Sign Language Recognition System

Final Year Project   
2018-2022



*By*

**Muhammad Umar 181036**

**Tasmiyah Ali 181110**

Supervisor:

**Mr. Saud Arshad**

Department of Computer Science

Faculty of Computing & Artificial Intelligence (FCAI)

Air University, Islamabad

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type (Nature of project) | | | [ ✓ ] Development [ ✓ ] Research | | |
| Area of specialization | | | [ ✓ ] WebApp [ ] Mobile App  [ ] AI based [ ] Embedded System | | |
| FYP ID | | |  | | |
| **Project Group Members** | | | | | |
| Sr.# | Reg. # | Student Name | | Email ID | \*Signature |
| (i) | 181036 | Muhammad Umar | | 181036@students.au.edu.pk |  |
| (ii) | 181110 | Tasmiyah Ali | | 181110@students.au.edu.pk |  |
| (iii) |  |  | |  |  |

\* The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to the work of others

# Plagiarism Certificate

This is to certify that, I **Muhammad Umar** S/D of **Raja Yasir Ali**, group leader of FYP under registration no **181036** at Computer Sciences Department, Air University. I declare that my FYP report is checked by my supervisor.

Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name of Group Leader: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature:\_\_\_\_\_\_\_\_

Name of Supervisor: Mr. Saud Arshad

Designation: Lecturer

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

HoD: Dr. Awais Ahmad

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Sign Language Recognition System

**Change Record**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Author(s)** | **Version** | **Date** | **Notes** | **Supervisor’s Signature** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**APPROVAL**

|  |  |
| --- | --- |
| **Project Supervisor** | |
| Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  | |
| Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Signature:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |

|  |  |
| --- | --- |
| **Chair Department** | |
|  |  |
| Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Signature:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

# Dedication

*This work is dedicated to our parents.*

*For their endless love, sup,port and encouragement*

# Acknowledgments

We would like to express our gratitude to our supervisor, Sir Saud Arshad whose expertise and understanding, added considerably to our experience. We would also like to thank our families for their support and our friends, who also helped us in this project.

Abstract

In the realm of multimodal communication, sign language is and continues to be, one of the most understudied areas. In line with recent advances in the field of deep learning, there are far-reaching implications and applications that neural networks can have for sign language interpretation. Sign language is not enough for communication between people with hearing ability or people with speech disabilities. The gestures made by the people with a disability gets mixed or disordered for someone who has never learned this language. In this project, we are using a convolutional neural networks to classify static images of gestures. While the translation process between signs and a spoken or written language is formally called ‘interpretation,’ the function that interpreting plays is the same as that of translation for a spoken language. The work presented in this project is an extension of examining the difficulties in the classification of words in Sign Language. The user must be able to capture images of hand gestures using a web camera, the system must predict and show the name of the captured image. The captured image undergoes a series of preprocessing steps which include various computer vision techniques, Convolutional Neural Networks (CNN) are used for feature extraction and to train our models and identify the pictures. 2D CNN, Inceptionresnetv2, and VGG modes were trained and a web-based Gui was implemented using flask to get images from live camera feed and display the predictions.

# Table of Contents

[Final Year Project 2018-2022 i](#_Toc103183942)

[BS in Computer Science i](#_Toc103183943)

[Plagiarism Certificate iii](#_Toc103183944)

[Dedication vi](#_Toc103183945)

[Acknowledgements vii](#_Toc103183946)

[Table of Contents ix](#_Toc103183947)

[List of Figures xii](#_Toc103183948)

[List of Tables xiii](#_Toc103183949)

[List of Abbreviations xiv](#_Toc103183950)

[Chapter 1 1](#_Toc103183951)

[Introduction & Background 1](#_Toc103183952)

[1.1. Background 2](#_Toc103183953)

[1.2. Motivations and Challenges 2](#_Toc103183954)

[1.3. Goals and Objectives 3](#_Toc103183955)

[1.4. Literature Review/Existing Solutions 3](#_Toc103183956)

[1.5. Gap Analysis 3](#_Toc103183957)

[1.6. Proposed Solution 3](#_Toc103183958)

[1.7. Project Plan 3](#_Toc103183959)

[1.7.1. Work Breakdown Structure 4](#_Toc103183960)

[1.7.2. Roles & Responsibility Matrix 4](#_Toc103183961)

[1.7.3. Gantt Chart 5](#_Toc103183962)

[Chapter 2 6](#_Toc103183963)

[Software Requirement Specifications 6](#_Toc103183964)

[2.1 Introduction 7](#_Toc103183966)

[2.1.1 Purpose 7](#_Toc103183967)

[2.1.2 Document Conventions 7](#_Toc103183968)

[2.1.3 Intended Audience and Reading Suggestions 7](#_Toc103183969)

[2.1.4 Product Scope 8](#_Toc103183970)

[2.2 Overall Description 8](#_Toc103183971)

[2.2.1 Product Perspective 8](#_Toc103183972)

[2.2.2 User Classes and Characteristics 8](#_Toc103183973)

[2.2.3 Operating Environment 9](#_Toc103183974)

[2.2.4 Design and Implementation Constraints 9](#_Toc103183975)

[2.2.5 Assumptions and Dependencies 9](#_Toc103183976)

[2.3 External Interface Requirements 9](#_Toc103183977)

[2.3.1 User Interfaces 9](#_Toc103183978)

[2.3.2 Hardware Interfaces 9](#_Toc103183979)

[2.3.3 Software Interfaces 10](#_Toc103183980)

[2.3.4 Communications Interfaces 10](#_Toc103183981)

[2.4 System Features 10](#_Toc103183982)

[2.4.1 System Feature 1 10](#_Toc103183983)

[2.4.1.1 Description and Priority 10](#_Toc103183984)

[2.4.1.2 Stimulus/Response Sequences 10](#_Toc103183985)

[2.4.1.3 Functional Requirements 11](#_Toc103183986)

[2.4.2 System Feature 2 11](#_Toc103183987)

[2.4.2.1 Description and Priority 11](#_Toc103183988)

[2.4.2.2 Stimulus/Response Sequences 11](#_Toc103183989)

[2.4.2.3 Functional Requirements 11](#_Toc103183990)

[2.4.3 System Feature 3 (and so on) 12](#_Toc103183991)

[2.5 Nonfunctional Requirements 12](#_Toc103183992)

[2.5.1 Performance Requirements 12](#_Toc103183993)

[2.5.2 Safety Requirements 12](#_Toc103183994)

[2.5.3 Security Requirements 12](#_Toc103183995)

[2.5.4 Usability Requirements 12](#_Toc103183996)

[2.5.5 Reliability Requirements 13](#_Toc103183997)

[2.5.6 Maintainability/Supportability Requirements 13](#_Toc103183998)

[2.5.7 Portability Requirements 13](#_Toc103183999)

[2.5.8 Efficiency Requirements 13](#_Toc103184000)

[2.6 Domain Requirements 13](#_Toc103184001)

[Chapter 3 14](#_Toc103184002)

[Literature Review 14](#_Toc103184003)

[3.1 DiffereCamera-Basedon Methods 16](#_Toc103184006)

[3.1.1 Camera Based Acquisition 16](#_Toc103184007)

[3.1.2 Glove Based Acquisition 17](#_Toc103184008)

[3.1.3 Kinect Based Acquisition 18](#_Toc103184009)

[Chapter 4 20](#_Toc103184012)

[Use Case Analysis 20](#_Toc103184013)

[4.1. Use Case Model 21](#_Toc103184016)

[4.2. Use Cases Description 22](#_Toc103184017)

[Chapter 5 27](#_Toc103184018)

[Proposed Approach 27](#_Toc103184019)

[5.1 Internal Storage 28](#_Toc103184028)

[5.2 Data Augmentation and Pre-Processing 28](#_Toc103184029)

[5.3 Feature Extraction 29](#_Toc103184030)

[5.4 Training Model 29](#_Toc103184031)

[5.5 Results and Discussion 29](#_Toc103184032)

[Chapter 6 31](#_Toc103184033)

[Implementation & Results 31](#_Toc103184034)

[6.1 Different components of your proposed approach 32](#_Toc103184035)

[6.2 Implementation of proposed approach 32](#_Toc103184036)

[6.3 Results 32](#_Toc103184037)

[6.4 Tools and Techniques 33](#_Toc103184045)

[Chapter 7 34](#_Toc103184048)

[Business Plan 34](#_Toc103184049)

[**Chapter 7:** Business Plan 35](#_Toc103184050)

[7.1 Business Description 35](#_Toc103184054)

[7.2 Market Analysis & Strategy 35](#_Toc103184055)

[7.3 Competitive Analysis 35](#_Toc103184056)

[7.4 Products/Services Description 35](#_Toc103184057)

[7.5 SWOT Analysis 36](#_Toc103184058)

[7.5.1 Strengths 36](#_Toc103184059)

[7.5.2 Weaknesses 36](#_Toc103184060)

[7.5.3 Opportunities 36](#_Toc103184061)

[7.5.4 Threats 36](#_Toc103184062)

[Chapter 8 37](#_Toc103184063)

[Conclusion & Future Work 37](#_Toc103184064)

[**Chapter 8:** Conclusion & Future Work 38](#_Toc103184065)

[8.1 Achievements and Improvements 38](#_Toc103184067)

[8.2 Critical Review 38](#_Toc103184068)

[8.3 Future Enhancements/Recommendations 39](#_Toc103184069)

[Appendices 42](#_Toc103184077)

[Appendix A: Information / Promotional Material 43](#_Toc103184078)

[References 44](#_Toc103184079)

# List of Figures

1.4 Work break down Structure

4.1 Use Case for Sign Language Recognition system

5.1 Proposed Approach

5.5 Demo Output

# List of Tables

Table 4.1: Start webcam

Table 4.2: Capture image

Table 4.3: Capture Gesture

Table 4.3: Extract Features

Table 4.3: Recognize Gesture

Table 4.3: Display Result

# List of Abbreviations

SLR Sign Language Recognition

CNN Convolutional Neural Networks

ROI Region Of Interest

# 

# Chapter 1

# Introduction & Background

Chapter 1: Introduction

This chapter covers the key aspects of Sign Language Recognition (SLR), starting with a brief introduction, the goals and objectives of SLR to the motivations and challenges faced.

## Background

Sign language recognition and translation is a research area with high potential impact. Sign Language Recognition (SLR) system is the technique to identify a sequence of developed signs and translated them into text or speech with proper meaning. SLR is a combination of research fields including pattern recognition, natural language processing, computer vision and linguistics.

Building successful sign language processing systems requires an understanding of deaf culture to create systems that align with user needs and desires, and of sign languages to build systems that account for their complex linguistic aspects. SLR systems can be used as a connection between an individual and computer systems. The sign language recognition can be done either in two ways. One is with physical devices like gloves, Kinect etc. and the other is computer vision based.

We intend to introduce an SLR system that would be easy to use and simpler version using computer vision techniques and that will not use any external or expensive hardware.

## Motivations and Challenges

This project is a first step towards building a possible sign language translator, which can take communications in sign language and translate them into written language. Such a translator would greatly lower the barrier for many deaf and mute individuals to be able to better communicate with others in day to day interactions.  
This project is further motivated by the isolation that is felt within the deaf community. Loneliness and depression exists in higher rates among the deaf population, especially when they are immersed in a hearing world. Large barriers that profoundly affect life quality stem from the communication disconnect between the deaf and the hearing. Some examples are information deprivation, limitation of social connections, and difficulty integrating in society.

## Goals and Objectives

The aim of this project is to use the corresponding gestures to recognize words in Sign Language and to contribute to the field of automatic sign language recognition.

**Objectives**

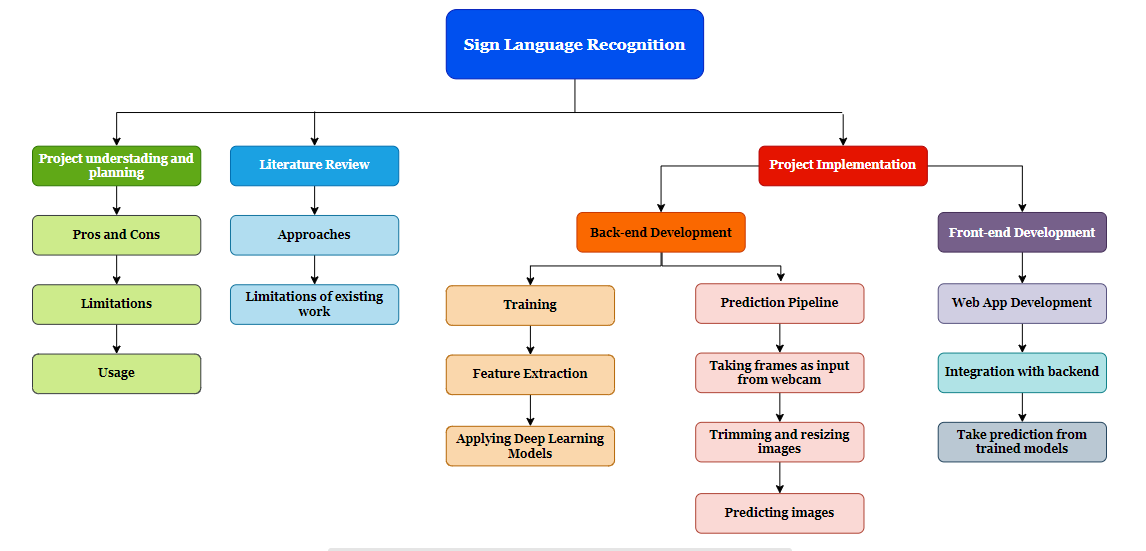
To remove the invisible communication barrier while communicating with mute/deaf people. The system would be able to recognize a selected subset of a sign language. The system would be Open Source and be easily accessible to a wide range of potential users. It would be cost-efficient and easy to set up with any expensive hardware.

## Project Plan

* Literature Review
* Design specification
* Development
* Training
* Documentation

## Work Breakdown Structure

* Study of the previously done research on the Sign Language Recognition system.
* Selection of relevant models.
* Training of model.
* Implementation of the model.
* Analysis of model.
* Development of front end.
* Documentation.
* Presentation.



## Roles & Responsibility Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| 1. **Tasks** | **Sir Saud Arshad** | **Muhammad Umar** | **Tasmiyah Ali** |
| Literature Review | C/A | R/A | R/A |
| Design Specification | C/A | R | R |
| Development | I | R/A | A |
| Training | C/I | R | R |
| Documentation | I | A | R |

**R – Responsible**

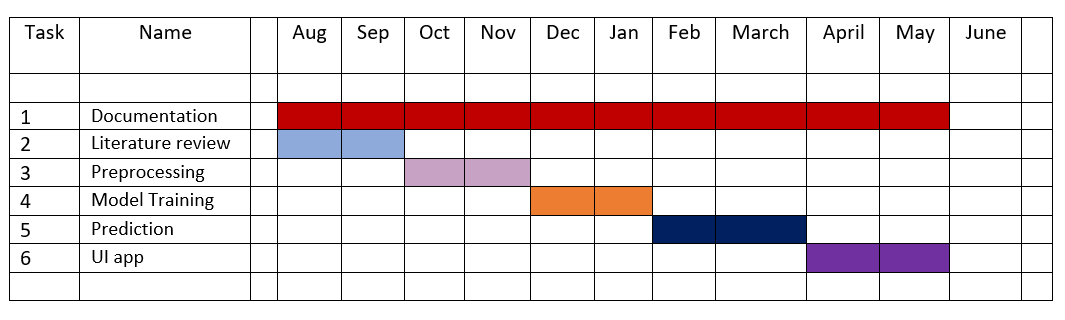
Who will be responsible for getting the task done**A – Accountable**

Who is accountable for a particular task and hence, the signatory authority**C – Conform**

The SMEs/expert who can be consulted regarding the issues**I – Inform**

Who are the people who need the project progress report

## Gantt Chart



# Chapter 2

# Software Requirement Specifications

**Chapter 2:** Software Requirement Specifications



## Introduction

This document is a software requirement specification of the Sign Language Recognition System. Through this document, we are going to provide all specifications and functionalities of the project. This document will mention the functionality, that is what the resulting application is supposed to do, external interfaces which interacts the users, performance, attributes, that is if the application is portable, or maintainable, and design constraints imposed on the implementation such as implementation language, input specifications and output expectations.

## Purpose

The purpose of this software is to make a very convenient system that will help break down the communication barrier between deaf community and normal hearing people, to give our society a system that will be for their (deaf and mute community) ease to be in touch with the majority.

## Document Conventions

## The SRS document uses few different font sizes for clear distinction. For example, 14pt for the heading and 12pt for the paragraph text. The main headings are numbered with whole numbers like 1. Purpose 2. Overall description. The subheadings are numbered with decimals like 2.1.1 purpose, 2.1.2 document conventions etc. All the requirements mentioned in the document have their own priority

## Intended Audience and Reading Suggestions

The SRS is intended for Students and Teachers and this product is useful in all public places, academic institutes, organizations, etc.

* Deaf community
* Organizations and all other people who have to deal with deaf community.

This SRS document contains product scope, overall description, external interface requirements, Non-functional Requirements and Domain Requirements. Contents should be read in the sequence it is mentioned. The SRS has been organized approximately in order of increasing specificity.

## Product Scope

The project which is going to be presented in this document is called Sign Language Recognition. This application is planned to be used by speechless people in order to ease their life, and also government offices that should serve all of its citizens equally, private companies that want to reach and serve speechless people as well, corporations and foundations which aims to help speech-disordered people.

In this project, it is aimed to recognize Sign Language and translate it into text only. Obviously, our program will have some limitations like it will recognize limited number of gestures. For the beginning, the program will recognize only 20-25 gestures pre-defined. Yet, it will be possible to define more gestures once it is proved that the program works well enough.

It is a web-based application that will take inputs as images from the live camera feed and as the user will click the button it will show the meaning of the sign performed in-front of webcam.

## Overall Description

## Product Perspective

Sign Language will serve speech-impaired people and not have difficulty in communication by providing their gestures converting to the text for people who do not know Sign Language. This project is a follow-up of Sign Language recognition systems using computer vision techniques.

Many different ways and Machine learning techniques have been used in this field and currently we are working with static images to predict signs using a cnn architectures.

## User Classes and Characteristics

Most of the users of this product will be the deaf community as its sole purpose is to make a system for them to communicate more conveniently. Other user classes will be organizations, institutions, and public places where there is a need for translators however people who are differently able to are everywhere around the globe so these systems should be available everywhere for their convenience.

## Operating Environment

This system will be able to operate on even an average pc and will not require any special hardware as it was also the aim of our project to create a very cost-efficient system. This system is developed on windows using python 3.8 and html css. Some software requirements are mentioned further.

## Design and Implementation Constraints

The dataset should be collected from different signers in different lighting conditions and from different angles to make the system more accurate.

Users can perform signs while sitting as well as standing as this system only detects hands and does not depend on body posture or position, only hands should be static and in the right position.

## Assumptions and Dependencies

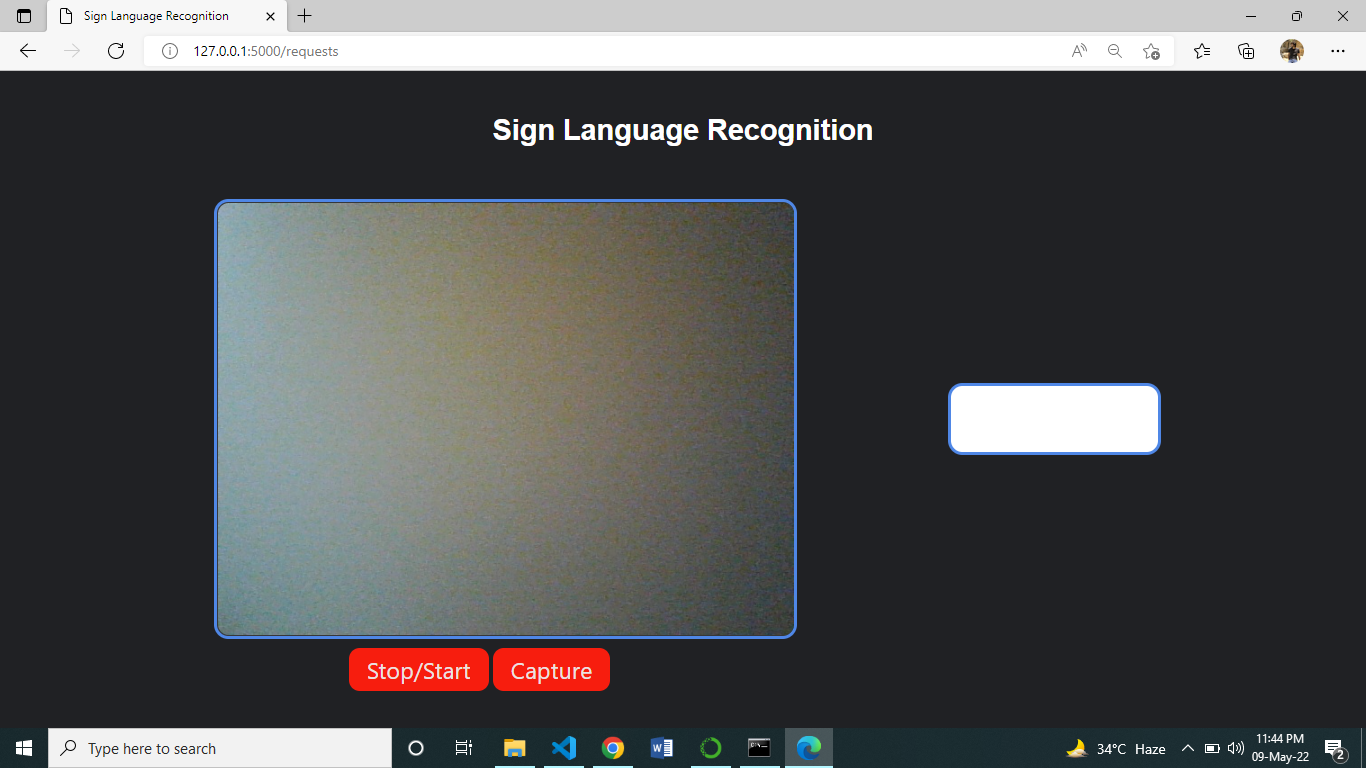
* User must know how to perform sign in front of camera
* System must respond in reasonable time
* System will not perform in dark conditions

## External Interface Requirements

## 

## User Interfaces

There will be a single screen user interface with two buttons to start/stop the camera and for capturing the frame and a textbox will be used to display output. On the screen there will be a window showing full frame of the camera feed.



## 

## Hardware Interfaces

The system will require only a webcam for capturing images, no other hardware devices are required. Camera feed will be the input of the system and the system will predict the captured image and show the output on screen.

## Software Interfaces

This system is implemented using python3 on vs code. OpenCV is used for image processing, HTML/CSS for GUI design, Tensorflow-keras for model training, and flask for front-end/back-end integration. Frames from the webcam are received the predicted text is the output displayed.

## Communications Interfaces

Flask which is a micro web framework written in Python, is used for integration of front end.

It uses a templating language called Jinja2 which is used to create HTML, XML or other markup formats that are returned to the user via an HTTP request.

## System Features

The main system feature of this project is to predict the meaning of the signs performed in front of the camera and display them on the screen. From the camera feed, static images of hand will be extracted and will be classified.

## Hand Gesture Recognition

## Description and Priority

It is the only and main feature of the system in which our system is predicting meaning of the sign performed in front of the camera and displaying the output as translation of the sign.

## Stimulus/Response Sequences

Signer will perform the sign in front of the camera which will be captured and passed as the input to the model which will predict the captured image.

## Functional Requirements

REQ-SF1-1: Clear and stable camera

REQ-SF1-2: Static hand gesture

REQ-SF1-3: lighting conditions

## Nonfunctional Requirements

## Performance Requirements

The system’s performance would be very efficient i.e. rapid response to convey proper message to the user when a sign is performed.

## Safety Requirements

If the interpretation of the hand gesture is not correct due to any reason i.e. unstable or blur camera, bad lightning conditions or any type of noise in the frame it can cause a misunderstandings especially in learning processes.

## Security Requirements

* 1. User can only perform signs and cannot add or remove gestures.
  2. User has access constraints.

## Usability Requirements

* Interface will be easy to use.
* No useless buttons or widgets.
* System will be accurate and minimum error rate

## Reliability Requirements

* It will be able to recognize

## Maintainability/Supportability Requirements

* System should restart if it is unable to access camera.

## Portability Requirements

* System will be portable enough to run on any computer

## Efficiency Requirements

* System will be cost-efficient as no external hardware equipment is required such as gloves or special cameras.

## Domain Requirements

This system can be used in every region as differently abled people are everywhere in the world and it is a developed as a generalized system.

# Chapter 3

# Literature Review

**Chapter 3:** Literature Review

Sign Language Recognition is a software that will be able to classify the various hand gestures in sign language and display their meaning as text. This is  
also helpful in communication between speaking community and the person who neither speaks nor hears. We have studied several research papers and other publications to search for different approaches and techniques that can be used to detect sign language.

Lean Karlo[1] et. al. developed the system as a source for people new in Sign Language that involves detection of hand signs based on modeling of skin ccolor The Convolutional Neural Network (CNN) provides the pictures into the model for image classification. For image training Keras was used. The system acquired 93.67% of average testing precision, of which ASL alphabet recognition was 90.04%, for static word recognition 97.52% and 93.44% for number recognition, when provided with a uniform background and proper lighting condition, surpassing results of other related studies. The method was used for quick computation and is done in real time.

The paper by Jie Huang et. al.[2] focuses on decomposing continuous Sign Language Recognition (SLR) into single isolated worlds and solving these problems using temporal segmentation. The problems encountered are diverse such as incorrect labeling of isolated fragments, incorrect detection of diverse movements. To solve this problem Hierarchical Attention Networks (HAN) are used which are extensions of Long Short Term Memory (LSTM). The process adopted includes passing the entire video through the HAN’s and acquiring the output word by word. The biggest fear of temporal segmentation is that inaccurate recognition of transitional movements between hand gestures can lead to drastically different output results. For video description generation a two-layer LSTM coupled with a CNN provides promising results.

In this paper, the authors, Shobhit Sinha et. al.[3] are using Convolutional Neural Network to devise a robust model that is able to understand 29 American Sign Language characters consisting of 3 special characters and 26 alphabets. The authors further hosted the model over an instantaneous video interface which gives predictions instantaneously gives the corresponding English alphabets on the screen just like subtitles. This system is looked at as a translator from American Sign Language to English for alphabets. Authors have conceptualized this method in this paper and have explored a few uses that can be executed. ( Shobhit Sinha, Siddhartha Singh, Sumanu Rawat and Aman Chopra, “Real-Time” Prediction of American Sign Language Using Convolutional Neural Networks”, International Conference on Advances in Computing and Data Sciences, ICACDS 2019)

2. 1. Different Acquisition Methods

Different methods are available in the sign language recognition system. Some of them are mentioned below.

* + 1. Camera Based Acquisition

The camera based or vision based approach, acquires the movements of the performer’s hand using a camera. This type of acquisition, the users feel free while signing and the cost of the system is very low and is more powerful because of their flexibility, portability, user friendly.

E. E. Hemayed [4] developed a classification framework based on an Arabic sign language alphabet which transforms signs into voice. The system mainly focuses on passive and simple changeable gestures. The dataset consists of 150 signs and gesture images. The colour images of the gesture inputs are acquired with the help of web camera. To abstract the skin blobs, the YCbCr space was used. To separate the hand gesture, the Prewitt edge detector is used. K-Nearest Neighbor Algorithm (KNN) with Principal Component Analysis (PCA)was used in the recognition stage. The system obtained a classification rate of about 97%.

M. Maraqa [5], proposed an Arabic alphabet classification framework with fully recurrent neural networks. The dataset consists of 900 specimens of Arabic sign language with single handed images. It consists of 30 gestures performed by 2 signers. A digital camera and coloured gloves were also used for the image acquisition in their experiments. Border information, the center of area, and the direction ofthe gesture is used to obtain a feature vector for the gesture. In this system two neural network architectures were used; an Elman recurrent network and a fully recurrent neural network which has full feedback loops. The Elman network achieves a recognition rate of 89.7% although a fully recurrent network enhanced the efficiency of 95.1%.

Al-Jarrah [6], developed an Adaptive Neuro-Fuzzy Inference Systems(ANFIS)to recognize the 30 manual words or gestures from the Arabic sign language. For each gesture, 60 samples were taken by 60 different subjects. Apart from the other hardware the author used camera for image acquisition and it is convenient for the common people. The major parts of the recognition framework include image or video collection, pre-processing, segmentation, feature extraction and classification. In the feature extraction method, they used the border information, the center of area, and the direction of the gesture to elicit a feature vector for the gesture. The system achieves a recognition accuracy of 93.55%.

* + 1. Glove Based Acquisition

To recognize the hand gestures, an electromechanical device is placed inside a glove. So the deaf and dumb people are recommended to wear a glove in order to gather information. This glove is linked to some sensors and these sensors acquire the gesture information related to the shape, movements, locations, etc. of the hands.

M. A. Mohandes [7], proposed a recognition system based on the Arabic language. Here there were both hands are used for the data acquisition. In this system, the activities are done with the help of two cyber gloves and two hand-tracking devices like Flock of Birds (FOB). There were two subjects to perform 100 double-handed sign from which 20 samples was taken into the database. Second-order statistics from sub-frames of the signs were used as features. The dimensionality of the feature vector was diminished with the help of the Principal Component Analysis method. Having SVM with 100 signs, the system recorded an accuracy of 99.6%.

X. Zhan et.al.[8], discussed a system for Chinese sign language CSL) classification systemhavingcomposite of 3D accelerometer sensors and multi-channel electromyography (EMG). By using the EMG signals the gesture divisions were detected.Using combined HMM and Decision Tree method,this system has been classified 72 sign language words. In addition to this, 40 CSLsentences was implemented and recognised to figure out their framework for sequential sign language. The accuracy of overall words and sentences are 93.1% and 73.5% respectively.

B. G. Lee et.al.[9], developed a smart wearable hand device sign language interpretation system. This framework consists of five flex-sensors, two pressure sensors, and a three-axis inertial motion sensor and it is used to recognize the real time 27 ASL alphabets. This system was developed in android based mobile, and the translation of text-to-speech of the alphabets is the one of the functionality. A built-in embedded SVM was used for the recognition. With the help of the pressure sensor placed on the middle finger, the classification rate is increased to 98.2%.

* + 1. Kinect Based Acquisition

Microsoft Kinect is the hardware system used to acquire the 3D level of information or depth and skeletal information. With the 3D positions of the body joints, we can easily track the movements of the body very accurately. For the 3D estimation of the person or objects in the space, Microsoft Kinect provides a good solution. The below discussions is based on the Microsoft Kinect based data-acquisition.

In [10], N. Pugeault et.al., created a fingerspelling ASL system in a continuous environment with the help of Kinect. Based on the Gabor filtering they extracted the hand shape features and Random forest is used to classify the letters of the ASL having a recognition rate of 75%. The authors took around 48000 samples of the signs. i.e. 500 for each sample. This system also included the dynamic letters from the ASL alphabets.

C. Zhang et.al. [11], presented a visual recognition framework based on individual ASL signs. This framework proposed a multimodality way to find out the recognition of ASL. So it includes hand gestures, facial expressions and body movements. ASL videos are acquired by Kinect sensor and a total of 61 video sequences were used. Linear SVM is used as the classifier and the classification rate of the corresponding system was 36.07%.

H. Wanget.al. [12], present a Kinect based SLR framework using RGB-D data. The proposed system was based the minimum. Hand motions are represented using 3D hand and hand postures were represented using HOG features. The hand posture is correctly described by HOG and grouping through K-means clustering. They created two types of datasets namely daily SL dataset and large vocabulary SL dataset having 370 and 1000 daily and isolated signs respectively.

# 

# Chapter 4

# Use Case Analysis

**Chapter 4:** Use Case Analysis

In this chapter, we have mentioned the use case Model of our system and all of the use cases are explained.



## Use Case Model

In this use case diagram, the system is presented from the user’s point of view, how a user can interact with this system. There are two entities, one is the user and the second is our system.

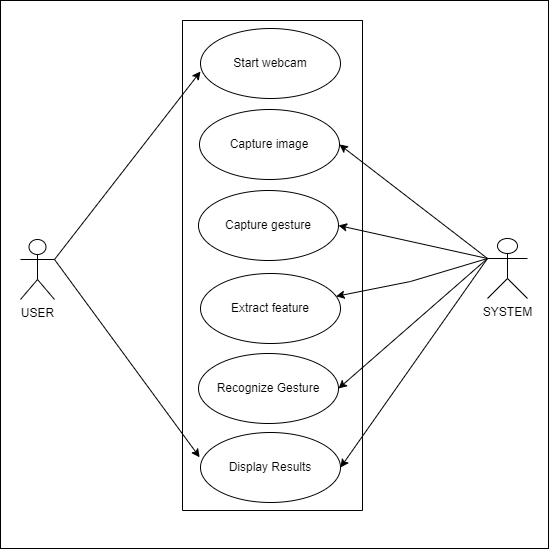


Figure 1: Use Case for Sign Language Recognition system

## Use Cases Description

Description for all the use cases is provided in the form of a table. These table describes in detail various entities of all the use cases.

* + 1. **User Login Function**

Table 4.1: Start webcam

|  |  |
| --- | --- |
| **Usecase name** | Start webcam |
| **Goal in context** | To perform gestures. |
| **Primary actor** | User |
| **Precondition** | Must know how to use the system. |
| **Trigger** | The user wants to use the app. |
| **Scenario** | 1. The user will start the system. 2. User will open the application. 3. Webcam will automatically start. |
| **Exceptions** | -- |
| **Priority** | High |
| **Frequency of use** | Frequent |

|  |  |
| --- | --- |
| **Usecase name** | Capture image |
| **Goal in context** | To capture the hand image |
| **Primary actor** | System |
| **Pre condition** | Web cam must be on |
| **Trigger** | User wants to sign. |
| **Scenario** | 1. The user will start the system. 2. User will open the application. 3. Webcam will automatically start. 4. User will sign infront of the webcam. 5. System will capture the image. |
| **Exceptions** | -- |
| **Priority** | High |
| **Frequency of use** | Frequent |

Table 4.2: Capture image

Table 4.3: Capture Gesture

|  |  |
| --- | --- |
| **Usecase name** | Capture gesture |
| **Goal in context** | To predict the sign |
| **Primary actor** | System |
| **Pre condition** | System and Web cam must be on. |
| **Trigger** | To detect the right sign |
| **Scenario** | 1. User starts the system. 2. User opens the application. 3. Web cam will automatically start. 4. User signs in the web cam. 5. System will capture the image. 6. System captures the gesture from the image. |
| **Exceptions** | -- |
| **Priority** | High |
| **Frequency of use** | Frequent |

Table 4.4: Extract feature

|  |  |
| --- | --- |
| **Usecase name** | Extract feature |
| **Goal in context** | To compare the sign |
| **Primary actor** | System |
| **Pre condition** | User must sign properly |
| **Trigger** | Comparing and predicting of sign |
| **Scenario** | 1. The user starts the system. 2. User opens the application. 3. Web cam will automatically start. 4. User signs in the web cam. 5. System will capture the image. 6. System captures the gesture from the image. 7. Features of the sign are extracted |
| **Exceptions** | -- |
| **To display the accurate resultsPriority** | High |
| **Frequency of use** | Frequent |
|  |  |

Table 4.5: Recognize Gesture

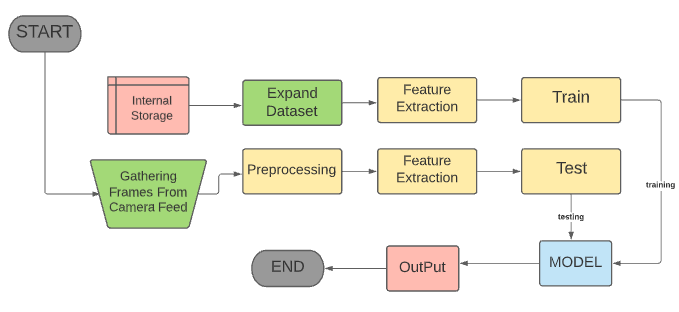
|  |  |
| --- | --- |
| **Usecase name** | Recognize gesture |
| **Goal in context** | To display the correct result |
| **Primary actor** | System |
| **Precondition** | User must sign properly |
| **Trigger** | To display the accurate results |
| **Scenario** | 1. The user starts the system. 2. User opens the application. 3. Webcam will automatically start. 4. User signs in the webcam. 5. System will capture the image. 6. System captures the gesture from the image. 7. Features of the sign are extracted by the trained model. 8. System will then recognize the gesture. |
| **Exceptions** | -- |
| **Priority** | High |
| **Frequency of use** | Frequent |

Table 4.6: Display result

|  |  |
| --- | --- |
| **Usecase name** | Display Result |
| **Goal in context** | Detect the sign and display it as text |
| **Primary actor** | System, User |
| **Precondition** | The gesture must be performed |
| **Trigger** | The user wants to communicate/ know the sign. |
| **Scenario** | 1. The user starts the system. 2. User opens the applicatiWebcam cam will automatically start. 3. User signs in the webcam. 4. System will capture the image. 5. System captures the gesture from the image. 6. Features of the sign are extracted 7. Result is displayed as text. |
| **Exceptions** | -- |
| **Priority** | High |
| **Frequency of use** | Frequent |

# Chapter 5

# Proposed Approach



**Figure 5.1: Proposed Approach**

8. 1. Internal Storage

Dataset is collected and stored in Internal Storage to train the model. It is the first step in the workflow sequence because no processing can be done without a dataset. The picture that is obtained has not been processed in any way.

## Data Augmentation and Pre-Processing

Different techniques are used to increase the amount of data by adding slightly modified copies of already existing data or newly created synthetic data from existing data. It acts as a regularize and helps reduce overfitting when training the model.

Each picture frame is preprocessed to eliminate noise using a variety of filters including erosion, dilation, and Gaussian smoothing, among others. The size of an image is reduced when a color image is transformed to grayscale. A common method for reducing the amount of data to be processed is to convert an image to greyscale. During the testing phase, the test images or videos are also preprocessed to extract the region of interest

* 1. Feature Extraction

Properly optimized feature extraction is the key to effective model construction. The feature vector thus obtained using any one of the feature extraction methods is used for training the classifier. Thus feature extraction is the most crucial step of sign language recognition since the inputs to the classifier are the feature vectors obtained from feature extraction.

CNN is a neural network that extracts input image features and another neural network classifies the image features. The input image is used by the feature extraction network. The extracted feature signals are utilized by the neural network for classification. The neural network classification then works on the basis of the image features and produces the output.

* 1. Training Model

After the CNN feature extraction by convolutional layers, and pooling layer, the model is trained. Artificial Neural Networks are used to classify images. We used 3 different models, 1st was 2d CNN model which was trained on black and white images, 2nd was Inception-resnet-V2 trained on RGB images and the third was VGG also trained on RGB images.

* 1. Results and Discussion

**Results**

****

We are still working towards developing a system although the pipeline we are following is supported by literature to yield similar results. The results, mentioned above, are of some previous work using the same architecture that we are following and we expect to witness similar results. We aim to expand our dataset by adding more classes to it and using multiple feature extraction algorithms which will support the improvement in the results of our model.

# Chapter 6

# Implementation & Results

**Chapter 6:** Implementation & Results

This chapter shows the implementation details of sign language recognition system. It also shows the steps required to achieve the complete sign recognition process. It also introduces a GUI we implement to facilitate interaction with the system and make it very comfortable.

1. Different components of your proposed approach

**Model**

Three Models were trained to see which one gives the best results and all are separately integrated with the front end.

**Backend**

When the image is passed to the prediction function, it returns a matrix of length equivalent to the number of labels containing probability of each class. Then it checks which class in that matrix has the maximum value, and returns that class name.

**GUI**

Front-end is developed using Html-CSS with a very basic design, with a box showing live camera feed, 2 buttons to start/stop the camera and a button to capture and predict the image.

Specific area of the image is cropped from the frame and resized to 200x200 for input. This image is passed to the model as input and the output from the prediction function is then displayed on the screen. Flask is used to integrate the models with frontend.

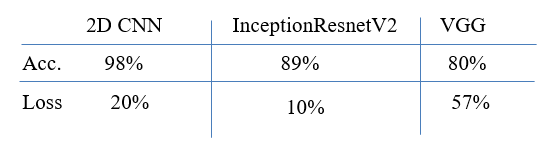
1. Implementation of proposed approach

In this section all the implementation details are presented including the softwares used.

### **6.2.1 Visual Studio Code**

Visual Studio Code is used as IDE in implementation of the backend and frontend of this system. Python 3.8. Keras and tensorflow are used for building and training of our models. Opencv is used for image processing.Html/css is used for front end.

## Results



Above are the Accuracy and loss percentages of three different Deep Learning models we applied on our dataset. 2D CNN model was trained using black and white images of static hand gestures and it performed very well in proper lightning condition.InceptionResnetV2 and VGG were trained on RGB images of static hand gestures. InceptionResnetV2 performed much better than VGG.



## Tools and Techniques

* Python 3.8
* OpenCV for image processing
* Keras – Tensorflow for model training
* Numpy
* HTML-CSS for gui design
* Flask is used for integration with frontend.

# Chapter 7

# Business Plan

**Chapter 7:** Business Plan

Sign Language Recognition System will allow people to understand what a deaf person is saying and communicate easily with them. This product will be a web based application.

2. 2. Business Description

We will use our product as a web based application. This application will be accessible to organizations, institutions, and even government offices where there is a need for interaction with the public. To whomever this product will be sold, its front screen will be designed according to them. For example their name will be mentioned.

## Market Analysis & Strategy

Our first marketing strategy will be to offer a reasonable pricing plan comparatively lower than our competitors. We will choose different platforms to promote and advertise our service like the use of social media platforms, YouTube, Facebook; LinkedIn, etc. At the start, we will also offer a free trial of seven days to the institutions.

## Competitive Analysis

This system will have to face very strong competition due to a lot of research or work already proposed and limited techniques in this field. All of these competitors are struggling for a share of the same market. We will try to make our system more convenient, user-friendly, and accurate for a better market.

## Products/Services Description

This system will provide a service that is converting the signs into a readable form for better communication between deaf community and normal hearing people, and is not common in our society because of many reasons such as unfamiliarity or maybe these systems are not very convenient and affordable.

## SWOT Analysis

* + 1. Strengths

Our system will be very convenient and affordable as no external hardware will be used such as glove or depth mapping sensors, and it will be predicting signs of daily use.

* + 1. Weaknesses

The main weakness is the lower recognition rate and high computing power consumption as compared to sensor-based recognition. Every Country has a different sign language so it will not be correct everywhere.

* + 1. Opportunities

This system can be expanded by adding more gestures to make this system spread and work in more areas.

* + 1. Threats

There is no such threat in our system as it will not require any personal information which can cause any harm.

# Chapter 8

# Conclusion & Future Work

**Chapter 8:** Conclusion & Future Work

Sign language is the tongue of the people who neither speak nor hear. Without sign language, they do not exist in the world. The human-machine interaction is developed through the gesture recognition system. Past work on sign language processing has largely been conducted by experts in different domains separately, limiting real-world utility. Many types of research have been conducted on sign gesture recognition using different techniques like ANN, LSTM, and 3D CNN. However, most of them require extra computing power. On the other hand, our research paper requires low computing power. In our research, we proposed to normalize and rescale our images to 64 pixels (2D CNN model) and 200 pixels (InceptionResnetV2 and VGG model) in order to extract features and make the system more robust. Currently, we are working on word-level sign language recognition with static hand images. CNN feature extractions (Convolutional and Pooling layers) are used to extract features from preprocessed images and a neural network is used for classifying. Html CSS is used for front-end design. Flask framework is used for integration of our models with front-end. We look forward to using more words in our datasets and improving the model so that it recognizes more gestures while at the same time getting high accuracy.

1. 1. Achievements and Improvements

Our Project will predict the meaning of the signs performed by the signer in front of the camera using different AI and Machine Learning techniques, we trained 3 models i.e. 2D CNN with a dataset of 45000 B/W images of 26 classes(84% accuracy), InceptionresnetV2 with a dataset of 87000 RGB images of 29 classes(96% accuracy) and VGG. This work can be extended to predict the longer sequences of text and predicting multiple signers at the same time. Predicting multiple signers in a single video feed is something not explored at all, one could extend our work to test the potential of the premise.

* 1. Critical Review

This work is done on Static images and it recognizes only static gestures where sign language cannot be completely covered only on static images e.g. words with moving hand gestures, to predict the moving gestures a sequence of frames or video classification methods are used and the complexity of the system also increases.

However there are many basic words whose signs does not require moving hands e.g. start, stop, half, child, call, okay etc., and these words can be predicted using static images.

* 1. Future Enhancements/Recommendations

Furthermore, this work can be extended and improved by exploring the potential of tokenization of full length video into multiple words.

## References

[1] Lean Karlo, et. al., “Static Sign Language Recognition Using Deep Learning”, International Journal of Machine Learning and Computing, 2019.

[2] Jie Huang, Wengang Zhou, Qilin Zhang, Houqiang Li, Weiping Li, “Video-Based Sign Language Recognition without Temporal Segmentation”, The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18), 2018

[3] Shobhit Sinha, Siddhartha Singh, Sumanu Rawat and Aman Chopra, “Real Time Prediction of American Sign Language Using Convolutional Neural Networks”, International Conference on Advances in Computing and Data Sciences, ICACDS 2019

[4] E. E. Hemayed and A. S. Hassanien, “Edge-based recognizer for Arabic sign language alphabet (ars2v-arabic sign to voice),” in Computer Engineering Conference (ICENCO), 2010 International. IEEE, 2010, pp. 121–127.

[5] M. Maraqa and R. Abu-Zaiter, “Recognition of Arabic sign language (ARSL) using recurrent neural networks,” in Applications of Digital Information and Web Technologies, 2008. ICADIWT 2008. First International Conference on the. IEEE, 2008, pp. 478–481.

[6] O. Al-Jarrah and A. Halawani, “Recognition of gestures in Arabic sign language using neuro-fuzzy systems,” Artificial Intelligence, vol. 133, no. 1, pp. 117–138, 2001.

[7] M. A. Mohandes, “Recognition of two-handed Arabic signs using the cyber glove,” Arabian Journal for Science and Engineering, vol. 38, no. 3, pp. 669–677, 2013.

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

[9] B. G. Lee and S. M. Lee, “Smart Wearable Hand Device for Sign Language Interpretation System with Sensors Fusion,” IEEE Sens. J., vol. 18, no. 3, pp. 1224–1232, 2018.)

[10] N. Pugeault, R. Bowden, “Spelling it out: Real-time ASL fingerspelling recognition”, International Conference on Computer Vision Workshops, 2011, pp. 1114–1119.

[11] C. Zhang, Y. Tian, and M. Huenerfauth, “Multi-modality American Sign Language recognition,” 2016 IEEE Int. Conf. Image Process., pp. 2881–2885, 2016.

[12] H. Wang, X. Chai, X. Chen, “Sparse observation (SO) alignment for sign language recognition”, Neurocomputing 175 (2016) 674–685.