhai, Y., Wang, D. and Su, J. (2021) Real-time sign language recognition based on video stream. *International Journal of Systems, Control and Communications*, 12(2), pp.158-174.

APPENDIX A

Table 1: Sample signs showing performance variation

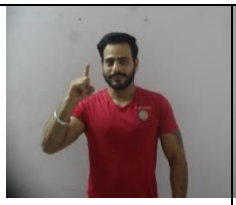
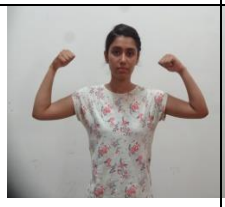
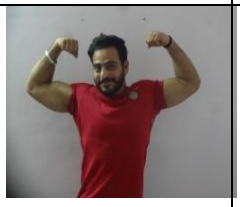
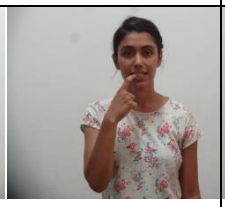
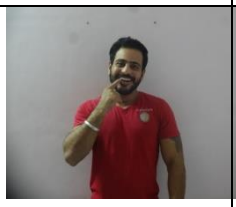
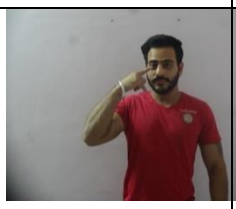
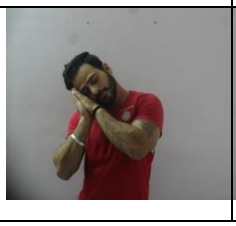
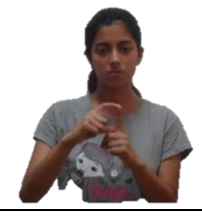
Name of Sign	Different Sign Samples				
One					
Strong					
Water					
Eye					
Sleep					

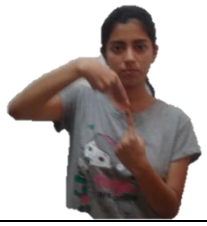
Table 2: Variation of 'Aeroplane' sign in different video frames of the same participant

Video Frames



Table 3: List of Static Signs

Sample Sign Image		
0	**1**	**2**
		
3	**4**	**5**
		
6	**7**	**8**
		
9	**10**	**A**
		
B	**C**	**D**
		
E	**F**	**G**

		
I	**K**	**L**
		
M	**N**	**O**
		
P	**Q**	**R**
		
S	**T**	**U**
		
V	**W**	**X**
		
Z	**Add**	**Afraid**

		
Bent	**Between**	**Blind**
		
Bottle	**Bowl**	**Brain**
		
Bud	**Chest**	**Claw**
		
Coolie	**Cough**	**Cow**
		
Devil	**Doctor**	**East**
		
Elbow	**Evening**	**Eye**

		
Faith	**Fat**	**Feel**
		
Fever	**Few**	**Fist**
		
Food	**Good**	**Gun**
		
Hair	**Hand**	**Head**
		
Hear	**Jain**	**King**
		
Leprosy	**Love**	**Me**

		
Nose	**Nurse**	**Oath**
		
Open	**Owl**	**Police**
		
Pray	**Promise**	**Shirt**
		
Shoulder	**Sick**	**Skin**
		
Sleep	**Soldier**	**Stand**
		
Strong	**Sunday**	**Telephone**

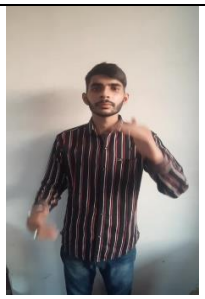
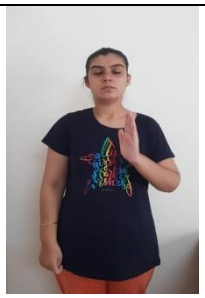
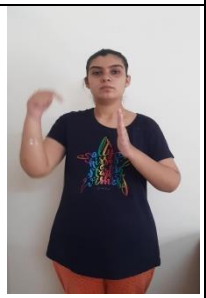
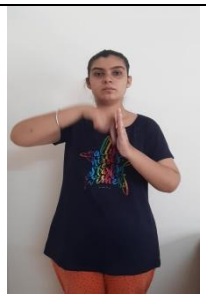
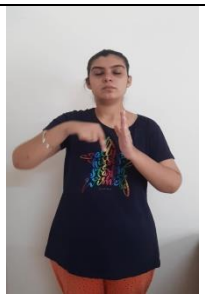
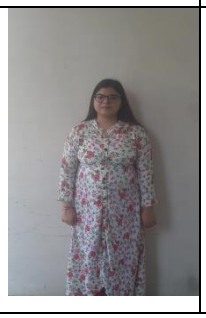
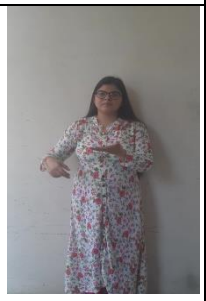
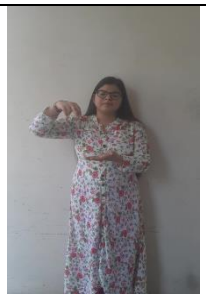
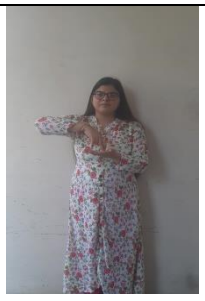
		
Thorn	**Thumbs Up**	**Tongue**
		
Trouble	**Water**	**Wedding**
		
West	**White**	**Word**
		
You		
		

Table 4: Categories of Signs

S No.	Single-handed Signs	Double Handed Signs	Signs with Facial Expression	Single-Handed signs with face	Double Handed signs with face
1	Zero	Ten	Cough	Blind	Oath
2	One	A	Devil	Eye	Sleep
3	Two	B	Owl	Fever	
4	Three	D	Tongue	Hair	
5	Four	E	Water	Head	
6	Five	F	White	Hear	
7	Six	G		Jain	
8	Seven	I		King	
9	Eight	K		Nose	
10	Nine	M		Nurse	
11	C	N		Sick	
12	L	P		Sunday	
13	O	Q		Telephone	
14	Bent	R		Trouble	
15	Bowl	S			
16	Brain	T			
17	Chest	U			
18	Claw	V			
19	East	W			
20	Faith	X			
21	Feel	Z			
22	Few	Add			
23	Fist	Afraid			
24	Food	Between			
25	Good	Bottle			
26	Me	Bud			
27	Open	Coolie			
28	Police	Cow			
29	Shirt	Doctor			
30	Thumbs Up	Elbow			
31	West	Evening			

32	Word	Fat			
33	You	Gun			
34		Hand			
35		Leprosy			
36		Love			
37		Pray			
38		Promise			
39		Shoulder			
40		Skin			
41		Soldier			
42		Stand			
43		Strong			
44		Thorn			
45		Wedding			
Total Signs	**33**	**45**	**6**	**14**	**2**
	100 Signs				

Table 5: Video frames of different signs

Name of Sign	Video Frames				
Above					
Decrease					
Depth					
Centre					

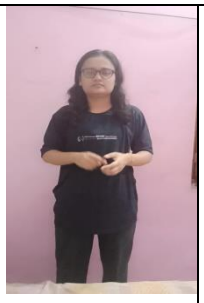
Education					
Airplane					

Table 6: List of Dynamic Signs

S No.	Name of Sign	S No.	Name of Sign
1	Above	26	Clip
2	Absent	27	Close
3	Accept	28	Cloth
4	Accompany	29	Community
5	Airplane	30	Confidence
6	Afternoon	31	Confirm
7	Apply	32	Copy
8	Award	33	Daily
9	Back	34	Decrease
10	Ball	35	Depth
11	Bank	36	Distribute
12	Below	37	Drive
13	Big	38	Drop
14	Calm	39	Earth
15	Captain	40	Edit
16	Caption	41	Education
17	Carpet	42	Electricity
18	Category	43	Email
19	Center	44	Emotion
20	Certificate	45	End
21	Change	46	File
22	Charity	47	Make
23	Cheque	48	Paint
24	Choose	49	Range
25	Circle	50	Respect