

An Efficient Approach for Interpretation of Indian Sign Language using Machine Learning

Rasha Anjum, Amatun Noor Sadaf, Maheen Sami,
Kamel Alikhan Siddiqui ¹

¹ Associate Professor



Published in IJIRMPS (E-ISSN: 2349-7300), Volume 11, Issue 1, (January-February 2023)

License: [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/)



Abstract

Non-verbal communication involves the usage of Sign Language. The sign language is used by people with hearing / speech disabilities to express their thoughts and feelings. But normally, people find it difficult to understand the hand gestures of the specially challenged people as they do not know the meaning of the sign language gestures. Usually, a translator is needed when a speech / hearing impaired person wants to communicate with an ordinary person and vice versa. In order to enable the specially challenged people to effectively communicate with the people around them, a system that translates the Indian Sign Language (ISL) hand gestures of numbers (1-9), English alphabets (A-Z) and a few English words to understandable text and vice versa has been proposed in this paper. This is done using image processing techniques and Machine Learning algorithms. Different neural network classifiers are developed, tested and validated for their performance in gesture recognition and the most efficient classifier is identified.

Keywords: Machine Learning, Python, ISL and Gesture Recognition

1. Introduction

Sign Languages vary throughout the world. There are around 300 different sign languages used across various parts of the world. This is because sign languages were developed naturally by people belonging to different ethnic groups. Perhaps, India does not have a standard sign language. Lexical variations and different dialects of Indian Sign Language are found in different parts of India. But, recently, efforts had been taken to standardize the Indian Sign Language (ISL). The ISL hand gestures are divided into two broad categories: (a) Static gestures (b) Dynamic gestures. The static ISL hand gestures of numbers (0-9), English alphabets (A-Z), and some English words. According to the 2011 census, there are around 50 lakh people in India with speech/hearing impairments. But, there are only less than 300 educated and trained sign language interpreters in India. So, people with speech/hearing impairments tend to become isolated and lonely, as they face difficulties in communicating with other normal people. This has a tremendous effect on both their social and working life. Due to the above mentioned challenges that the specially challenged people face, an automated real-time system that could translate English words to ISL and vice versa has been proposed in this paper. This system makes it easy for the specially challenged people to communicate effectively with the rest of the world. This could enhance their abilities and make them realize that they can do better in life. The proposed system performs two major tasks: (a) Gesture to Text conversion (b) Speech to Gesture conversion. Gesture to text conversion is

done using neural network classifiers. Speech to gesture conversion is done using Google Speech Recognition API.

This paper focuses on conversion of standard Indian Sign Language gestures to English, and conversion of English words (spoken) to Indian Sign Language gestures with highest possible accuracy. For this, different neural network classifiers are developed, and their performance in gesture recognition is tested. The most accurate and efficient classifier is chosen and is used to develop an application that converts ISL gestures to their corresponding English text, and speech to the corresponding ISL gestures.

1.1. Input Design

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. the following things are considered during Input Design:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

Objectives of Input Design

Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in 3 maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

1.2. Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should identify the specific output that is needed to meet the requirements. Select methods for presenting information. Create document, report, or other formats that contain information produced by the system. The output form of an information system should accomplish one or more of the following objectives: (a) convey

information about past activities, current status or projections of the future; (b) Signal important events, opportunities, problems, or warnings; (c) trigger an action; (d) confirm an action.

2. Hand Gesture Recognition Software based on ISL

Hand gestures are a powerful environment for communicating with communities with intellectual disability. It is useful for connecting people and computers. The expansion potential of this system can be known in public places where deaf people are communicating with ordinary people to send messages. In this article, we have provided a system of recognizing gestures continuously with the Indian Sign Language (ISL), which both hands are used to make every gesture. Gesture recognition continues to be a daunting task. We tried to fix this problem using the key download method. These key tips are useful for breaking down the sign language gestures into the order of the characters, as well as deleting unsupported frameworks. After the splitting gear breaks each character is regarded as a single and unique gesture. Pre-processing gestures are obtained using histogram (OH) with PCA to reduce the dimensions of the traits obtained after OH. The experiments were performed on our live ISL dataset, which was created using an existing camera.

3. System Analysis

3.1. Existing System

According to the 2011 census, there are around 50 lakh people in India with speech/hearing impairments. But, there are only less than 300 educated and trained sign language interpreters in India. So, people with speech/hearing impairments tend to become isolated and lonely, as they face difficulties in communicating with other normal people. This has a tremendous effect on both their social and working life. Due to the above mentioned challenges that the specially challenged people face, an automated real-time system that could translate English words to ISL and vice versa has been proposed in this paper. This system makes it easy for the specially challenged people to communicate effectively with the rest of the world. This could enhance their abilities and make them realize that they can do better in life.

Disadvantages of the Existing System

Recently, efforts had been taken to standardize the Indian Sign Language (ISL). The ISL hand gestures are divided into two broad categories: (a) static gestures, and (b) dynamic gestures. But the sign are not showing correct results due to small dataset.

3.2. Algorithm

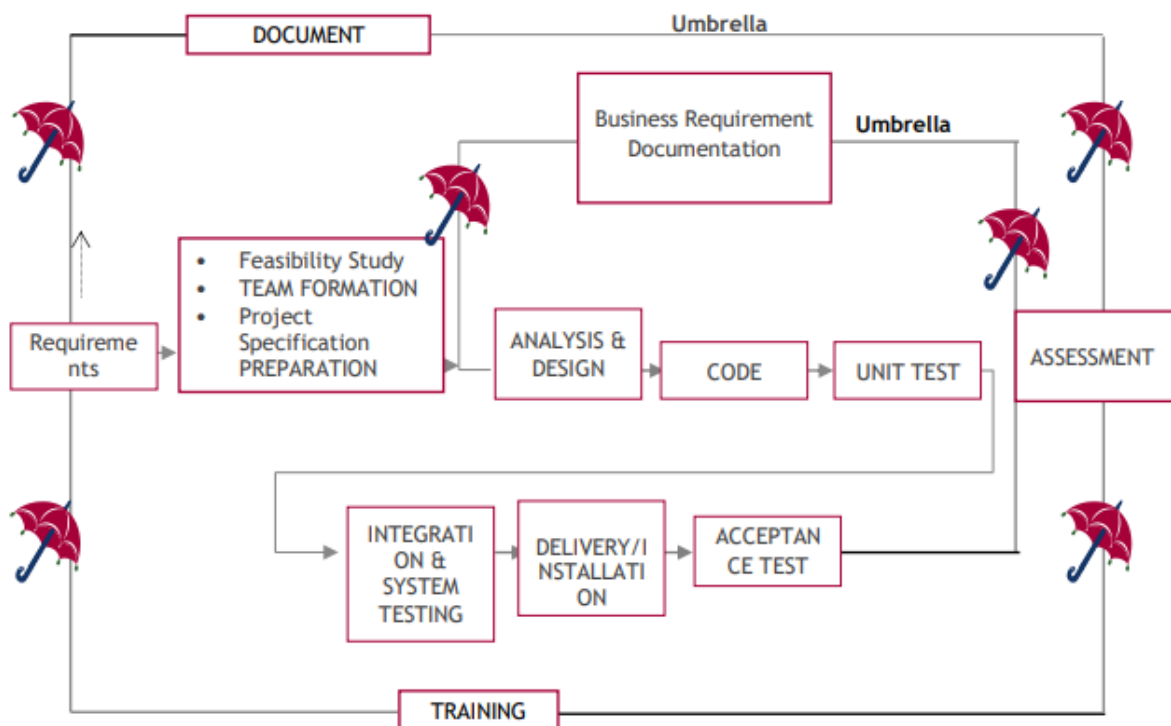
CNN 3.2 proposed system as mentioned in the above section, the proposed system for ISL interpretation performs two major tasks: (a) Gesture to Text conversion, (b) Speech to Gesture conversion. Gesture to text conversion involves four major steps: (a) Dataset collection, (b) Segmentation, (c) Feature Extraction, (d) Classification. An image dataset consisting of ISL hand gestures of 9 numbers (1-9), 26 English alphabets and a few English words is collected. After the dataset is ready, all the images in the dataset are pre-processed to mask the unwanted areas and to remove noise from the image. Hence, pre-processing the images prior to feeding them to a classifier improves the efficiency, accuracy and performance of the system. Hence, this step is very important in the image classification process. Here, feature extraction is done using the Speeded-Up Robust Feature (SURF) method. The SURF is used as a feature descriptor or as a feature detector. It is often used for applications like object detection, image classification etc. It is a fast and robust algorithm for representing and comparing images. It acts as a

blob detector in an image. The SURF features are calculated by finding the interest points in the image that contain the meaningful features using the determinants of Hessian matrices. For each interest point found in the previous process, the scale invariant descriptors are constructed.

Advantages of the Proposed System

- Resizing the images to the same size (for uniformity)
- Conversion of RGB image to Grayscale image
- Median Blur
- Skin Masking and Skin Detection
- Canny Edge Detection (to detect sharp edges in the image)

Figure: Umbrella Model

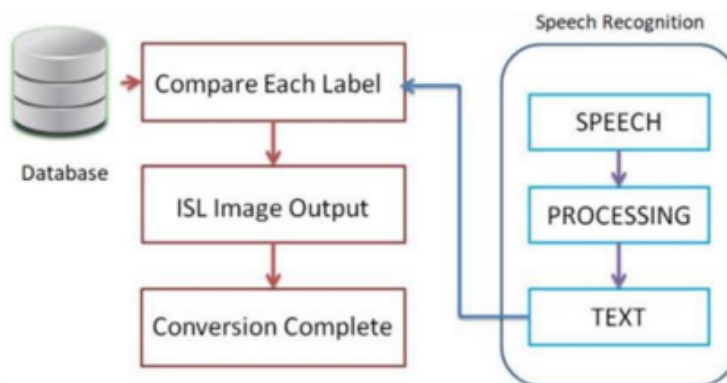


System Design

System Architecture

The architecture of a system reflects how the system is used and how it interacts with other systems and the outside world. It describes the interconnection of all the system's components and the data link between them. The architecture of a system reflects the way it is thought about in terms of its structure, functions, and relationships.

Figure: System Architecture



UML Diagram

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object oriented computer software. In its current form, UML is comprised of two major components: a meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. UML is a very important part of developing objects oriented software and the software development process. UML uses mostly graphical notations to express the design of software projects.

Goals

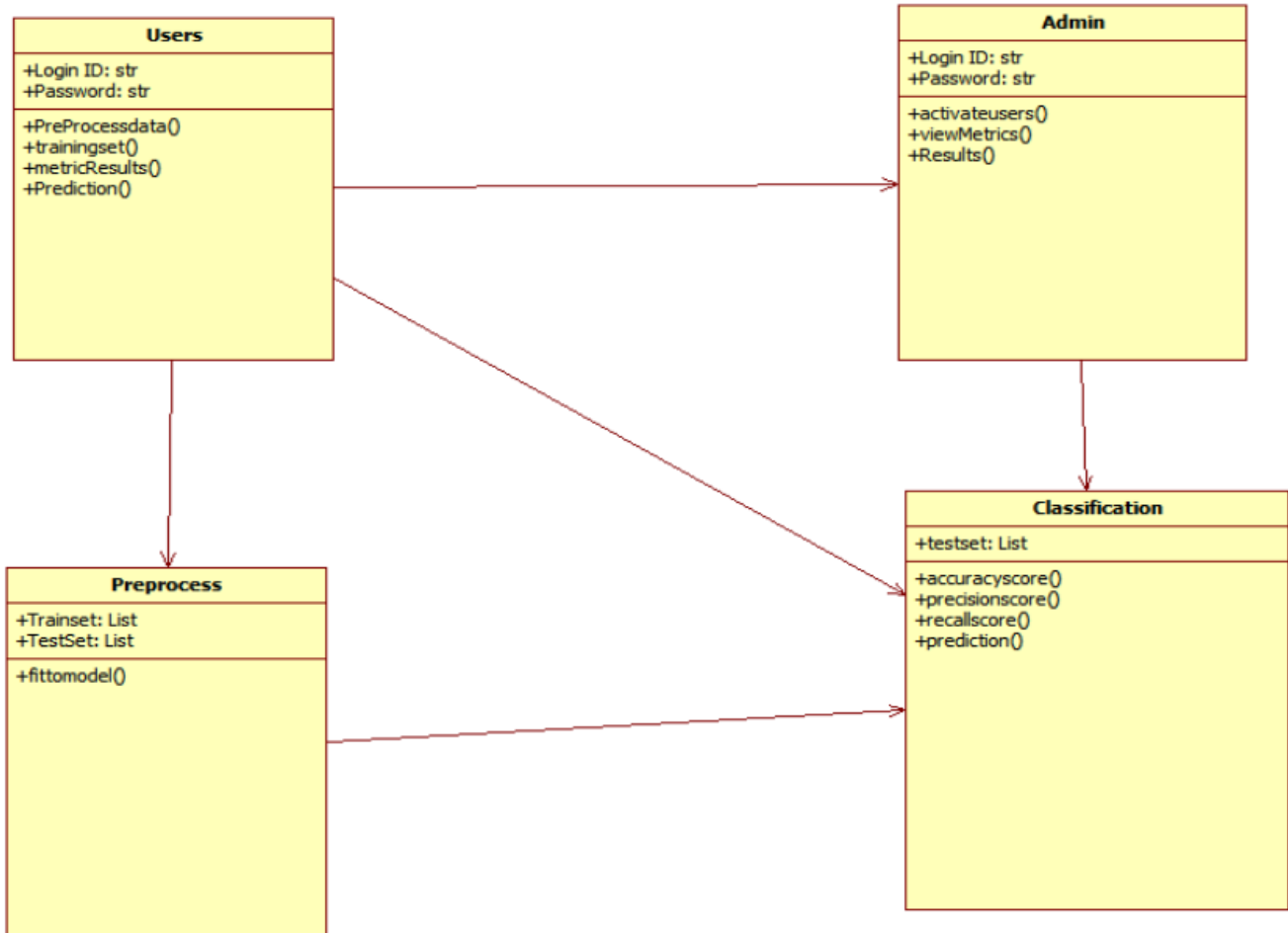
The primary goals in the design of the UML are as follows:

- Provide users a ready-to-use, expressive visual modeling language.
- Provide extensibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development process.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of OO tools market.
- Support higher level development concepts such as collaborations, frameworks.

Class Diagram

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes.

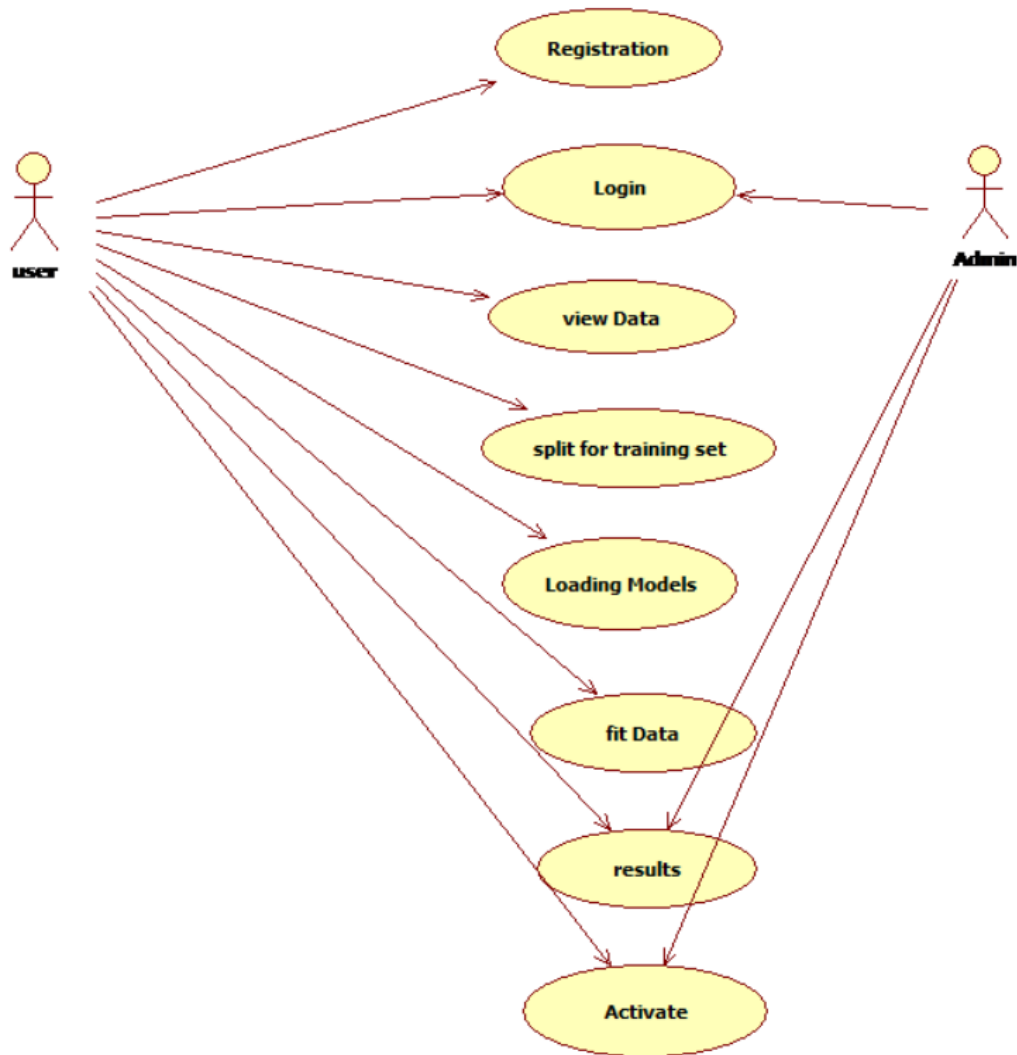
Figure: Class Diagram



Use Case Diagram

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

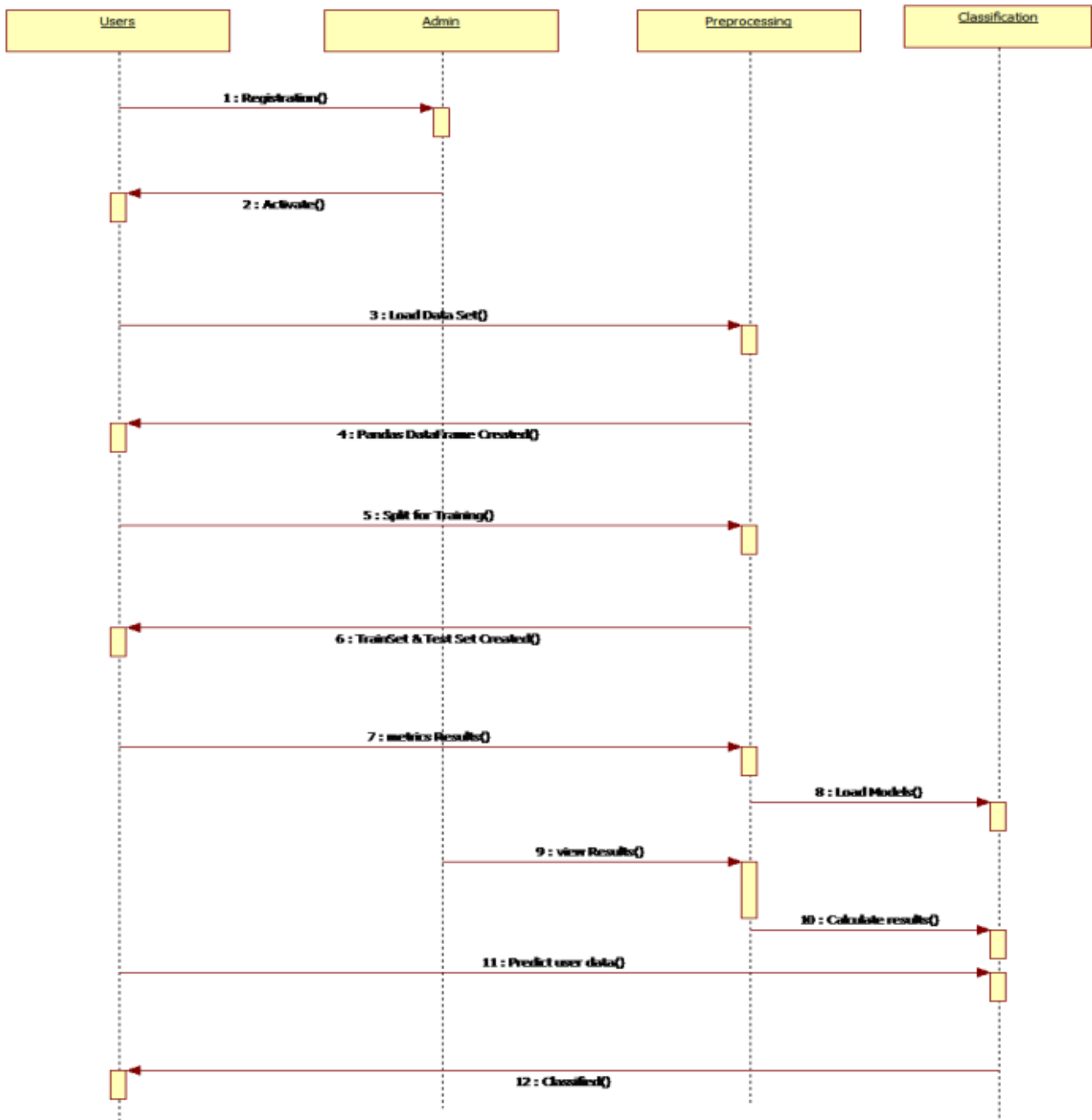
Figure: Use-case Diagram



Sequence Diagram

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

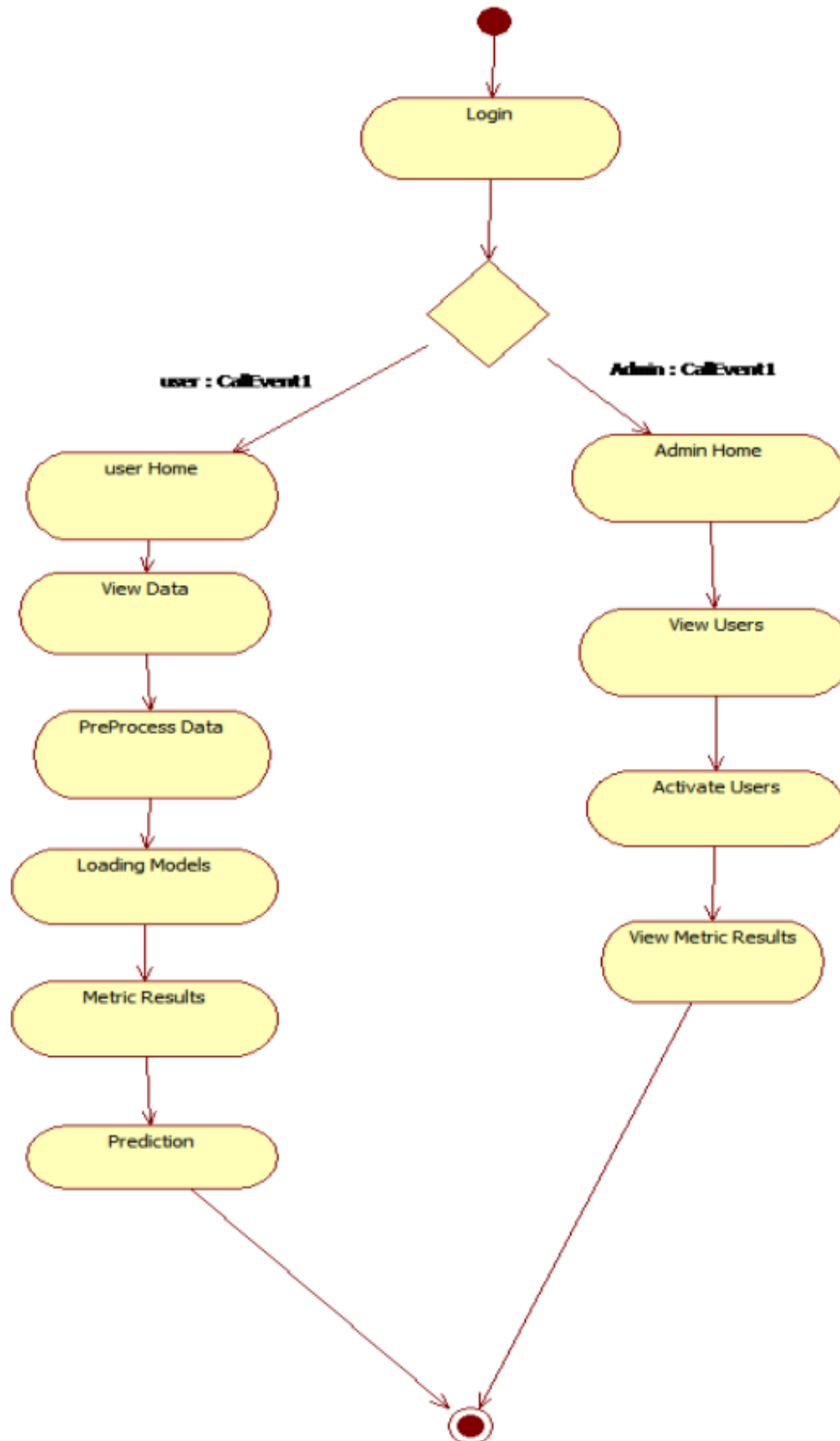
Figure: Sequence Diagram



Activity Diagram

Activity diagrams are graphical representations of workflows of step-wise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

Figure: Activity Diagram

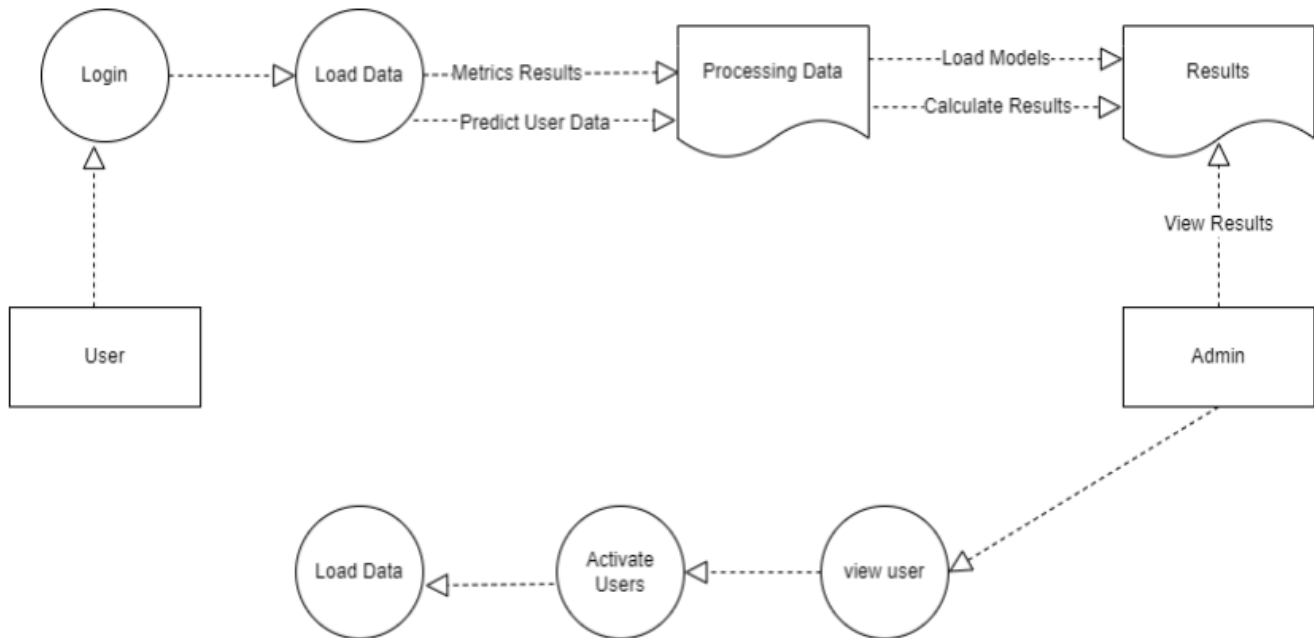


Data Flow Diagram

DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system. Data Flow Diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system. DFD shows how the information moves through the system and how it is modified by a series of

transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

Figure: Data Flow Diagram



Implementation

Introduction of Technologies Used

Python

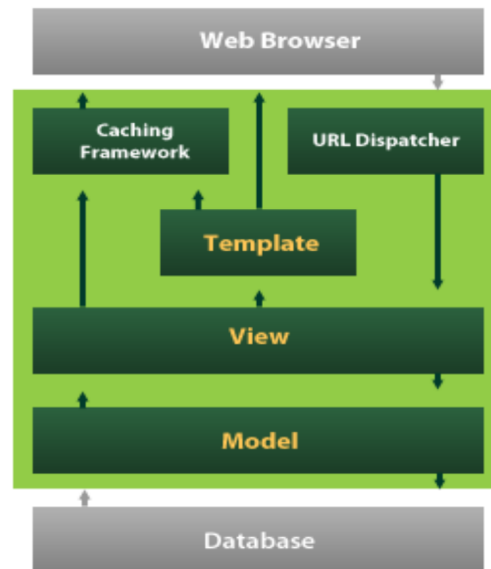
Python is a general-purpose language. It has wide range of applications from web development (like: Django and Bottle), scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user interfaces (Pygame, Panda3D). The syntax of the language is clean and length of the code is relatively short. It's fun to work in Python because it allows you to think about the problem rather than focusing on the syntax. Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber, etc. The biggest strength of Python is huge collection of standard library which can be used for the following:

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia
- Scientific computing

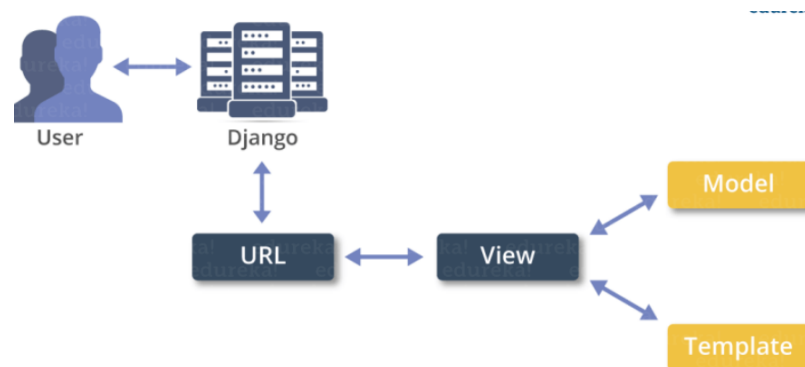
- Text processing and many more.

Django

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source. Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models.



Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.



Testing

Testing is one of the most important tasks in project is the phase in which one has to be cautions because all the efforts undertaken during the project will be very interactive. Testing is the most crucial stage in achieving successful system and giving the users confidence that the new system is workable and effective. Each program is tested individually at the time of development using the sample data and has verified that these programs link together in the way specified in the program specification. The computer system and its environment are tested to the satisfaction of the user.

Implementation

The implementation phase is less creative than system design. It is primarily concerned with user training, and file conversion. The system may be requiring extensive user training. The initial parameters of the system should be modified as a result of a programming. A simple operating procedure is provided so that the user can understand the different functions clearly and quickly. Different reports can be obtained either on a printer, which is available at the disposal of the user. The proposed system is very easy to implement. In general implementation is used to mean the process of converting a new or revised system design into an operational one.

Testing

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. Actually testing is the state of implementation which aimed at ensuring that the system works accurately and efficiently before the actual operation commence. The following is the description of the testing strategies, which were carried out during the testing period.

Software Testing

Software testing can be stated as the process of verifying and validating whether a software or application is bug-free, meets the technical requirements as guided by its design and development, and meets the user requirements effectively and efficiently by handling all the exceptional and boundary cases.

The process of software testing aims not only at finding faults in the existing software but also at finding measures to improve the software in terms of efficiency, accuracy, and usability. It mainly aims at measuring the specification, functionality, and performance of a software program or application. Software testing can be divided into two steps:

1. **Verification:** It refers to the set of tasks that ensure that the software correctly implements a specific function.
2. **Validation:** It refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements.

Verification: “Are we building the product right?”

Validation: “Are we building the right product?”

What are different types of software testing?

Software Testing can be broadly classified into two types:

1. Manual Testing

Manual testing includes testing software manually, i.e., without using any automation tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behavior or bug. There are different stages for manual testing such as unit testing, integration testing, system testing, and user acceptance testing. Testers use test plans, test cases, or test scenarios to test software to ensure the completeness of testing. Manual testing also includes exploratory testing, as testers explore the software to identify errors in it.

2. Automation Testing

Automation testing, which is also known as Test Automation, is when the tester writes scripts and uses another software to test the product. This process involves the automation of a manual process. Automation Testing is used to re-run the test scenarios quickly and repeatedly, that were performed manually in manual testing. Apart from regression testing, automation testing is also used to test the application from a load, performance, and stress point of view. It increases the test coverage, improves accuracy, and saves time and money when compared to manual testing.

What are the different types of Software Testing Techniques?

Software testing techniques can be majorly classified into two categories:

1. Black-box Testing

The technique of testing in which the tester doesn't have access to the source code of the software and is conducted at the software interface without any concern with the internal logical structure of the software is known as black-box testing.

2. White-box Testing

The technique of testing in which the tester is aware of the internal workings of the product, has access to its source code, and is conducted by making sure that all internal operations are performed according to the specifications is known as white-box testing.

Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional Test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:

- **Valid Input:** Identified classes of valid input must be accepted.
- **Invalid Input:** Identified classes of invalid input must be rejected.
- **Functions:** Identified functions must be exercised.

- **Output:** Identified classes of application outputs must be exercised.

Systems / Procedures

Interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, data fields, predefined processes, and successive processes must be considered for testing systematic coverage pertaining to identify business process flows. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test Strategy and Approach

Field testing will be performed manually and functional tests will be written in detail.

Test Objectives

All field entries must work properly.

Pages must be activated from the identified link.

The entry screen, messages and responses must not be delayed.

Features to be Tested

Verify that the entries are of the correct format.

No duplicate entries are be allowed.

All links are taking the users to the correct pages.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

System Testing

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to user the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

Module Testing

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example, the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. The comparison shows that the results proposed system works efficiently than the existing system. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

Integration Testing

After module testing, integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system.

Acceptance Testing

When that user find no major problems with its accuracy the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

Conclusion

From the results obtained, it is inferred that the SVM classifier along with the K-means clustering and BoV classifiers is best suited for gesture recognition. A user friendly application that can interpret Indian Sign Language has been developed using the most efficient SVM classifier (for gesture to text conversion) and Google Speech Recognition API (for speech to gesture conversion). Thus, a more reliable sign language interpretation system has been developed. Communication is a vital activity of human beings to live, as they can express their feeling, encourage cooperation and social bond, share their idea, and work together in society through communication only. People who are not able to hear or speak (hearing-impaired people) uses sign language as a mean of communication. Like spoken language, sign language also emerges and evolves naturally within hearing-impaired persons. It is a visual form of communication and in each country/region, where the hearing-impaired.

References

- [1] Cheok Ming Jin, Zaid Omar, Mohamed Hisham Jaward, "A Mobile Application of American Sign Language Translation via Image Processing Algorithms", in IEEE Region 10 Symposium, on IEEEExplore, 2016.
- [2] Sanket Kadam, Aakash Ghodke, Sumitra Sadhukhan, "Hand Gesture Recognition Software based on ISL", IEEE Xplore, 20 June 2019.
- [3] Kartik Shenoy, Tejas Dastane, Varun Rao, Devendra Vyavaharkar, "Real-time Indian Sign Language (ISL) Recognition", IEEE Xplore, 18 October 2018.

- [4] T. Raghuveera, R. Deepthi, R. Mangalashri, R. Akshaya, “A depth based ISL Recognition using Microsoft Kinect”, ScienceDirect, 2018.
- [5] Muthu Mariappan H., Gomathi V., “Real-time Recognition of ISL”, IEEE Xplore, 10 October 2019.
- [6] G. Ananth Rao, P.V.V. Kishore, “Selfie video based continuous Indian sign language recognition system”, ScienceDirect, 2018.
- [7] G. Ananth Rao, P.V.V. Kishore, “Sign Language Recognition based on Hand and Body Skeletal Data”, IEEE, 2018.
- [8] Suharjito, Ricky Anderson, Fanny Wiryana, Meita Chandra Ariesta, Gede Putra Kusum, “Sign Language Recognition Application Systems for Deaf-Mute People: A Review based on Input-Process-Output”, ScienceDirect, 2017.