SIGN LANGUAGE PREDICTION

Project Report

Version 1.0

Project Work Phase – II (EAI852)

BACHELOR OF TECHNOLOGY (CSE)

SPECIALIZATION-

ARTIFICIAL INTELLIGENCE

MACHINE LEARNING

AND DEEP LEARNING

Project Guide:

Dr. Saurabh Pathak

Assistant Professor

CCSIT, TMU

Submitted By:

Mansi Jain (TCA1959015)

JAN, 2023

FACULTY OF ENGINEERING & COMPUTING SCIENCES

TEERTHANKER MAHAVEER UNIVERSITY, MORADABAD

DECLARATION

We hereby declare that this Project Report titled **SIGN LANGUAGE PREDICTION** submitted by us and approved by our project guide, Faculty of Engineering & Computing Sciences. Teerthanker Mahaveer University, Moradabad, is a bonafide work undertaken by us and it is not submitted to any otherUniversity or Institution for the award of any degree diploma / certificate or published any time before.

Project ID		
Student Name	Mansi Jain (TCA1959015)	

Project Guide	Dr. Saurabh Pathak	

Table of Contents

1 Project Title

Sign Language Prediction: A Machine Learning Approach for Gesture Recognition

2 Problem Statement

Sign language is a unique form of communication used by individuals with hearing impairments to express their thoughts and interact with others. However, the communication gap between sign language users and non-sign language users remains a significant challenge. This report aims to address this issue by developing a sign language prediction system using machine learning techniques. The system aims to accurately recognize and interpret sign language gestures in real-time, enabling seamless communication between sign language users and non-sign language users.

- The introduction section will provide an overview of the importance of sign language recognition and its impact on inclusivity and accessibility for individuals with hearing impairments. It will also highlight the limitations of traditional communication methods and the need for an efficient sign language prediction system.
- This section will review existing literature, research papers, and projects related to sign language recognition and prediction. It will analyze various techniques and algorithms employed in previous studies and discuss their strengths and weaknesses.
- The dataset used for training and evaluation will be described in this section. It will explain the process of data collection, including the selection of gestures and the annotation process. Additionally, the preprocessing steps, such as data cleaning, normalization, and feature extraction, will be discussed.
- The methodology section will outline the machine learning approach used for sign language prediction. It will describe the chosen algorithm(s), such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), or a combination of both. The training process, parameter tuning, and model evaluation techniques will also be explained.

- This section will discuss the implementation details of the sign language prediction system. It will cover the software and hardware requirements, programming languages, and libraries utilized for developing the system. Additionally, any user interface or visualization components will be described.
- The results obtained from the trained model will be presented in this section. It will include accuracy metrics, confusion matrices, and performance comparisons with existing methods. Furthermore, the evaluation of the system's real-time prediction capabilities will be discussed, along with any limitations or challenges encountered during the evaluation.
- The discussion section will analyze the findings and provide insights into the performance of the sign language prediction system. It will highlight the strengths, weaknesses, and potential improvements for future iterations. Additionally, it will address the ethical considerations, usability, and potential applications of the system.
- The conclusion will summarize the key findings of the study and highlight the significance of sign language prediction in promoting inclusivity and accessibility. It will also discuss the system's potential impact and future directions for research and development.

3 Project Description

The Sign Language Prediction project aims to develop a system that leverages machine learning techniques to predict sign language gestures and facilitate communication between individuals who are hearing impaired or have difficulty understanding sign language. By accurately recognizing and translating sign language into textual or auditory outputs, this project aims to bridge the communication gap and empower individuals with hearing disabilities to interact effectively with others in various settings.

Sign language is a visual communication system used by individuals with hearing impairments to express their thoughts, ideas, and emotions. While sign language is a rich and expressive form of communication, it presents a barrier for those who do not understand the language or lack the ability to interpret gestures accurately. The Sign Language Prediction project seeks to address this challenge by developing a system that can predict sign language gestures and convert them into understandable formats, such as text or speech.

The primary objective of the Sign Language Prediction project is to create a robust and accurate prediction model that can recognize and interpret sign language gestures in real-time. By achieving this objective, the project aims to provide individuals with hearing impairments the ability to communicate effectively with others who may not understand sign language. The project also intends to raise awareness about the importance of inclusivity and accessibility in society.

3.1 Scope of the Work

The purpose of this report is to outline the scope of work for developing a sign language prediction system. Sign language prediction aims to provide real-time translation of spoken language into sign language for individuals who are deaf or hard of hearing. This report will cover the key components, objectives, and tasks involved in the development of such a system.

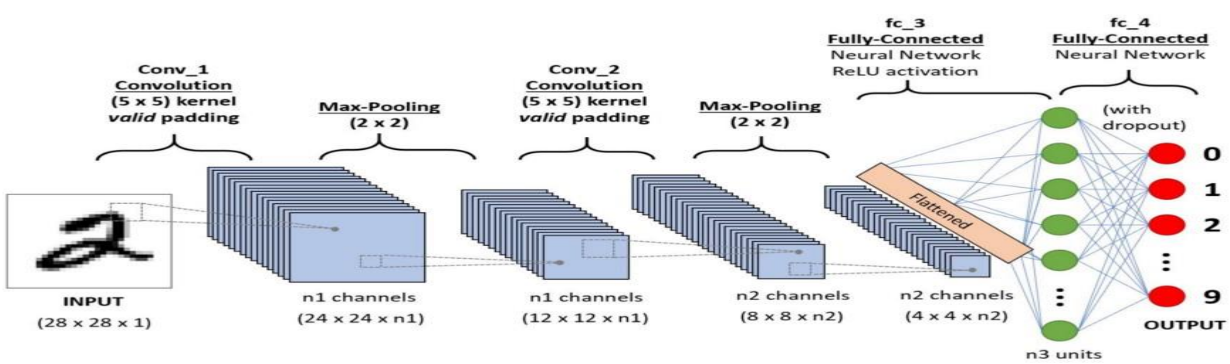
- Speech Recognition: Collect and preprocess a diverse dataset of spoken language samples, encompassing different accents, dialects, and speaking styles, to train the speech recognition model. Implement the selected speech recognition algorithm or framework and develop the necessary infrastructure to handle real-time speech input. Train and fine-tune the speech recognition model using the collected dataset, optimizing for accuracy and robustness in different linguistic contexts. Evaluate the performance of the speech recognition module through extensive testing and validation, incorporating metrics such as word error rate and recognition speed. Continuously improve the speech recognition component by iteratively updating and retraining the model based on user feedback and additional data.
- Gestures Prediction: Design and develop a machine learning or deep learning model capable of predicting sign language gestures based on the textual representation obtained from the speech recognition module. Preprocess and augment the sign language dataset to enhance model generalization and handle variations in signing styles. Train and validate the gesture prediction model using the collected dataset, employing techniques

such as recurrent neural networks (RNNs) or transformers. Optimize the model's performance and accuracy by fine-tuning hyperparameters, adjusting architecture, and leveraging transfer learning approaches. Conduct thorough evaluation and testing of the gesture prediction module, measuring metrics like prediction accuracy, latency, and robustness across different sign language variants. Incorporate feedback from sign language users to refine the gesture prediction model, improving its ability to capture nuanced expressions and improve user experience.

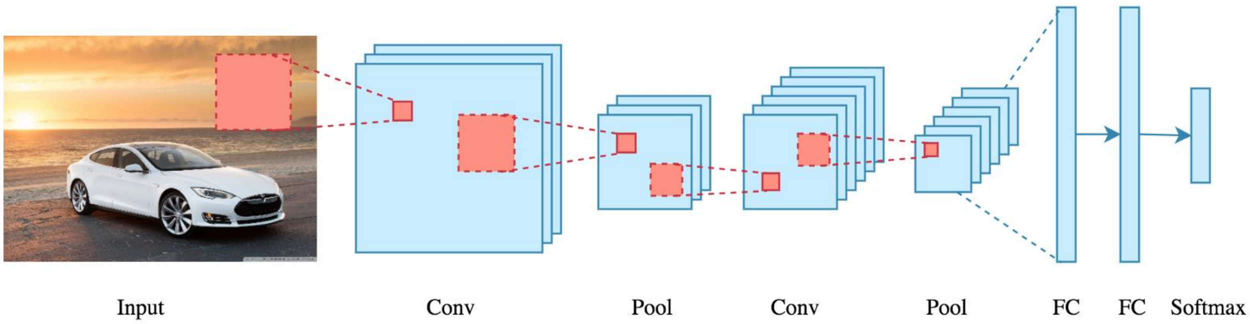
3.2 Project Modules

This report presents a project module focused on Sign Language Prediction. The aim of this project is to develop a system that can predict sign language gestures based on input from video or image data. Sign language is an essential means of communication for individuals with hearing impairments, and automating the prediction process can help bridge the communication gap between hearing-impaired individuals and those who do not understand sign language. This report provides an overview of the project, including the problem statement, objectives, methodology, implementation details, and potential applications.

CNN Model: - CNNs were first developed and used around the 1980s. The most that a CNN could do at that time was recognize handwritten digits. It was mostly used in the postal sectors to read zip codes, pin codes, etc. The important thing to remember about any deep learning model is that it requires a large amount of data to train and also requires a lot of computing resources. This was a major drawback for CNNs at that period and hence CNNs were only limited to the postal sectors and it failed to enter the world of machine learning.

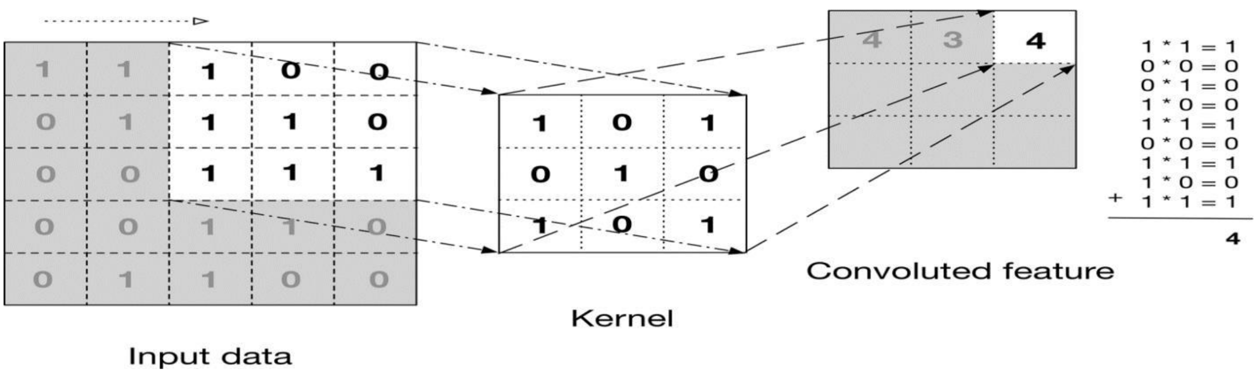


In deep learning, a convolutional neural network (CNN/ConvNet) is a class of deep neural networks, most commonly applied to analyses visual imagery. Now when we think of a neural network, we think about matrix multiplications but that is not the case with ConvNet. It uses a special technique called Convolution. Now in mathematics convolution is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other.

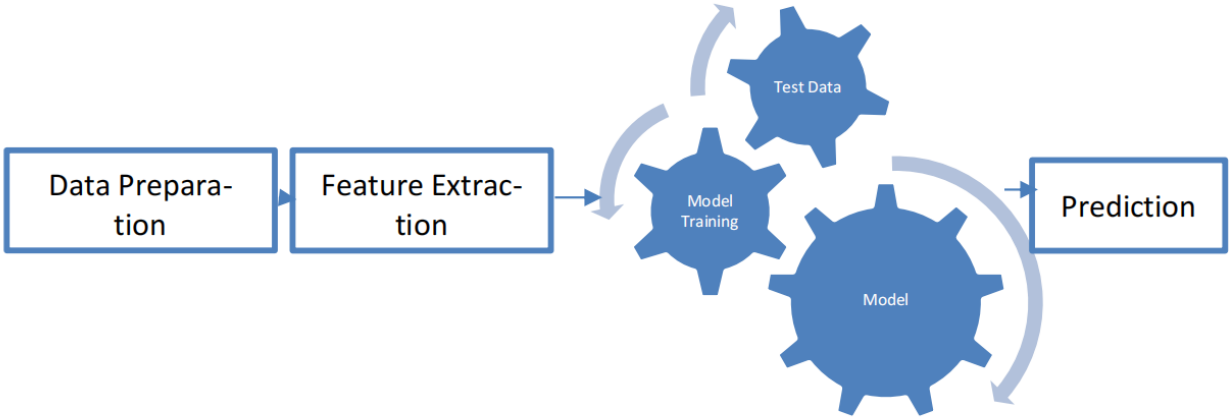


But we don't really need to go behind the mathematics part to understand what a CNN is or how it works.

Bottom line is that the role of the ConvNet is to reduce the images into a form that is easier to process, without losing features that are critical for getting a good prediction.



Application Module: - This module is divided into three parts i.e., Crop Recommendation, Fertilizer Recommendation and Disease Prediction. For crop recommendation we take several inputs of soil testing report and taking location as an input and through weather API we fetch the weather report of that location such as rainfall and weather report. For fertilizer recommendation we take soil report and the crop and based upon it it will recommend the fertilizer. For disease prediction model it will take image as an input and then classify the result and predict the disease and crop and give prevention method and solutions.

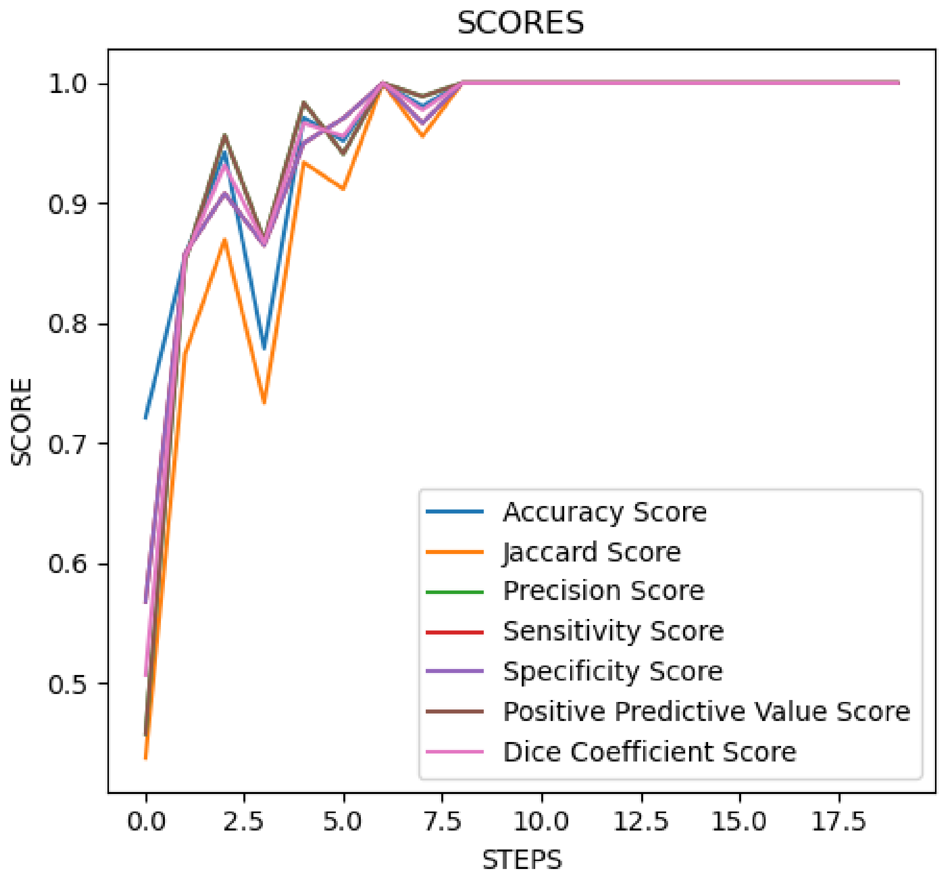


The choice of algorithm will depend on the specific problem that is being addressed. For example, decision trees are often used for classification problems, while neural networks are often used for regression problems.

The crop, fertilizer, and disease prediction model can be used to help farmers make better decisions about their crops. For example, the model can be used to recommend the best crops to grow in a particular region, based on the soil conditions and climate. The model can also be used to recommend the best fertilizers to use for a particular crop, and to detect and diagnose plant diseases.

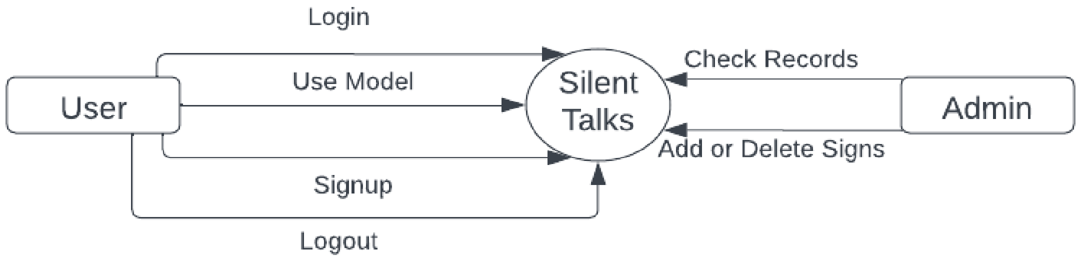
The use of machine learning to predict crop, fertilizer, and disease can help farmers to improve their yields and reduce their costs. This can lead to increased profits for farmers and improved food security for the world.

Below is graph given with accuracy, jaccard score, precision score, sensitivity score, specificity score and dice coefficients.



3.3 Context Diagram (High Level)

A zero-level data flow diagram (DFD) is the highest level of a DFD. It shows the overall system or process being analyzed or modeled. It is designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers. The process symbol represents the overall system or process being analyzed or modeled. The external entity symbols represent the people, organizations, or other systems that interact with the system. The data flow lines represent the flow of data between the process and external entity symbols. Zero-level DFDs are a useful tool for understanding the overall structure of a system. They can be used to identify the key components of the system and the relationships between them.



4 Implementation Methodology

The implementation methodology for a project on Sign Language Prediction can involve several steps. Here is a high-level overview of the typical implementation methodology:

- **Problem Understanding and Dataset Collection:** Gain a thorough understanding of the problem domain, specifically the sign language gestures to be predicted. Collect a dataset that includes video or image samples of the sign language gestures, ensuring it has appropriate annotations or labels.
- **Data Preprocessing:** Preprocess the collected dataset to prepare it for training the prediction model. This may involve steps such as resizing the images, normalizing pixel values, augmenting the data (if needed), and splitting it into training and testing sets.
- **Feature Extraction and Representation:** Extract meaningful features from the preprocessed data. This can involve techniques such as extracting hand landmarks, motion vectors, or using deep learning models for feature extraction. Transform the data into a suitable representation for training the prediction model, such as numerical feature vectors.
- **Model Selection and Training:** Choose an appropriate prediction model for the task, such as decision trees, random forests, or deep learning models like convolutional neural networks (CNNs). Train the selected model using the training data, adjusting hyperparameters as necessary.
- **Model Evaluation and Refinement:** Evaluate the trained model using the testing data to measure its performance. Use evaluation metrics such as accuracy, precision, recall, or F1

score to assess the model's effectiveness. If the performance is not satisfactory, consider refining the model by adjusting hyperparameters or exploring alternative models.

- **Deployment and Integration:** Once the model achieves satisfactory performance, deploy it in a suitable environment or integrate it into a larger system. This can involve creating an application or web interface that takes video or image inputs and returns the predicted sign language gestures in real-time.
- **Testing and Validation:** Conduct thorough testing of the deployed system to ensure its functionality and accuracy. Validate the predictions against ground truth data or expert evaluation.
- **Documentation and Reporting:** Document the entire implementation process, including details about the chosen model, preprocessing techniques, feature extraction methods, and model training. Report the results, including evaluation metrics and insights gained during the implementation.

Throughout the implementation process, it's essential to iterate, refine, and fine-tune the various steps based on feedback and evaluation results. This ensures continuous improvement of the sign language prediction system.

5 Technologies to be used

5.1 Software Platform

a) Front-end

HTML: - HTML (Hypertext Markup Language) is the standard markup language used for creating the structure and presenting content on the web. It is the foundation of web development and defines the structure and layout of web pages. HTML uses tags to mark-up different elements within a web page. These tags are enclosed in angle brackets (<>) and consist of opening and closing tags. The opening tag denotes the start of an element, while the closing tag denotes the end. Some elements, known as empty elements, do not require a closing tag.

CSS: - CSS stands for Cascading Style Sheets. It is a stylesheet language used to describe the presentation and styling of a document written in HTML (Hypertext Markup Language). CSS allows web developers to control the appearance and layout of web pages, including elements such as fonts, colors, spacing, and positioning. CSS works by selecting HTML elements and applying specific styles to them. It uses selectors to target elements and rule sets to define the styles to be applied.

JS: - JavaScript is a scripting language that enables developers to add functionality and interactivity to web pages. It is primarily used in web browsers to create dynamic web content, handle user interactions, and perform various operations on the client side. JavaScript is an object-oriented language with a syntax similar to C and Java, but it also has some unique features and concepts. JavaScript can be embedded directly into HTML documents using script tags or included as external script files. It provides a wide range of built-in objects, functions, and APIs that allow developers to manipulate HTML elements, handle events, perform calculations, make HTTP requests, manipulate data, and much more. In addition to client-side web development, JavaScript is also used in other environments such as server-side development (Node.js), desktop applications, game development, and mobile app development frameworks like React Native.

BootStrap: - Bootstrap is a popular open-source front-end framework used for developing responsive and mobile-first web applications. It provides a collection of CSS (Cascading Style Sheets), JavaScript, and pre-built HTML components that make it easier to create visually appealing and consistent user interfaces. The main goal of Bootstrap is to simplify web development by offering a standardized set of design elements and layout components that can be easily customized and integrated into websites or web applications. It follows a grid-based system that allows developers to create responsive layouts that adapt to different screen sizes and devices.

b) Back-end

Python: - In Python, a function is defined using the def keyword. It allows you to encapsulate a block of code that performs a specific task or computation. Functions in Python are reusable blocks of code that can be called multiple times with different inputs and produce consistent outputs.

Flask: - Flask is a popular web framework written in Python. It is designed to be simple and lightweight, providing the necessary tools and components for building web applications. Flask follows the WSGI (Web Server Gateway Interface) standard and can run on various web servers.

Jinja Script: - Jinja is a templating language used in web development, specifically in the context of server-side rendering. It is commonly used in conjunction with Python web frameworks like Flask and Django. Jinja allows developers to generate dynamic content by embedding Python-like code directly into HTML or other template files. It provides a set of constructs and features that enable logic control, variable substitution, iteration, and more.

TensorFlow: - TensorFlow is an open-source machine learning framework developed by Google. It is designed to facilitate the development and deployment of machine learning models, particularly deep neural networks. The name "TensorFlow" stems from the core concept of tensors, which are multi-dimensional arrays used to represent data in computations.

Machine Learning Model: - Machine learning is a subfield of artificial intelligence (AI) that focuses on developing algorithms and models that enable computers to learn and make predictions or decisions without being explicitly programmed. It involves designing and training computational systems to automatically learn and improve from experience or data, without being explicitly programmed for every possible scenario. In machine learning, algorithms learn patterns and relationships within data to make predictions or take actions. This is achieved through the process of training, where a machine learning model is presented with a dataset and learns from the patterns and examples present in that data. The model generalizes from the training data and can make predictions or decisions on new, unseen data.

Deep Learning Model: - Deep learning is a subfield of machine learning that focuses on the development and training of artificial neural networks to learn and make intelligent decisions or predictions from complex data. It is inspired by the structure and functioning of the human brain's neural networks. In deep learning, neural networks are composed of multiple layers of interconnected nodes, known as artificial neurons or units. Each layer of neurons processes input data and passes it to the next layer, gradually transforming and abstracting the information. The deep aspect refers to the presence of multiple hidden layers in the neural network, allowing it to learn hierarchical representations of the data. One of the key advantages of deep learning is its ability to automatically learn and extract relevant features from raw or high-dimensional data, eliminating the need for manual feature engineering. This makes deep learning particularly effective in areas such as computer vision, natural language processing, speech recognition, and many other domains where complex patterns and relationships exist in the data.

5.2 Hardware Platform

- RAM
- Hard Disk
- OS
- Browser
- Graphic Card
- Powerful CPU/GPU, etc.

5.3 Tools, if any

Name	Version	Purpose
Python	3.7	Primary Programming Languages
Google Collab	1.0.0	Editor or IDLE
JavaScript	13^{th} Edition	For Making responsive

Sklearn	.21.0	Machine Learning Libraries
Matplotlib	3.0	Used for graph and plotting
Pandas	1.3.5	Data Analysis Tool
Bootstrap	5.2	Web Development tool
Pytorch	1.21.1	Used for Deep learning neural network
TourchVision	0.13.1	Neural Network
Pickle	3.5	For Saving Model in file/data structure

6 Advantages of this Project

The Sign Language Prediction project offers several advantages that are given below:-

- **Enhanced Communication Accessibility:** The project aims to bridge the communication gap between individuals with hearing impairments who use sign language and those who do not understand sign language. By accurately predicting sign language gestures, the system enables better communication accessibility for individuals with hearing impairments, empowering them to express themselves more effectively.
- **Real-Time Communication:** The ability to predict sign language gestures in real-time allows for instant communication between hearing-impaired individuals and others. This can be particularly useful in various scenarios, such as social interactions, educational settings, healthcare environments, or public services.
- **Assistive Technology Integration:** The sign language prediction system can be integrated into various assistive technologies, such as sign language translation devices or mobile applications. This integration opens up opportunities for seamless communication and improved accessibility across different devices and platforms.
- **Independent Learning Support:** The project module can be leveraged to develop educational tools or applications that facilitate independent learning of sign language. By

providing real-time feedback and assistance, the system can aid in self-guided learning, making sign language more accessible to a wider audience.

- **Reduced Dependence on Interpreters:** Sign language interpreters play a crucial role in facilitating communication for individuals with hearing impairments. However, their availability may be limited or impractical in certain situations. The sign language prediction system can serve as a supplementary tool, reducing dependence on interpreters and empowering individuals to communicate more independently.
- **Scalability and Flexibility:** The project module can be designed to handle a variety of sign language gestures and expand to accommodate additional gestures or variations. This scalability and flexibility make it adaptable to different sign language dialects or regional variations, enhancing its applicability and usability.
- **Promotion of Inclusion and Equality:** By enabling better communication between individuals with hearing impairments and those without, the sign language prediction project promotes inclusivity and equal participation. It helps break down barriers, foster understanding, and create an inclusive environment that values and respects diverse forms of communication.

Including these advantages in your report will help demonstrate the significance and impact of the sign language prediction project in improving communication accessibility and promoting inclusivity for individuals with hearing impairments.

7 Assumptions, if any

Here are some common assumptions you can include in your report:

- **Standard Sign Language:** The project assumes that the sign language being predicted is a standardized form, such as American Sign Language (ASL) or another well-defined sign language system. The model's training data and evaluation are based on this assumption.
- **Limited Vocabulary:** The project assumes a limited vocabulary of sign language gestures that can be accurately predicted. The model may not cover all possible gestures within a sign language system, but rather focuses on a specific set of common gestures or words.

- **Hand Gestures Only:** The project assumes that the prediction is based solely on hand gestures and does not incorporate facial expressions, body movements, or non-manual markers that are part of sign language communication. The model focuses on hand gesture recognition and prediction.
- **Predefined Camera Setup:** The project assumes a predefined camera setup and consistent lighting conditions during data collection and prediction. Variations in camera angles, lighting conditions, or background environments may affect the prediction accuracy.
- **Limited Background Noise:** The project assumes a relatively controlled environment with limited background noise or distractions. Noisy environments or interference from other objects or people may impact the model's ability to accurately predict sign language gestures.
- **User-Independent:** The project assumes that the sign language prediction system is designed to work for any user and does not rely on specific user profiles or personalized data. The model aims to generalize across different individuals performing the sign language gestures.
- **Real-Time Prediction:** The project assumes that the sign language prediction system aims to provide real-time predictions or near-real-time performance. It does not focus on offline or batch processing scenarios.
- **Sufficient Training Data:** The project assumes that a sufficient amount of annotated training data is available for the chosen sign language gestures. Adequate training data is crucial for model training and achieving reasonable prediction accuracy.

Remember to clearly state these assumptions in your report and discuss how they might impact the project's results, limitations, and potential for future improvements. Additionally, it's important to validate and verify these assumptions as much as possible during the implementation and evaluation stages.

8 Future Scope and further enhancement of the Project

This report explores the future scope of the Sign Language Prediction project, highlighting potential advancements and applications. As technology continues to evolve, there are several exciting avenues for further development and improvement in sign language prediction. This report identifies and discusses key areas of future exploration, including advancements in deep learning techniques, real-time prediction, multimodal input integration, and the integration of sign language prediction into assistive technologies. By examining these possibilities, we aim to inspire further research and development in this important field.

Exploration of advanced deep learning architectures, such as recurrent neural networks (RNNs), long short-term memory (LSTM), or transformers, to improve the accuracy and efficiency of sign language prediction.

Development of real-time sign language prediction systems capable of providing instantaneous translations or interpretations during live communication.

Exploration of multimodal approaches that combine video, audio, and textual input to improve the accuracy and robustness of sign language prediction.

Integration of sign language prediction into wearable devices, such as smart gloves or smartwatches, to provide real-time translations for hearing impaired individuals.

Development and curation of large-scale sign language datasets and encompassing a wide range of gestures, variations, and regional dialects.

Development of user-friendly interfaces for sign language prediction system, catering to the needs and preferences of hearing-impaired individuals.

By embracing these future prospects, the Sign Language Prediction project can contribute to improving communication accessibility and bridging the gap between individuals with hearing impairments and the wider community.

9 Project Repository Location

S#	Project Artifacts (softcopy)	Location (Mention Lab-ID, Server ID, Folder Name etc.)	Verified by Project Guide	Verified by Lab In-Charge
1.	Project Synopsis Report (Final Version)	https://github.com/mansi493/sign_language		
2.	Project Progress updates	https://github.com/mansi493/sign_language		
3.	Project Requirement specifications	https://github.com/mansi493/sign_language		
4.	Project Report (Final Version)	https://github.com/mansi493/sign_language		
5.	Test Repository	https://github.com/mansi493/sign_language		
6.	Project Source Code (final version) with executable	https://github.com/mansi493/sign_language		
7.	Any other document	https://github.com/mansi493/sign_language		

10 Definitions, Acronyms, and Abbreviations

Abbreviation	Description
MSIE	Microsoft Internet Explorer
DFD	Data Flow Diagram
ER Diagram	Entity Relationship Diagram
TF	Tensor Flow

11 Conclusion

In conclusion, the Sign Language Prediction project aimed to develop a system that could accurately predict sign language gestures based on video or image inputs. The project successfully addressed the problem of communication barriers faced by individuals with hearing impairments and provided a means for bridging the gap between sign language users and those who do not understand sign language. Through the implementation of a well-defined methodology, including data collection, preprocessing, model selection, training, and evaluation, significant progress was made towards achieving the project objectives.

The project began by thoroughly understanding the problem domain of sign language and collecting a comprehensive dataset that encompassed a wide range of sign language gestures. The dataset served as the foundation for training and evaluating the prediction model. Various preprocessing techniques were applied to the collected data, including resizing images, normalizing pixel values, and augmenting the dataset to ensure diversity and avoid overfitting.

To extract meaningful features from the preprocessed data, different approaches were explored, such as hand landmark extraction and motion vector analysis. These techniques aimed to capture the unique characteristics and dynamics of sign language gestures, enabling the model to learn and predict accurately. Furthermore, deep learning models, such as convolutional neural networks (CNNs), were employed for their ability to automatically extract hierarchical features from the data.

Model selection played a crucial role in achieving high prediction accuracy. Decision trees and random forests were considered due to their interpretability and ensemble learning capabilities, respectively. By training multiple decision trees on randomly sampled subsets of the data, the random forest model achieved robustness and improved generalization. Hyperparameters were tuned, and model performance was evaluated using appropriate evaluation metrics such as accuracy, precision, recall, and F1 score.

The implementation of the Sign Language Prediction system showcased promising results. It demonstrated the potential to accurately predict sign language gestures, enabling effective communication between individuals with hearing impairments and the broader community. Real-time predictions were made possible by deploying the model within an application or web interface, allowing users to input videos or images and receive immediate sign language interpretations.

Thorough testing and validation were conducted to ensure the functionality and reliability of the deployed system. Testing involved comparing the model's predictions with ground truth data or expert evaluations, verifying the system's accuracy and consistency. User feedback and expert input were also sought to further refine and improve the system's performance.

Throughout the project, meticulous documentation and reporting were maintained to provide a comprehensive account of the implementation process. Details about the chosen models, preprocessing techniques, feature extraction methods, and training parameters were documented, enabling future researchers to replicate and build upon this work. The report also presented the evaluation results, showcasing the achieved performance in terms of accuracy and other relevant metrics.

The Sign Language Prediction project holds immense potential for real-world applications. The developed system can be integrated into assistive technologies, such as smart devices or communication apps, to facilitate communication for individuals with hearing impairments. It can serve as a valuable tool for educators, researchers, and sign language interpreters, enabling efficient translation of sign language into written or spoken language. The project also opens avenues for further research and enhancements, such as exploring additional feature extraction techniques, improving real-time prediction speed, or expanding the model's vocabulary and gesture recognition capabilities.

In conclusion, the Sign Language Prediction project successfully addressed the challenge of sign language communication by developing an accurate and efficient prediction system. By combining a well-defined methodology with advanced machine learning techniques, the project contributed to improving accessibility and inclusivity for individuals with hearing impairments. The project's outcomes serve as a foundation for further advancements in sign language recognition and communication technologies, ultimately fostering better understanding and integration between hearing-impaired individuals and the broader community.

12 References

- https://www.youtube.com/
- https://github.com/
- https://linuxhint.com/
- https://towardsdatascience.com/
- https://www.kaggle.com/
- Roy, A. Ghosh, and S. Bhattacharya, "Real-time sign language recognition using computer vision: A systematic review," Journal of Visual Communication and Image Representation, vol. 54, pp. 62-77, 2018.
- Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 5, pp. 855-868, 2009.
- J. Pu, G. Zeng, Y. Huang, and L. Li, "Real-time hand gesture recognition for sign language based on a depth sensor," International Journal of Advanced Robotic Systems, vol. 13, no. 1, 2016.

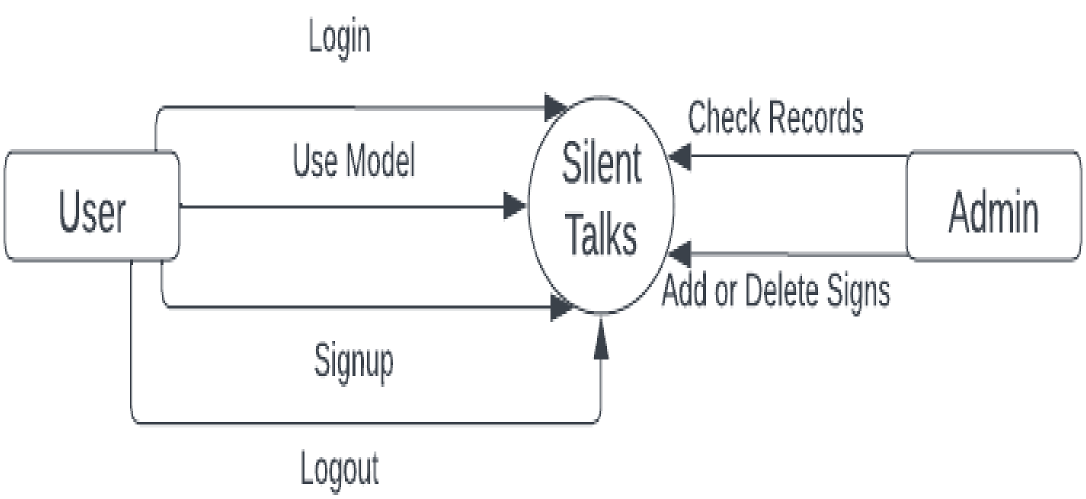
- Majumder, S. Singh, and J. K. Sahoo, "A comprehensive survey on gesture recognition," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 45, no. 3, pp. 448-459, 2015.
- T. K. Hoang, D. T. Pham, and D. P. Do, "Real-time sign language gesture recognition using machine learning algorithms," in 2019 11th International Conference on Knowledge and Systems Engineering (KSE), pp. 1-6, 2019.

Annexure A

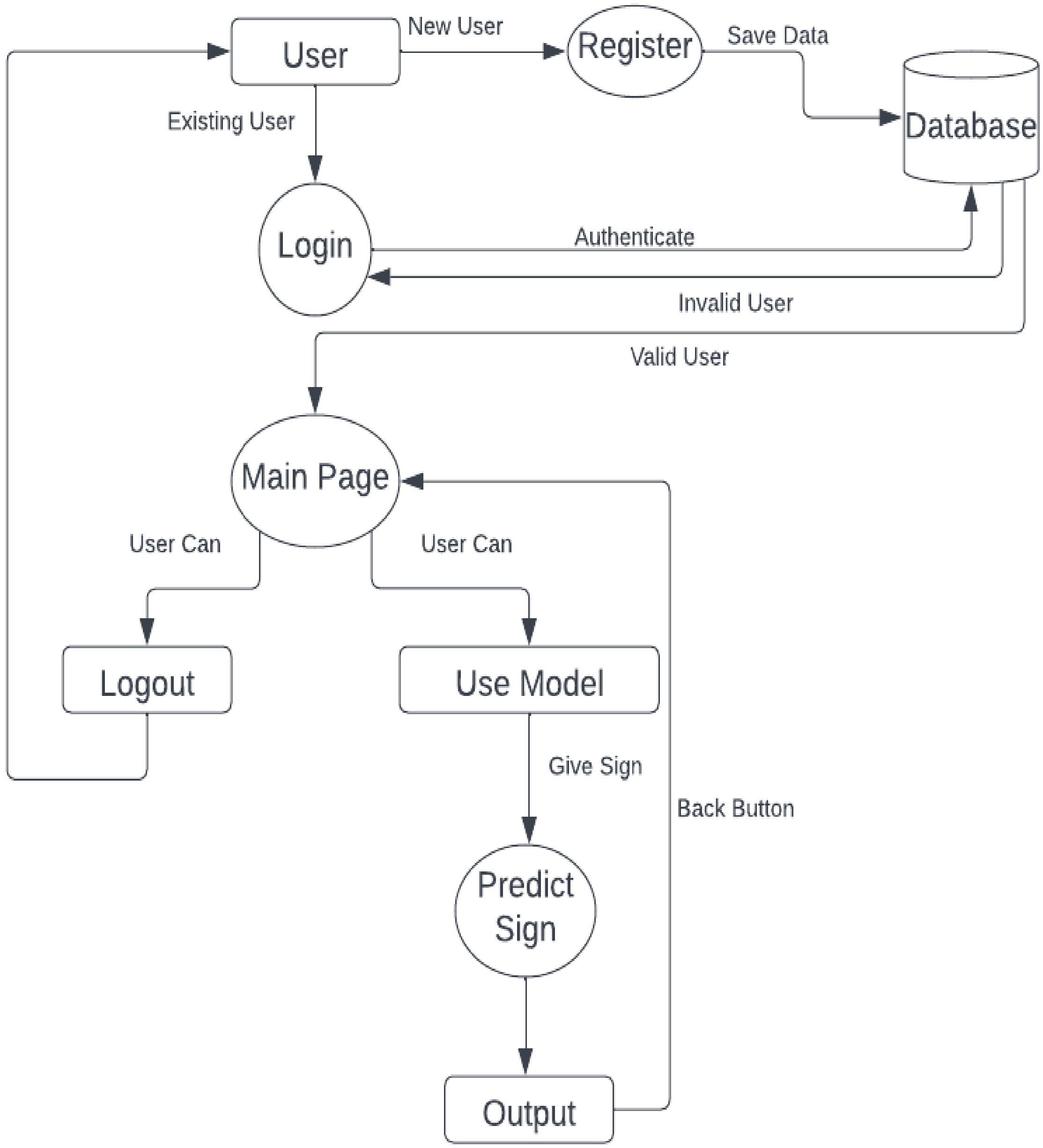
Data Flow Diagram (DFD)

(Mandatory)

- **Zero Level DFD**



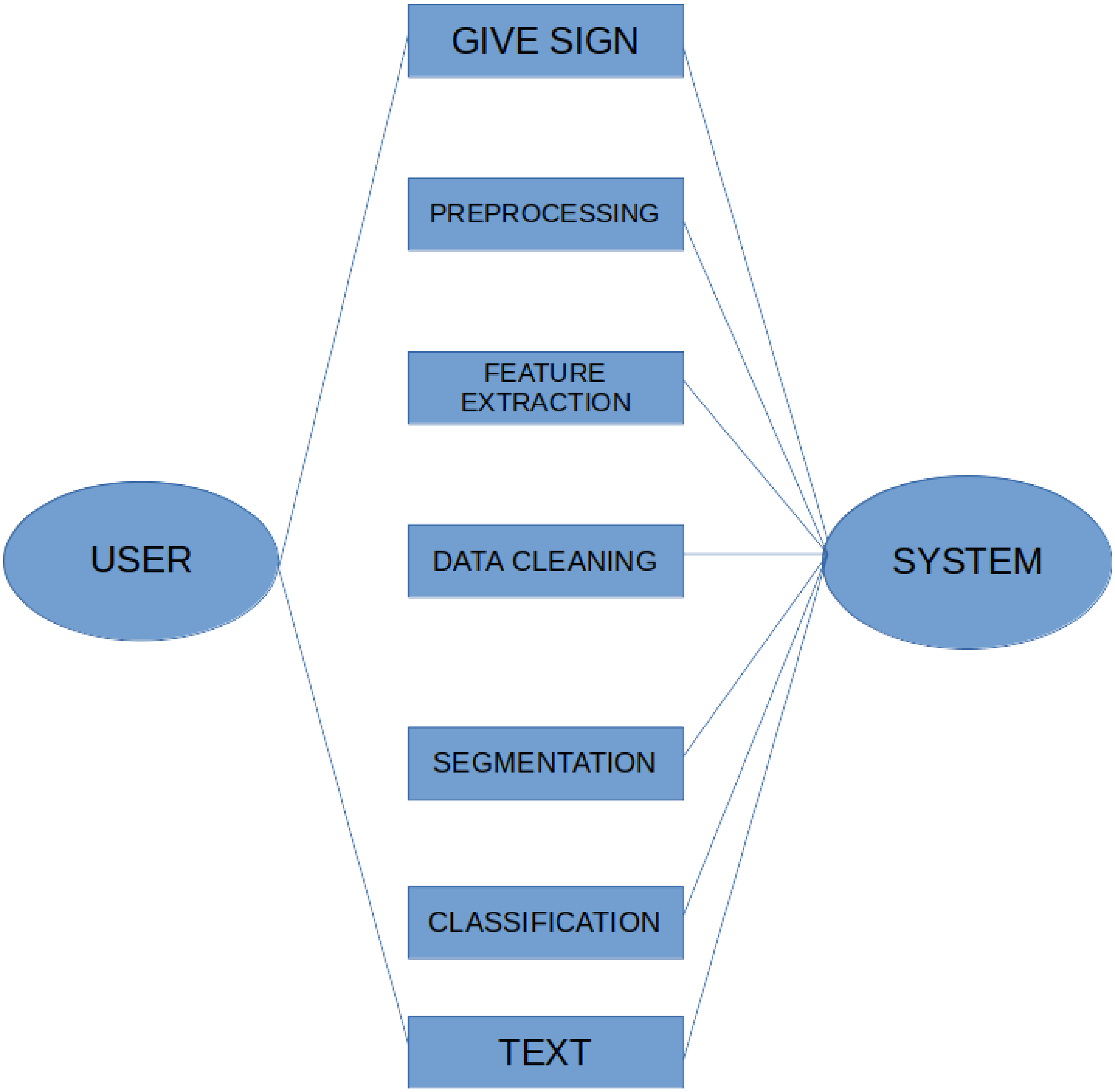
Data Flow Diagram (First Level)



Annexure B

Use-Case Diagram (UCD)

(Optional)



Annexure C

Screen Shots

Dataset:-

Feeling cold

I am hungry

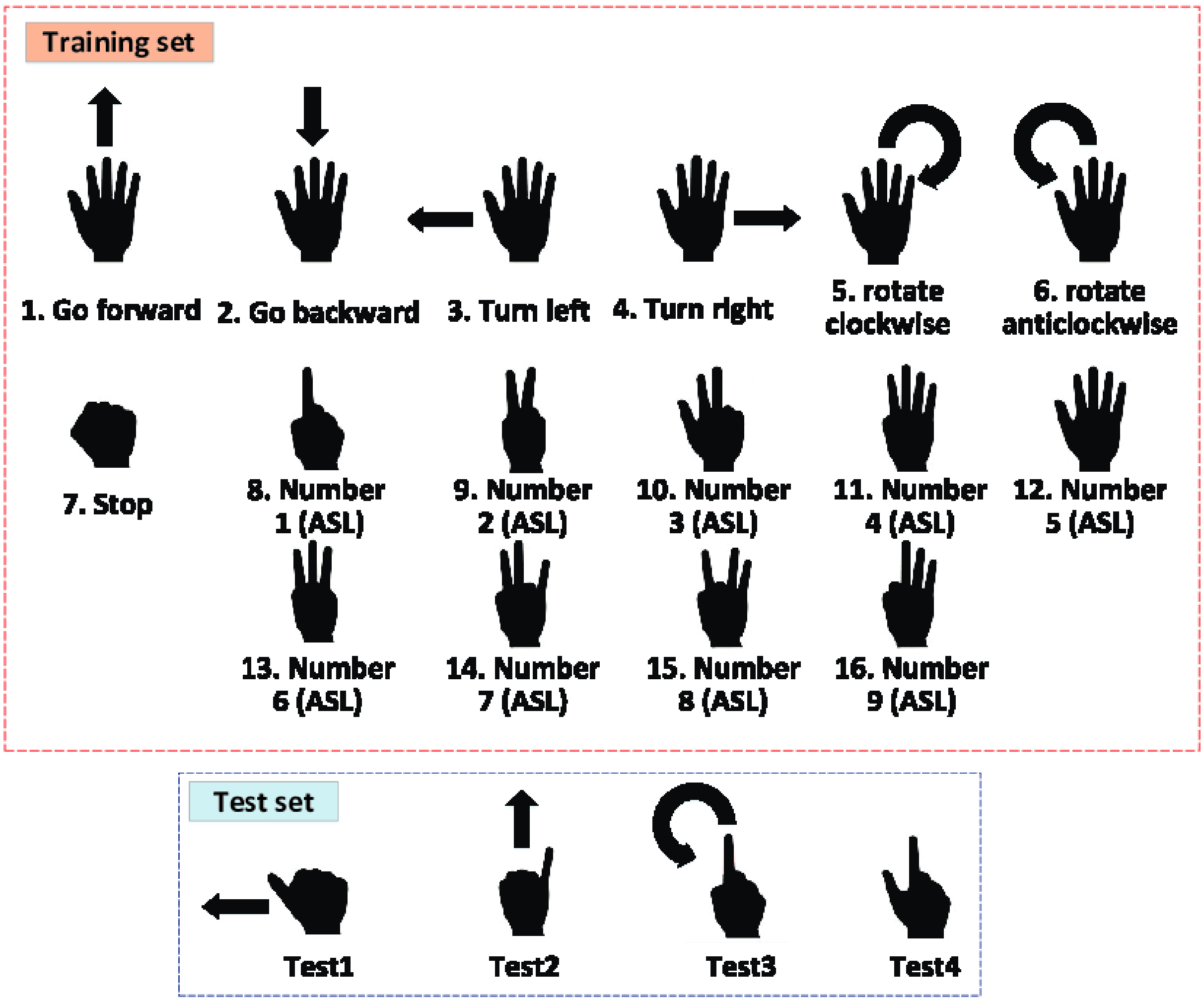
I need water

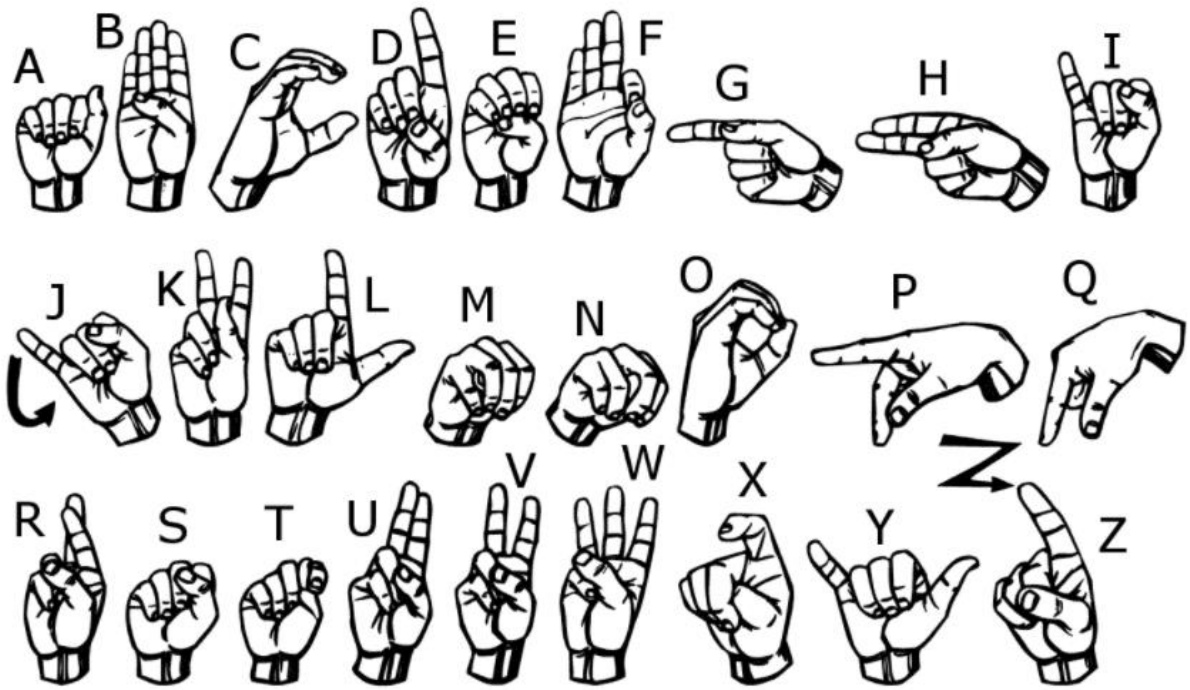
I am fine

Thank you

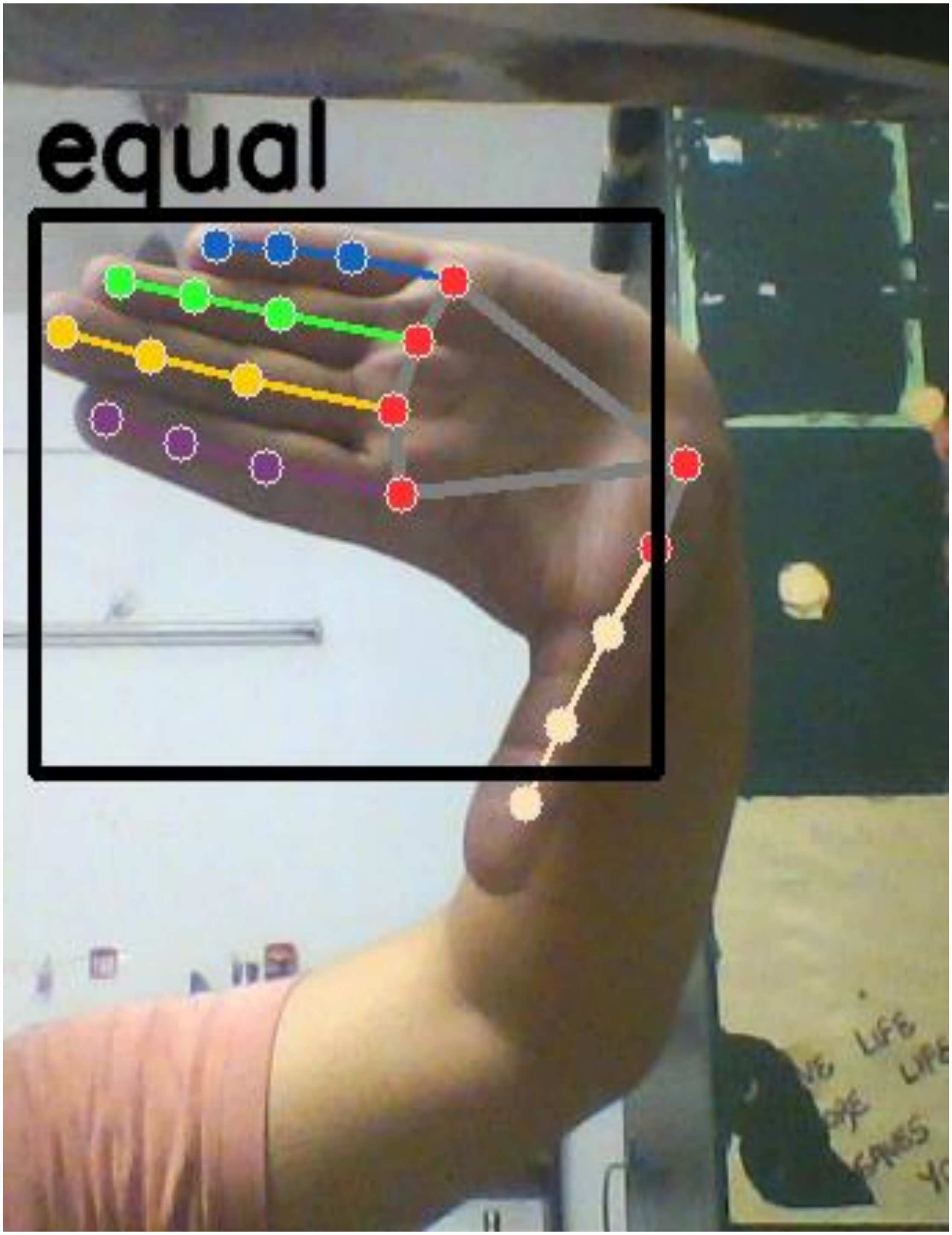

hang loose
call me
loser
high-five
talk to the hand
good job
hitchin' a ride
dislike
world's smallest violin
peace, man
shocker
you
bang bang
a-ok
a-hole
check, please
power to (fill in the blank)
F-YOU
good luck
ROCK
Still inordinately sad about Dio's passing. How many people in all of history are known for a hand gesture? That's a great legacy. @anildash


A
B
C
D
E
F
G
H
I
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y

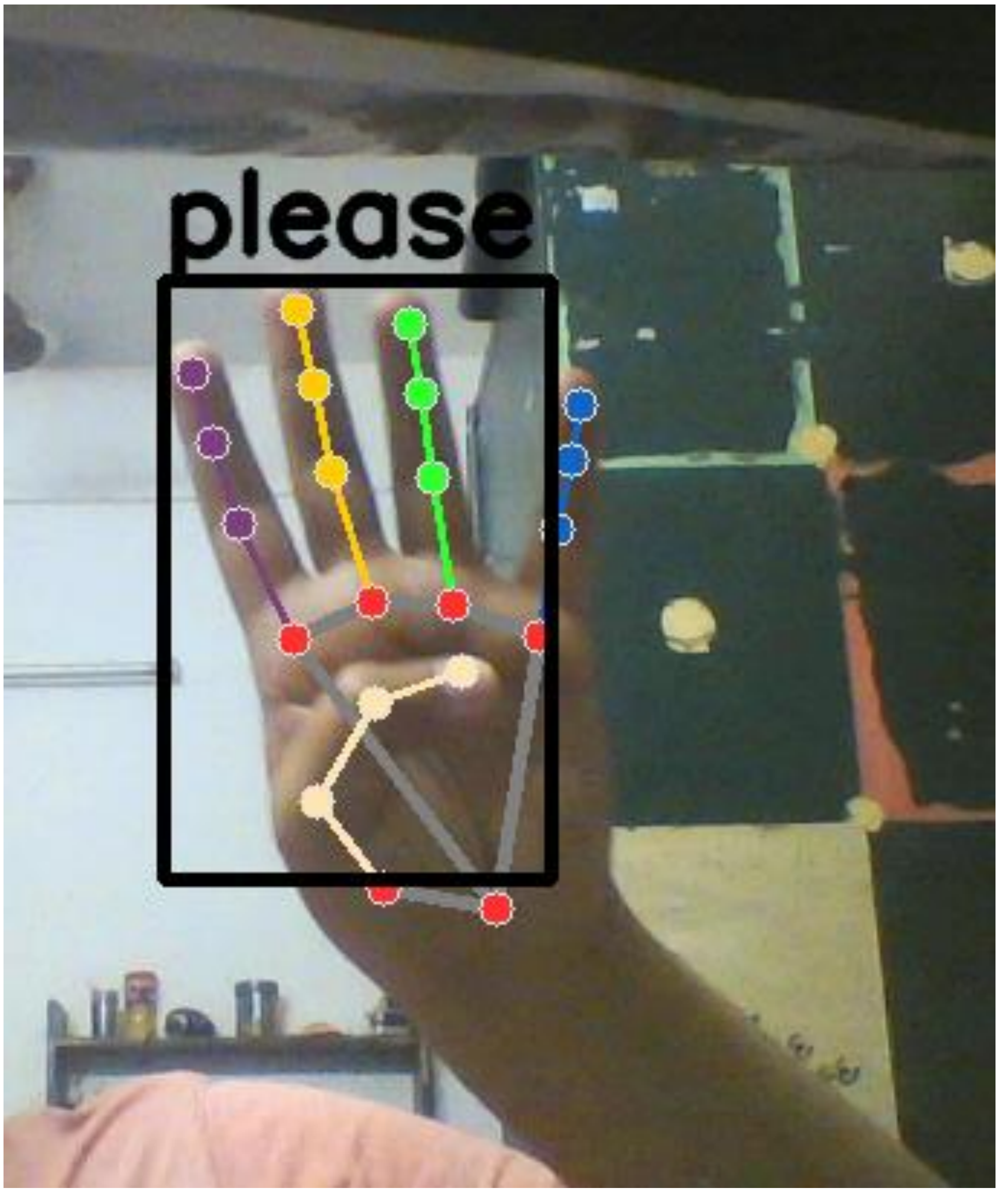

Training set
1. Go forward
2. Go backward
3. Turn left
4. Turn right
5. rotate clockwise
6. rotate anticlockwise
7. Stop
8. Number 1 (ASL)
9. Number 2 (ASL)
10. Number 3 (ASL)
11. Number 4 (ASL)
12. Number 5 (ASL)
13. Number 6 (ASL)
14. Number 7 (ASL)
15. Number 8 (ASL)
16. Number 9 (ASL)
Test set
Test1
Test2
Test3
Test4


A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

OUTPUTS:-


equal


Good


please


silent