A PRELIMENERY REPORT ON

Sign language recognition using deep learning

SUBMITTED TO AN EMPOWERED AUTONOMOUS INSTITUTE, AFFILIATED SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF TECHNOLOGY

(ARTIFICIAL INTELLIGENCE)

SUBMITTED BY

ADITI KANNAWAR	Exam No: 2020AAIT1101028
PRANALI PATIL	Exam No: 2020AAIT1101054
RANI MANWAR	Exam No: 2020AAIT1101039
TANAY MAPARE	Exam No: 2020AAIT1101040

GHraisoni COLLEGE Engineering and Management Pune

DEPARTMENT OF ARTIFICIAL INTELLIGENCE

G H RAISONI COLLEGE OF ENGINEERING AND MANAGEMENT

WAGHOLI, PUNE 412207

AN EMPOWERED AUTONOMOUS INSTITUTE,

AFFILIATED TO SAVITRIBAI PHULE PUNE

UNIVERSITY

2023-24

CERTIFICATE

This is to certify that the project report entitles

" Sign language recognition using deep learning"

Submitted by

ADITI KANNAWAR	Exam No: 2020AAIT1101028
PRANALI PATIL	Exam No: 2020AAIT1101054
RANI MANWAR	Exam No: 2020AAIT1101039
TANAY MAPARE	Exam No: 2020AAIT1101040

are bonafide student of this institute and the work has been carried out by them under the supervision of **Prof. Sonali sonavane** and it is approved for the partial fulfilment of the requirement of An Autonomous Institute, Affiliated to Savitribai Phule Pune University, for the award of the degree of **Bachelor of Technology** (Artificial Intelligence).

(Prof. Sonali Sonavane)
Project Guide
Department of Artificial Intelligence

(Prof. R.Y. Sable)
H.O.D.
Department of Artificial Intelligence

(**External Examiner)**

(Dr. R. D. Kharadkar)
Director
GHRCEM, Pune

Place: Pune

Date:

ACKNOWLEDGMENT

It is our privilege to acknowledge with deep sense of gratitude towards our project guide, **Prof. Sonali Sonavane** for her valuable suggestions and guidance throughout course of study and timely help given in the progress of my dissertation on "**SIGN LANGUAGE RECOGNITION SYSTEM USING DEEP LEARNING**".

It is needed a great moment of immense satisfaction to express out profound gratitude towards our H.O.D **Prof. Rachana Sable**, whose real enthusiasm was a source of inspiration for our work. Our special thanks to **Dr. R. D. Kharadkar**, Director, G H Raisoni college of Engineering and Management, Pune for his valuable support.

We would also like to thank all other faculty members of Artificial Intelligence Department who directly or indirectly kept the enthusiasm and momentum required to keep the work towards an effective dissertation alive in us and guided in their own capacities in all possible ways.

Aditi Kannawar(04)
Rani Manwar (47)
Pranali patil (41)
Tanay Mapare (75)

ABSTRACT

The predominant means of communication is speech; however, there are persons whose speaking or hearing abilities are impaired. Communication presents a significant barrier for persons with such disabilities. The use of deep learning methods can help to reduce communication barriers. This paper proposes a deep learning-based model that detects and recognizes the words from a person's gestures. Deep learning models, namely, LSTM and GRU (feedback-based learning models), are used to recognize signs from isolated Indian Sign Language (ISL) video frames. The four different sequential combinations of LSTM and GRU (as there are two layers of LSTM and two layers of GRU) were used with our own dataset, IISL2020. The proposed model, consisting of a single layer of LSTM followed by GRU, achieves around 97% accuracy over 11 different signs. This method may help persons who are unaware of sign language to communicate with persons whose speech or hearing is impaired.

LIST OF FIGURES

FIGURE NO	ILLUSTRATION	PAGE NO
1.1	Methodology	5
3.1	Analysis model	27
4.1	System Architecture	30
4.2	Sign language recognition Flow Diagram	32
4.3	Block diagram of software	33
6.1	Project implementation	39
7.1	Data collection	46
7.2	Data model training using media pipe	47
7.3	Sign/Gesture detection	48

LIST OF ABBREVIATIONS

ABBREVIATION	ILLUSTRATION
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
GPU's	Graphics Processing Units

TABLE OF CONTENTS

CONTENT

CONTENT	Page No.
CERTIFICATE	**II**
ACKNOWLEDGMENT	**III**
ABSTRACT	**IV**
LIST OF FIGURES	**V**
LIST OF TABLES	**VI**
LIST OF ABBREVIATIONS	**VII**

Sr.No.	Title of Chapter	Page No.
01	**Introduction**	01
1.1	Overview	01
1.2	Motivation	02
1.3	Problem Definition	03
1.4	Project Scope & Limitations	03
1.5	Methodologies of Problem solving	05
02	**Literature Survey**	08
03	**Software Requirements Specification**	13
3.1	Assumptions and Dependencies	13
3.2	Functional Requirements	15
	3.2.1 System Feature 1(Functional Requirement)	16
	3.2.2 System Feature2 (Functional Requirement)	16
	3.2.3 Hardware Interfaces	16
	3.2.4 Software Interfaces	17
3.3	Nonfunctional Requirements	18
	3.3.1 Performance Requirements	19
	3.3.2 Safety Requirements	21
	3.3.3 Software Quality Attributes	23
3.4	System Requirements	24
	3.4.1 Software Requirements (Platform Choice)	24
	3.4.2 Hardware Requirements	24
	3.4.3 Software quality attributes	25
	3.4.4 Application	25
3.5	Analysis Models	27
04	**System Design**	30
4.1	System Architecture	30
4.2	Data Flow Diagrams	32

4.3	Block Diagram For Software	33
05	**Project Plan**	34
5.1	Project Resources	34
5.2	Risk Management	35
5.2.1	Risk Identification	35
5.2.2	Risk Analysis	36
5.2.3	Overview of Risk Mitigation, Monitoring, Management	37
5.3	Project Task Set	38
06	**Project Implementation**	39
6.1	Tools and Technologies Used	39
6.2	Algorithm Details	40
07	**Result**	46
7.1	Data collection	46
7.2	Data model training using Media pipe	47
7.3	Sign Detection	48
08	**Conclusions**	50
8.1	Conclusions	50
8.2	Future Work	51
09	**References**	54

1. INTRODUCTION

There are various ways of communication or expression, but the predominant mode of human communication is speech; when it is hindered, people must use a tactile-kinesthetics mode of communication instead. In India, the overall percentage of persons with this disability in the population was 2.2 percent from July 2018 to December 2018, according to the National Statistical Office survey report 76th round of the National Sample Survey (NSS) . Sign language is one of the greatest adaptations for persons with speech and hearing impairments. It is also known as a visual language. Generally, it has five basic parameters: hand shape, orientation, movement, location, and components such as mouth shape and eyebrow movements. Research has been conducted on voice generation using smart gloves, which could give a voice to sign language movements. However, those who do not know sign language usually undervalue or reject persons with such an impairment because of the lack of proper communication between them. Hence, this paper proposes a system aimed at removing the communication gap and giving every individual a fair and equal chance. It involves taking a video of the person making hand gestures, processing it, and passing it to the proposed model, which predicts words one by one. The system then generates a meaningful sentence out of those words that can then be converted into the language selected by the communicator.

1.1 OVERVIEW

The project on "Sign Language Recognition using Deep Learning" involves the development of a deep learning-based model that detects and recognizes sign language gestures, converting them into English alphabet characters. The project utilizes Convolutional Neural Networks (CNN) for image classification tasks, specifically focusing on converting sign language gestures to text. The evaluation metric used is accuracy, measuring the ratio of correctly classified samples to the total number of samples. The project includes an end-to-end pipeline for model building, hyperparameter tuning, and deployment, emphasizing the importance of normalizing input data and batching for efficient training. Overall, the project aims to bridge the communication gap between individuals with hearing impairments and those who use sign language by enabling accurate and real-time recognition of sign language gestures through deep learning techniques. This project focuses on the development of a deep learning-based sign language recognition system that can accurately translate sign language gestures into text or speech, with the aim of improving communication and accessibility for individuals with hearing impairments.

1.2 MOTIVATION OF THE PROJECT

The primary motivation behind this project is to enable effective communication and bridge the gap between individuals with hearing impairments and those who do not know sign language. The project aims to develop a deep learning-based system that can accurately detect and recognize sign language gestures, converting them into text or speech. This would greatly improve accessibility and inclusivity for the deaf and hard-of-hearing community. The researchers were motivated by the feeling of working on a project that could have a positive impact on people's lives and make a difference in the field of accessibility. Previous approaches to sign language recognition have been limited, either relying on contact-based systems like sensor gloves or vision-based systems with a single camera, which have limitations in capturing the full range of sign language gestures. The advent of deep learning techniques and the availability of powerful computational resources have made it more feasible to tackle the complex problem of sign language recognition, which was previously challenging. The researchers saw an opportunity to leverage the advancements in deep learning, computer vision, and sensor technology to develop a more robust and practical sign language recognition system that could be deployed in real-world scenarios. In summary, the primary motivation behind this project is to improve communication and accessibility for individuals with hearing impairments by developing an accurate and reliable sign language recognition system using deep learning techniques.

1.3 PROBLEM DEFINITION

The project aims to develop a deep learning-based sign language recognition system to facilitate communication between individuals with hearing impairments and those unfamiliar with sign language. By accurately detecting and translating sign language gestures into text or speech, the system seeks to overcome communication barriers, enhance inclusivity, and improve accessibility for the deaf and hard-of-hearing community. Through real-time recognition and deployment in various settings, the project aims to empower individuals with hearing impairments, advance assistive technology, and contribute to a more inclusive society.

1.4 PROJECT SCOPE AND LIMITATIONS

1.4.1 PROJECT SCOPE

The primary objective of the project is to develop a deep learning-based model that can accurately detect and recognize sign language gestures, converting them into text or speech to facilitate communication between individuals with hearing impairments and those who do not know sign language. The project involves the use of computer vision and deep learning techniques, such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTMs), to process and classify sign language gestures from visual input (images or video). The project utilizes the Sign Language MNIST dataset, which contains grayscale images of handwritten digits representing the letters of the alphabet, as the primary dataset for training and evaluating the deep learning models. The project aims to address the limitations of previous sign language recognition approaches by leveraging advancements in deep learning and computer vision, as well as techniques like hand tracking using Media Pipe, to improve the accuracy and robustness of the sign language recognition system. The project scope includes the development of a real-time sign language recognition system that can be deployed in various settings, such as educational institutions, workplaces, and public spaces, to enhance communication and accessibility for individuals with hearing impairments. The project also explores the potential for expanding the scope to include recognition of more complex sign language gestures, facial expressions, and body postures, as well as the translation of sign language to text or speech in multiple languages.

1.4.2 LIMITATIONS

1. **Generalization Capability:** The project faced challenges with the generalization capability of the models, leading to difficulties in accurately classifying certain signs. This limitation indicates that the models may struggle to consistently recognize a wide range of sign language gestures, impacting the overall performance and reliability of the system.

2. **Ambiguity in Gesture Recognition:** The system exhibited confusion between several signs, such as distinguishing between the signs for "U" and "W." This ambiguity in gesture recognition highlights a limitation in the model's ability to differentiate between similar signs accurately, potentially leading to errors in translation and communication.

3. **Performance Issues:** While the project demonstrated potential in real-time sign language recognition, the model's performance was not consistently optimal. The need for multiple attempts to achieve a good demo video suggests that the system may encounter performance issues, affecting its usability and effectiveness in practical applications.

4. **Need for Improvement:** The project acknowledged the necessity for further analysis and enhancements to the system. Recommendations for improvement include collecting more high-quality data, exploring different convolutional neural network architectures, and redesigning the vision system. These improvements are crucial to address the identified limitations and enhance the system's accuracy and usability in real-world scenarios

In summary, the detailed limitations of the "Sign Language Recognition using Deep Learning" project include challenges related to generalization capability, ambiguity in gesture recognition, performance issues, and the need for ongoing improvements to enhance the system's accuracy and usability in practical applications.

1.5 METHODOLOGIES OF PROBLEM SOLVING

The key methodologies employed in the "Sign Language Recognition using Deep Learning" project include the use of deep learning models (LSTM, GRU, and CNN), computer vision techniques (hand tracking and dual-camera setup), dataset preparation, real-time recognition, and an iterative approach to improve the system's performance and accuracy.

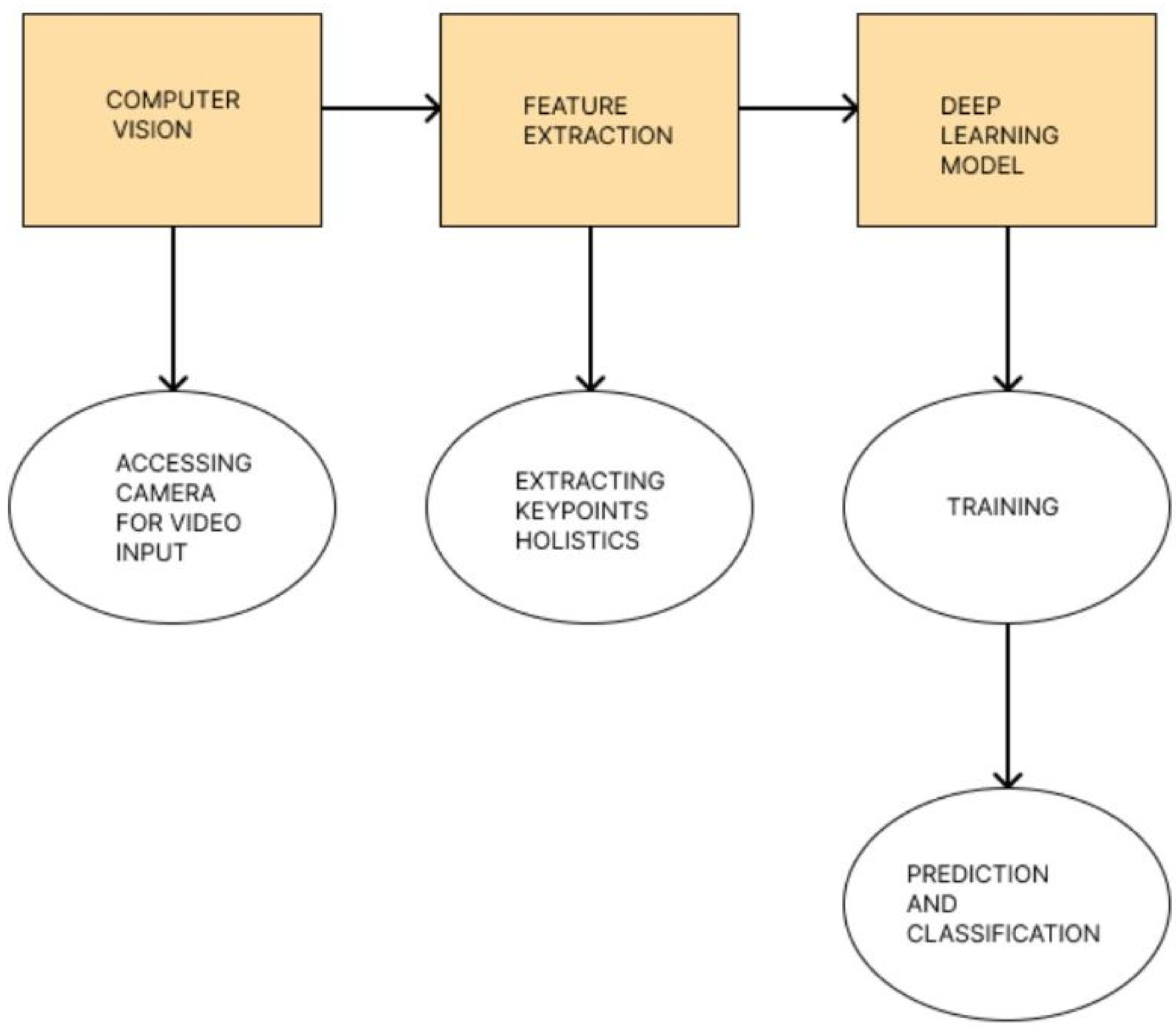

Fig. 1.1 Methodology

Deep Learning Models:

The project utilized deep learning models, specifically Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), for recognizing signs from isolated sign language video frames.

Convolutional Neural Networks (CNNs) were also employed to convert sign language gestures into the corresponding English alphabet characters.

Computer Vision Techniques:

The project leveraged computer vision techniques, such as hand tracking using MediaPipe, to improve the accuracy and robustness of the sign language recognition system.

The use of a dual-camera setup, with a head-mounted camera and a chest-mounted camera, provided different perspectives (top-view and bottom-view) to address the ambiguity in sign language gesture recognition.

Dataset Preparation:

The project utilized the Sign Language MNIST dataset, which contains grayscale images of handwritten digits representing the letters of the alphabet, for training and evaluating the deep learning models.

The researchers also explored the use of a custom dataset of 24 static signs from the Panamanian Manual Alphabet to develop the initial prototype.

Real-Time Recognition:

The project aimed to develop a real-time sign language recognition system that could be deployed in various settings, such as educational institutions, workplaces, and public spaces, to enhance communication and accessibility for individuals with hearing impairments.

The researchers tested the system's performance in real-time scenarios, using a chest-mounted camera and running the bottom-view model on a laptop.

Iterative Improvement:

The project acknowledged the need for further analysis and enhancements to the system, including collecting more quality data, exploring different CNN architectures, and redesigning the vision system to address the identified limitations.

The researchers suggested that an orthography corrector or a word predictor could be integrated to improve the overall performance and accuracy of the sign language recognition system.

2. LITERATURE SURVEY

[2.1] Sign Language Recognition in the Hindi Language Based on Computer Vision [1]

The paper proposes a system using CNNs and RNNs to recognize Hindi sign language, aiding speech-impaired communication. Challenges include dataset size and real-time recognition. Goals: expand dataset, enhance accuracy, recognize words, and translate to Hindi.

[2.2] Hindi Sign Language Detection Using CNN [2]

The paper proposes CNN-based Hindi Sign Language detection. It includes data collection, OpenCV sign detection, preprocessing, CNN training, and evaluation. Goals: aiding communication for the deaf, addressing accuracy concerns, and enhancing real-time implementation

[2.3] Indian Sign Language Recognition using Convolutional Neural Network [3]

Paper on Indian Sign Language Recognition with CNNs: Covers image acquisition, segmentation, and CNN training. Aims to enhance communication, accessibility, and autonomy. Challenges: cultural sign variations, image name prediction. Future: improve image processing, enable bidirectional communication, real-time interpretation.

[2.4] Sign Language Recognition using Convolutional Neural Network with Customization[4]

Sign Language Recognition: Includes data collection, preprocessing, CNN building, GUI, real-time accuracy evaluation. Emphasizes real-time processing, feature extraction. Challenges include incremental learning for accuracy. Future: mitigate accuracy decline, explore alternative learning, advance computer vision.

[2.5] Gesture Recognition Based on a Convolutional Neural Network–Bidirectional Long Short-Term Memory Network for a Wearable Wrist Sensor with Multi-Walled Carbon Nanotube/Cotton Fabric Material [5]

Gesture Recognition using CNN-BiLSTM: Involves fabricating sensors, integrating them into a wristband, collecting gestures, and testing accuracy. Utilizes MWCNT/CF composite for flexible, sensitive pressure sensors. Challenges include limited gesture coverage and real-world applicability. Aims for advancements in HCI, health monitoring, assistive tech, smart textiles, and industrial applications.

[2.6] Sign Language Recognition System Using Deep Neural Network [6]

Paper on Sign Language Recognition System: Utilizes 2-layer CNN for classification using Python. Identifies six sign languages, achieves high accuracy, and enables device control. Lacks discussion on CNN implementation challenges. Future: enhance performance, video recognition, integrate with device control.

[2.7] A Static Hand Gesture Based Sign Language Recognition System using Convolutional Neural Networks [7]

Hand Gesture Recognition: Involves hand segmentation, dataset creation, CNN design, training/testing, live testing, and evaluation. Achieves high accuracy, real-time output, and offers a customizable dataset. Challenges include lighting, static gestures, computational time, background, and Indian Sign Language. Future focus: advancing CNNs for dynamic gestures, diverse sign languages, lighting, and translation.

[2.8] A Static Hand Gesture Based Sign Language Recognition System using CNN [8]

Includes hand segmentation, dataset creation, CNN design, training/testing, live testing, and evaluation. High accuracy and real-time output. Limitations: lighting, static gestures, computational time, background, Indian Sign Language. Future: advance CNNs for dynamic gestures, diverse sign languages, lighting, translation.

[2.9] Mudra: Convolutional Neural Network based Indian Sign Language Translator for Banks [9]

Mudra: CNN-based Indian Sign Language Translator for Banks. Involves hand segmentation, dataset creation, CNN design, training, live testing, and evaluation. Aims to bridge communication gap with the deaf-mute community, facing challenges due to the complexity of Indian Sign Language. Future steps include dataset expansion, sentence addition, automation, text-to-speech integration, and model exploration.

[2.10] SIGN LANGUAGE RECOGNITION FOR HINDI VARNAMALA USING CNN [10]

Hindi Varnamala Sign Language Recognition: Involves dataset creation, CNN training, prediction, and deployment as a web application. Aims to foster unity and communication in India. Focuses on expanding the dataset, improving accuracy, and ongoing refinement for effective Indian Sign Language (ISL) recognition. Future steps include enhancing accuracy, dataset expansion, and commercialization for broader adoption.

[2.11] A Multitask Sign Language Recognition System Using Commodity Wi-Fi [11]

Multitask Sign Language Recognition: Wi-SignFi system employs CSI data, eight-layer CNN, and KNN. Utilizes Wi-Fi sensing for wide coverage, low privacy concerns, and cost-effectiveness. Challenges include accuracy affected by input resolution. Future: integrate IoT, explore sensing, embed real-time applications.

[2.12] Dynamic Sign Language Recognition Based on CBAM with Autoencoder Time Series Neural Network[12]

Dynamic Sign Language Recognition: Employs convolutional self-coding network with CBAM for recognition, with preprocessing. Achieves 89.90% accuracy and generalization through continuous sample expansion. Challenges include monotonous input, lack of optical flow, and multimodal fusion. Future: fuse multimodal data, advance neural architectures for better accuracy.

Sr. No.	Name of the paper	Authors	Methodology	Advantages	Disadvantages	Future Work
1.	Sign Language Recognition in the Hindi Language Based on Computer Vision	Dr. Chhaya Grover Avni Rajpoot Aditya Verma Ayush Patel	Deep learning with CNNs and RNNs including preprocessing techniques.	Aims to translate sign language into Hindi, improving communication for speech-impaired individuals.	Limitation: Need larger, diverse Hindi sign dataset; challenges in real-time recognition.	Expand dataset, improve accuracy, recognize words, translate to Hindi.
2.	Hindi Sign Language Detection Using CNN	Prof. Pragya Sinha Ifham Khwaja Kaushik Rathod Naman Sanklecha	Data collection, OpenCV sign detection, preprocessing, CNN training, evaluation.	Improves communication for deaf and mute people, facilitating education and social interaction.	Concerns: Accuracy, equipment, lack of diverse training data hinder development.	Accuracy improvement, language coverage, real-time implementation, user interface, collaboration.
3.	Indian Sign Language Recognition using Convolutional Neural Network	Rachana patil Vivek Patil Abhishek Bahuguna Mr. Gaurav Datkhile	Image acquisition, segmentation, CNN training for Indian Sign Language.	Enhanced communication, accessibility, tech innovation, autonomy.	Model limitations: cultural sign variations, predicting image names, potential unaddressed issues.	Enhance image processing, enable bidirectional communication, real-time interpretation.
4.	Sign Language Recognition using Convolutional Neural Network with Customization	Heramba Limaye Shraddha Shinde Anurag Bapat Nimish Samant	Data collection, preprocessing, CNN building, GUI, real-time accuracy evaluation.	Real-time processing, feature extraction, and spatial correlation utilization.	Document lacks CNN disadvantages; notes incremental learning challenges for accuracy.	Future research: mitigate accuracy decline, alternative learning, advanced computer vision.
5.	Gesture Recognition Based on a Convolutional Neural Network–Bidirectional Long Short-Term Memory Network for a Wearable Wrist Sensor with Multi-Walled Carbon Nanotube/Cotton Fabric Material	Yang Song, Mengru Liu, Feilu Wang, Jinggen Zhu Anyang Huand Niuping Sun	Steps: fabricate sensors, integrate into wristband, collect gestures, CNN-BiLSTM, test accuracy.	Flexible, sensitive pressure sensor from MWCNT/CF composite, integrated into a wristband for accurate gesture recognition.	Challenges: Limited gesture coverage, real-world applicability, scalability, comparative analysis.	Advancements: HCI, health monitoring, assistive tech, smart textiles, industrial applications.
6.	Sign Language Recognition System Using Deep Neural Network	Surejya Suresh Mithun Haridas T.P. Supriya M.H	Method: 2-layer CNN for sign language classification using Python libraries.	TIdentifies six sign languages, utilizes deep learning, achieves high accuracy, and enables real-world device control.	Document lacks discussion on drawbacks or challenges of CNN implementation.	Enhance performance, video recognition, integrate with device control for sign language.

7.	A Static Hand Gesture Based Sign Language Recognition System using Convolutional Neural Networks	Diksha Hatibaruah Anjan Kumar Talukdar Kandarpa Kumar Sarma	Hand segmentation, dataset creation, CNN design, training/testing, live testing, evaluation.	High accuracy, real-time output, and customizable dataset.	Limitations: lighting, static gestures, computational time, background, Indian Sign Language.	Advance CNNs for dynamic gestures, diverse sign languages, lighting, translation.
8.	A Static Hand Gesture Based Sign Language Recognition System using Convolutional Neural Networks	Diksha Hatibaruah Anjan Kumar Talukdar Kandarpa Kumar Sarma	Hand segmentation, dataset creation, CNN design, training/testing, live testing, evaluation.	High accuracy, real-time output, and customizable dataset.	Limitations: lighting, static gestures, computational time, background, Indian Sign Language.	Advance CNNs for dynamic gestures, diverse sign languages, lighting, translation.
9.	Mudra: Convolutional Neural Network based Indian Sign Language Translator for Banks	Gautham Jayadeep Vishnupriya N V Vyshnavi Venugopal Vishnu S Geetha M	Method: Hand segmentation, dataset creation, CNN design, training, live testing, evaluation.	aims to bridge the communication gap between the deaf-mute community and banks using deep neural network models and demonstrating the effectiveness of advanced technology	Complexity of Indian Sign Language presents challenges in gesture recognition.	Expand dataset, add sentences, automate, add text-to-speech, explore models.
10.	SIGN LANGUAGE RECOGNITION FOR HINDI VARNAMALA USING CNN	Pratibha Gupta Priya Rajput Priyanka Katiyar Srishti Sharma Shaivya Shukla Dr. Umesh Dwivedi	Create dataset, train CNN, predict, deploy as web application.	Hindi sign language fosters unity, understanding, bridges communication in India.	Expand dataset, improve accuracy, ongoing refinement for effective ISL recognition.	Improve accuracy, expand dataset, commercialize for broader adoption.
11.	A Multitask Sign Language Recognition System Using Commodity Wi-Fi	Zhongjian Gao Chien-Cheng Lee Lianhui Zheng Ruige Zhang Xiaofu Xu1	Wi-SignFi: CSI data, eight-layer CNN, KNN for multitask recognition.	Wi-Fi sensing: wide range, less privacy concerns, cost-effective, easy deployment.	Accuracy influenced by input resolution, lightweight CNN may capture limitations.	Integrate IoT, capture dynamics, explore sensing, embed real-time applications.
12.	Dynamic Sign Language Recognition Based on CBAM with Autoencoder Time Series Neural Network	Yanglai Huang, Jing Huang, Xiaoyue Wu Yu Jia	Utilize convolutional self-coding network with CBAM for recognition, including preprocessing.	Method improves accuracy (89.90%) and generalization via continuous sample expansion.	Limitations: monotonous input, lack of optical flow, multimodal fusion.	Fuse multimodal data, advance neural architectures for better recognition accuracy.

3. SOFTWARE REQUIRMENTS SPECIFICTION

3.1 ASSUMPTIONS AND DEPENDENCIES

In any project or system, there are usually assumptions and dependencies that need to be considered in order to ensure its success. In the case of sign language recognition using deep learning, here are some possible assumptions and dependencies:

3.1.1 ASSUMPTIONS

- The project assumes that deep learning models, such as LSTM and GRU, can effectively detect and recognize sign language gestures from video frames, enabling accurate translation of sign language into text or speech.

- It assumes that the Sign Language MNIST dataset and custom datasets used for training the models adequately represent the diversity of sign language gestures and can generalize well to real-world scenarios.

- The project assumes that leveraging computer vision techniques, like hand tracking and dual-camera setups, can enhance the accuracy and robustness of the sign language recognition system, addressing challenges in gesture recognition and performance issues.

- It assumes that real-time sign language recognition systems developed through deep learning and computer vision can be deployed in various settings to improve communication and accessibility for individuals with hearing impairments.

- The project assumes that iterative improvements, such as collecting more quality data, exploring different CNN architectures, and integrating additional features like orthography correctors, can enhance the overall performance and accuracy of the sign language recognition system

3.1.2 DEPENDENCIES

- **Model Dependencies:**
  The project relies on deep learning models, specifically Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), for recognizing sign language gestures from video frames. These models play a crucial role in the accurate detection and translation of sign language gestures into text or speech.

- **Data Dependencies:**
  The project is dependent on high-quality datasets, such as the Sign Language MNIST dataset and custom datasets, for training and evaluating the deep learning models. The availability of diverse and representative data is essential for the models to learn and generalize effectively to real-world sign language gestures.

- **Technological Dependencies:**
  The project relies on computer vision techniques, such as hand tracking using Media Pipe and dual-camera setups, to enhance the accuracy and robustness of the sign language recognition system. These technological dependencies are crucial for addressing challenges in gesture recognition and improving the system's performance in real-time scenarios.

- **Research Dependencies:**
  The project may have dependencies on prior research and studies in the field of sign language recognition, deep learning, and computer vision. Leveraging existing research findings and methodologies can provide valuable insights and guidance for the development and improvement of the sign language recognition system

3.2 FUNCTIONAL REQUIREMENTS

1. **Sign Language Gesture Recognition:**

   The system should be able to accurately detect and recognize sign language gestures from visual input, such as video frames or images.

   The recognition should cover a wide range of sign language gestures, including both static and dynamic signs.

2. **Real-Time Translation:**

   The system should be capable of performing real-time translation of sign language gestures into text or speech, enabling seamless communication between individuals with hearing impairments and those who do not know sign language.

3. **Accessibility and Inclusivity:**

   The project aims to develop a system that can be deployed in various settings, such as educational institutions, workplaces, and public spaces, to improve accessibility and inclusivity for the deaf and hard-of-hearing community.

4. **Robustness and Accuracy:**

   The system should be robust and accurate in recognizing sign language gestures, minimizing ambiguity and errors in translation.

   The project should explore techniques like computer vision, dual-camera setups, and advanced deep learning models to enhance the overall performance and reliability of the sign language recognition system.

5. **Iterative Improvement:**

   The project should have a mechanism for continuous improvement, allowing for the collection of more quality data, exploration of different model architectures, and integration of additional features (e.g., orthography correctors) to enhance the system's capabilities over time.

6. **User-Friendly Interface:**
   The system should provide a user-friendly interface, making it accessible and easy to use for both individuals with hearing impairments and those who do not know sign language.

7. **Scalability and Adaptability:**
   The project should consider the scalability of the system, allowing for the recognition of a wider range of sign language gestures and the potential expansion to support multiple sign language dialects or languages.

3.2.1 SYSTEM FEATURE 1(FUNCTIONAL REQUIREMENT)

- Machine Learning Algorithm
- The trained model for Algorithm

3.2.2 SYSTEM FEATURE 2 (FUNCTIONAL REQUIREMENT)

- Dataset for Sign language recognition using deep learning.

3.2.3 HARDWARE INTERFACES

- **Laptops:** The personnel use laptops to access the system and view result.
- **Graphics Processing Units (GPUs):** GPUs are designed for parallel processing and are commonly used for accelerating machine learning tasks.

3.2.4 SOFTWARE INTERFACES

1. **Input Interface:**

   The project utilizes visual input, such as video frames or images, captured through a camera or webcam to detect and recognize sign language gestures.

   The interface likely includes functionality to capture and process the live video or image input from the user.

2. **Gesture Recognition:**

   The deep learning models, such as Convolutional Neural Networks (CNNs), LSTM, and GRU, are used to analyse the input visual data and recognize the sign language gestures.

   The interface should provide feedback to the user about the recognized gestures, potentially displaying the translated text or speech output.

3. **Real-Time Interaction:**

   The project aims to develop a real-time sign language recognition system, which suggests the interface should enable seamless and responsive interaction with the user.

   The interface may include features like hand tracking, gesture detection, and continuous recognition to provide a smooth and intuitive experience.

4. **Accessibility and Usability:**

   Since the project's goal is to improve communication and accessibility for individuals with hearing impairments, the interface should be designed with a focus on user-friendliness and accessibility.

   This may include features like clear visual feedback, intuitive controls, and the ability to adjust settings or preferences.

5. **Integration and Deployment:**

   The project mentions the potential for deploying the sign language recognition system in various settings, such as educational institutions, workplaces, and public spaces.

The interface may need to be designed with modularity and scalability in mind, allowing for easy integration and deployment in different environments.

3.3 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements are the characteristics that describe the performance, usability, and security of the system. They do not specify what the system should do, but rather how it should do it. Here are some non-functional requirements for our Sign language recognition using deep learning:

1. **Accuracy and Reliability:**

   The system should be able to accurately recognize and translate sign language gestures with a high degree of reliability, minimizing errors and ambiguity in the translation process.

   The project should explore techniques like hybrid approaches, combining deep learning models and computer vision, to enhance the overall accuracy and robustness of the sign language recognition system.

2. **Real-Time Performance:**

   The system should be capable of performing real-time sign language recognition and translation, enabling seamless and responsive communication between individuals with hearing impairments and those who do not know sign language.

3. **Scalability and Adaptability:**

   The project should consider the scalability of the system, allowing for the recognition of a wider range of sign language gestures and the potential expansion to support multiple sign language dialects or languages.

The system should be adaptable to different environments and settings, such as educational institutions, workplaces, and public spaces, to improve accessibility and inclusivity.

4. **User-Friendliness and Accessibility:**

The system should provide a user-friendly interface, making it accessible and easy to use for both individuals with hearing impairments and those who do not know sign language.

The project should prioritize the needs and requirements of the target user group, ensuring the system is designed with a focus on inclusivity and usability.

5. **Maintainability and Iterative Improvement:**

The project should have a mechanism for continuous improvement, allowing for the collection of more quality data, exploration of different model architectures, and integration of additional features to enhance the system's capabilities over time.

The system should be maintainable and easily updatable to address any issues or evolving requirements in the future.

3.3.1 PERFORMANCE REQUIREMENTS

- **Accuracy:** Accuracy is a fundamental measure of how well the model performs in correctly recognizing signs. It's typically defined as the percentage of correctly recognized signs over the total number of signs.

- **Real-time Recognition:** In many applications, real-time recognition is crucial, especially for interactive systems like sign language translation or interpretation tools. The model should be able to recognize signs quickly with minimal latency.

- **Robustness:** The model should be robust to variations in lighting conditions, backgrounds, signer appearance, and hand orientations. This ensures that it can perform well in diverse environments.

- **Vocabulary Size:** The size of the sign vocabulary the model can recognize is an important factor. For example, some applications may require recognition of a small set of basic signs, while others may require recognition of a larger, more diverse vocabulary.

- **Generalization:** The model should generalize well to unseen signers and variations in signing styles. This is particularly important for applications deployed in the real world where signers may have different signing styles.

- **Efficiency:** The model should be computationally efficient, especially for deployment on resource-constrained devices like smartphones or embedded systems.

- **Data Efficiency:** Deep learning models for sign language recognition often require large amounts of data for training. However, efficient use of data is important, especially when labelled data is scarce.

- **Confusion Matrix Analysis:** Understanding which signs are frequently confused by the model can provide insights into areas for improvement. For example, some signs may look similar, leading to confusion.

- **User Feedback Integration:** Systems should ideally incorporate mechanisms to collect user feedback to continuously improve recognition accuracy and adapt to user-specific signing styles.

- **Integration with other Systems:** For certain applications, integration with other systems such as natural language processing (for translation) or user interfaces (for interaction) may be necessary. The performance should be measured in terms of the overall effectiveness of these integrated systems.

- **Error Analysis:** Detailed error analysis can help identify common mistakes made by the model and guide improvements. For instance, understanding if errors are due to misclassification, ambiguity in signs, or environmental factors can be crucial.

- **Adaptability:** The ability of the model to adapt and improve over time with more data or feedback is desirable, especially for systems deployed in dynamic environments.

3.3.2 SAFETY REQUIREMENTS

1. **User Safety:**

   • The system should be designed with user safety in mind, ensuring that the input capture devices (e.g.

   cameras) do not pose any physical or privacy risks to the users.

   • The system should not require users to wear or use any specialized equipment that could be potentially hazardous or uncomfortable.

2. **Data Privacy and Security:**

   • The project should implement robust data privacy and security measures to protect the personal information and visual data of the users, especially for individuals with hearing impairments.

   • Appropriate data handling protocols, encryption, and access controls should be in place to ensure the confidentiality and integrity of the user data.

3. **Environmental Safety:**

• The deployment of the sign language recognition system should consider any potential environmental impacts, such as the energy consumption or heat generation of the hardware components.

• The system should be designed to operate safely and efficiently within the intended deployment environments, without posing any risks to the surrounding infrastructure or personnel.

4. **Accessibility and Inclusivity:**

   • The project should prioritize the safety and accessibility of the sign language recognition system for users with diverse abilities, including those with additional disabilities or special needs.

   • The interface and interaction design should be inclusive and accommodate the needs of all users, ensuring a safe and inclusive experience.

5. **Regulatory Compliance:**

   • The project may need to adhere to relevant safety regulations, standards, or guidelines related to the development and deployment of assistive technologies, especially in public or institutional settings.

   • Compliance with applicable laws and regulations regarding data privacy, accessibility, and the use of computer vision or deep learning technologies should be ensured.

3.3.3 SOFTWARE QUALITY ATTRIBUTES

Software quality attributes are the characteristics of the software system that determine its overall quality.

Some of the important software quality attributes for our drone delivery system include:

Accuracy:

The project aims for high accuracy in recognizing sign language gestures, with models achieving remarkable accuracy levels above 90% and 96.47% in recognizing American Sign Language and British Sign Language gestures, respectively.

Real-Time Performance:

The system is designed for real-time sign language recognition, ensuring prompt and responsive translation of gestures into text or speech, enhancing communication for individuals with hearing impairments.

Efficiency:

The use of Convolutional Neural Networks (CNNs) and Computer Vision techniques ensures efficient processing of visual data, enabling the system to detect and classify sign language gestures with a high degree of accuracy upon sufficient training.

Robustness:

The system is expected to be robust in recognizing a wide range of sign language gestures, minimizing errors and confusion in gesture classification, and ensuring reliable translation into text or speech.

Scalability:

The project considers scalability to accommodate a growing dataset of sign language gestures and the potential expansion to support multiple sign languages or dialects, ensuring the system's adaptability to diverse user needs.

User-Friendliness:

The interface is designed to be user-friendly, making it accessible and easy to use for individuals with hearing impairments and those unfamiliar with sign language, enhancing the overall user experience and inclusivity.

3.4 SYSTEM REQUIREMENTS

3.4.1 SOFTWARE REQUIREMENTS (PLATFORM CHOICE)

- Visual studio - Visual Studio is a comprehensive integrated development environment (IDE) developed by Microsoft that supports multiple programming languages and platforms, offering a wide array of features to enhance software development. It includes a Solution Explorer for navigating project files, an Editor for writing code with syntax highlighting and IntelliSense, a menu bar for accessing commands, an Error List for identifying issues, a Search function for quickly finding tools and options, a built-in Terminal, and customization options. Visual Studio also provides features for building, running, and debugging applications, as well as code editor tools like outlining, code snippets, and generating code from usage. Additionally, Visual Studio integrates with GitHub Copilot, an AI-powered coding companion, and supports cloud development with Azure services

- Windows 7 to higher

3.4.2 HARDWARE REQUIREMENTS

- Processor Any Processor after Pentium 3

- Ram 8 GB

- Active internet connection is needed

3.4.3 LIBRARIES

1. TensorFlow – TensorFlow is utilized in sign language recognition systems through its support for building Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. With TensorFlow's Keras API, CNN models can be easily constructed for static gestures, while LSTM networks are ideal for dynamic gestures. Transfer learning with pre-trained models and real-time detection using TensorFlow's object detection API and OpenCV further enhance sign language recognition systems. TensorFlow's versatility and high-level APIs make it a popular choice for developing advanced deep learning models in this field.

2. Open cv – OpenCV is a crucial library used in sign language recognition systems with deep learning. It facilitates real-time computer vision tasks like gesture detection and recognition. Its functions accelerate machine perception, making it ideal for applications like sign language recognition.

3. Mediapipe – Mediapipe is a crucial library used in sign language recognition systems with deep learning. It enables efficient extraction of hand landmarks and poses from images and videos. By combining Mediapipe with Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTMs), highly accurate real-time sign language recognition systems can be developed. Mediapipe's ability to detect hand gestures and poses in a computationally efficient manner makes it ideal for building practical sign language recognition applications

3.4.4 APPLICATIONS

1. Improving Communication and Accessibility for the Deaf and Hard-of-Hearing Community:

- The primary application of this project is to bridge the communication gap between individuals with hearing impairments and those who do not know sign language.

- By translating sign language gestures into text or speech, the system aims to enhance accessibility and enable seamless communication for the deaf and hard-of-hearing community.

2. Deployment in Various Settings:

- The project envisions deploying the sign language recognition system in different settings, such as educational institutions, workplaces, and public spaces, to improve inclusivity and accessibility for individuals with hearing impairments.

3. Assistive Technology for the Deaf and Mute:

- The project's goal is to develop an assistive technology that can help deaf and mute individuals communicate more effectively with those who do not know sign language.

4. Potential for Cross-Language Sign Language Recognition:

The project suggests the possibility of expanding the system to support multiple sign language dialects or languages, further enhancing its reach and applicability.

5. Integration with Text-to-Speech or Speech-to-Text Conversion:

- Combining the sign language recognition system with text-to-speech or speech-to-text conversion capabilities can enable two-way communication, improving accessibility for individuals with hearing impairments.

3.4 ANALYSIS MODELS:

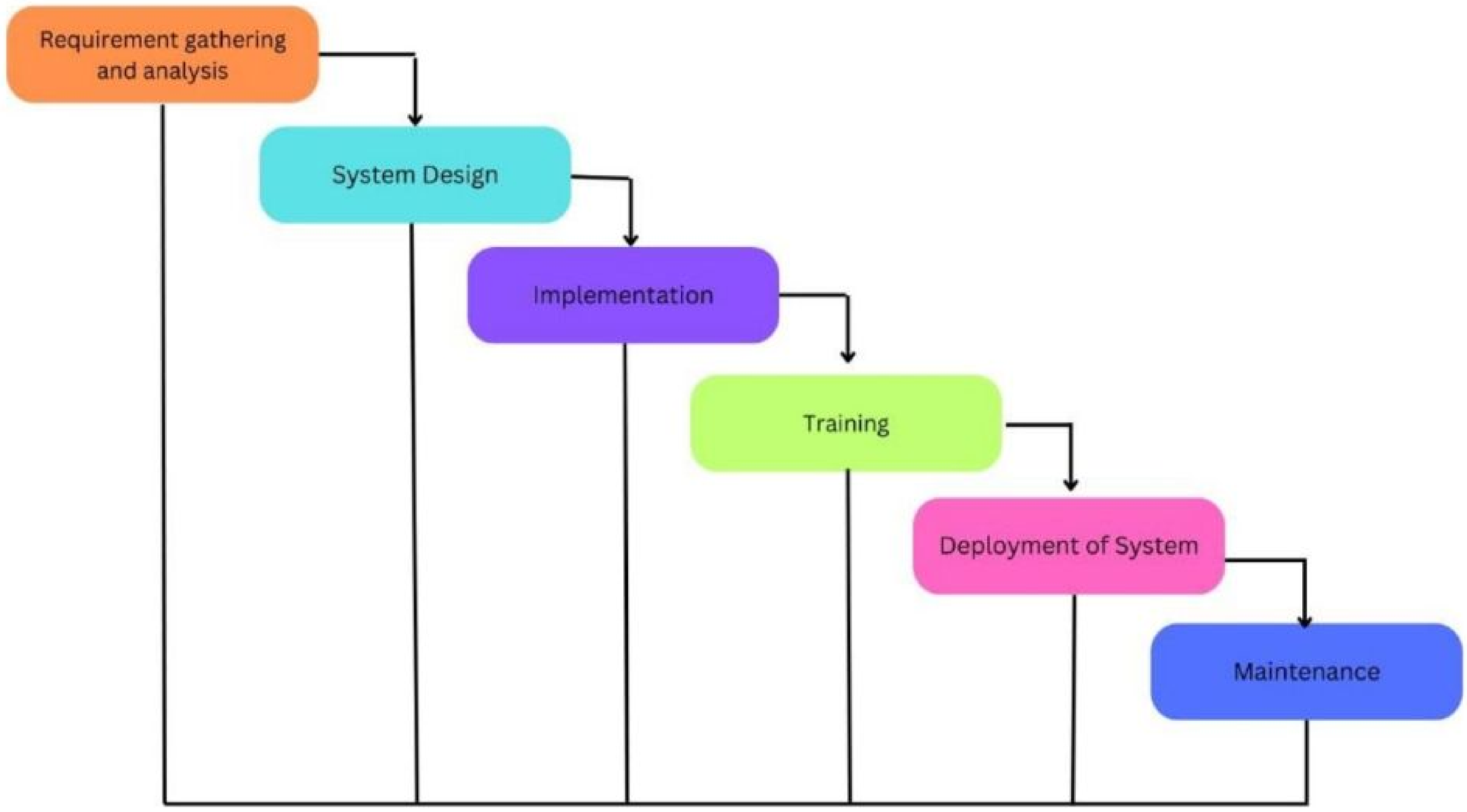

Fig 3.1 Analysis Model

We are using the waterfall model for our project**.**

1. Requirement gathering and analysis: In this step of the waterfall we identify what various requirements are needed for our projects such as software and hardware required, database, and interfaces.
2. System Design: We design some data flow diagrams to understand the system flow and system module and sequence of execution.
3. Implementation: In the implementation phase of our project, we have implemented various models required to successfully get the expected outcome at the different module levels.
4. Testing: The different test cases are performed to test whether the project modules are giving the expected outcome in the assumed time.
5. Deployment of System: Once the functional and non-functional testing is done, the product is deployed in the customer environment or released into the market.

6. Maintenance: There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released.

The first step in the process is data collection, where a large dataset of sign language gestures is gathered. This dataset is then used to train a deep learning model, which learns to recognize different signs based on the patterns in the data. The model is typically trained using a type of deep learning algorithm known as a convolutional neural network (CNN), which is well-suited for processing visual data.

Once the model has been trained, it can be deployed in a real-time recognition system. This system typically involves capturing video input of sign language gestures, preprocessing the data to make it suitable for input into the deep learning model, and then passing the preprocessed data through the model to obtain a prediction of the sign being performed. The output of the model can then be converted into text or speech for the hearing user to understand.

One of the main challenges in developing a sign language recognition system using deep learning is the need for a large and diverse dataset. In order for the deep learning model to learn to recognize a wide range of signs, it needs to be trained on a dataset that contains examples of many different signs performed by various individuals. Collecting and labeling such a dataset can be a time-consuming and labor-intensive process, but it is essential for the success of the system.

Another challenge is the need to account for variations in sign language gestures performed by different individuals. Just as with spoken languages, sign languages can vary in terms of regional dialects and individual differences in signing styles. The deep learning model must be robust enough to generalize across these variations and accurately recognize signs regardless of who is performing them.

To address these challenges, researchers are exploring various methods for augmenting datasets with synthetic data, using techniques such as data augmentation and domain adaptation. These methods can help to increase the diversity of the training data and make the deep learning model more robust to variations in signing styles.

In conclusion, the system architecture of a sign language recognition system using deep learning is a complex and multifaceted process that involves data collection, model training, and real-time recognition. While there are challenges to overcome, advancements in deep learning technology offer great promise for improving accessibility and inclusion for the deaf and hard of hearing community. By continuing to research and develop these systems, we can help to break down communication barriers and create a more inclusive society for all.

4 SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

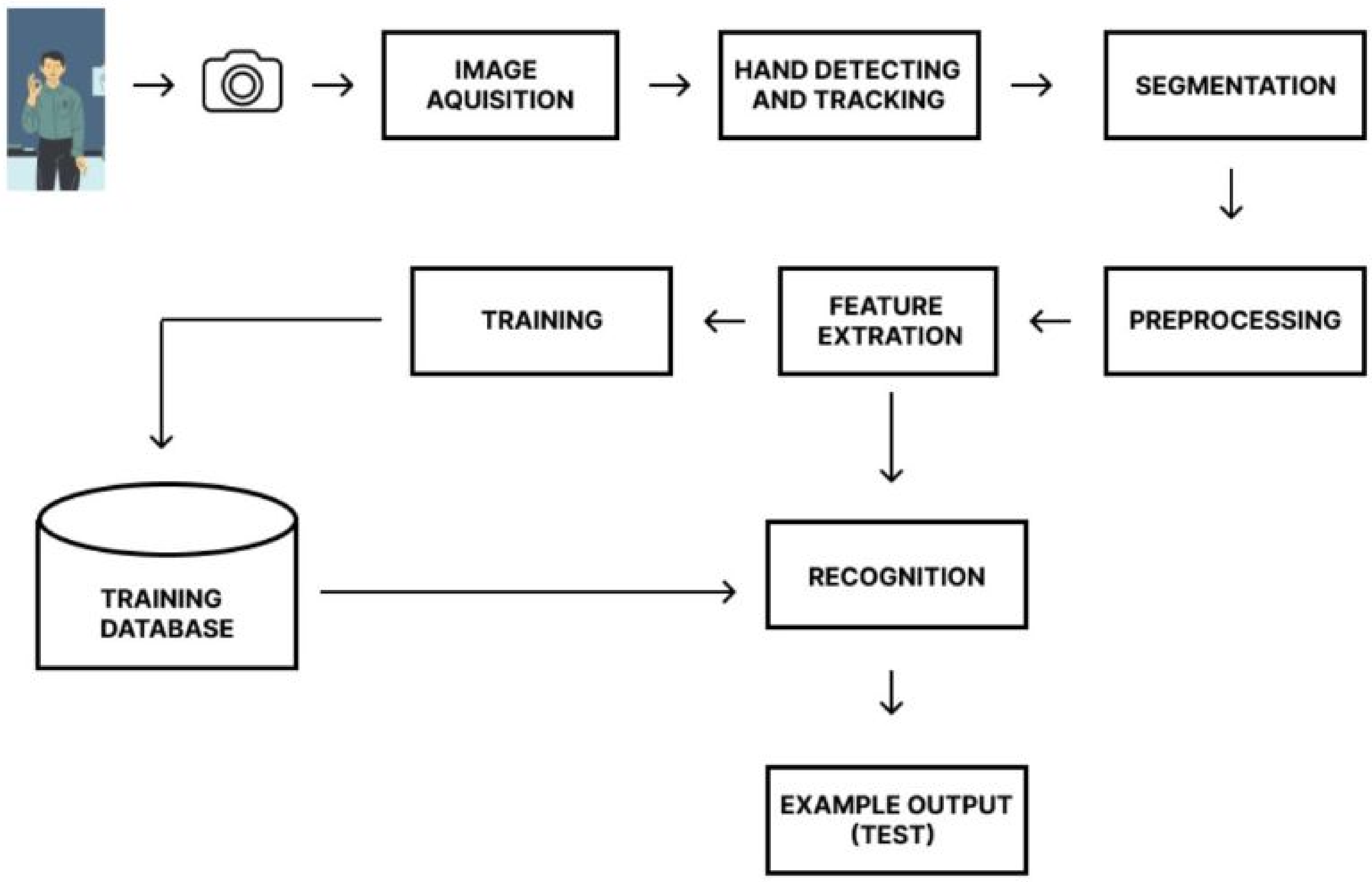

Fig. 4.1 System Architecture

- **Image acquisition**: This step involves capturing the hand gesture image from the camera.
- **Hand detecting and tracking**: The system locates and tracks the hand in the image.
- **Segmentation**: This stage isolates the hand from the background in the image.
- **Preprocessing**: The image of the hand is cleaned up and prepared for further processing.

- **Feature extraction**: The system identifies key features of the hand gesture (like shape, size, orientation) that are important for recognition.
- **Recognition**: The extracted features are compared against a database of known hand gestures to identify the specific gesture being made.
- **Training**: The system is trained on a dataset of known hand gestures and their corresponding features.
- **Training database**: A database stores the information about the hand gestures and their features for training. The output of the system is the recognized hand gesture, which can be used for various applications like controlling devices, interacting with virtual environments, or communicating with computers.

4.2. FLOWCHART

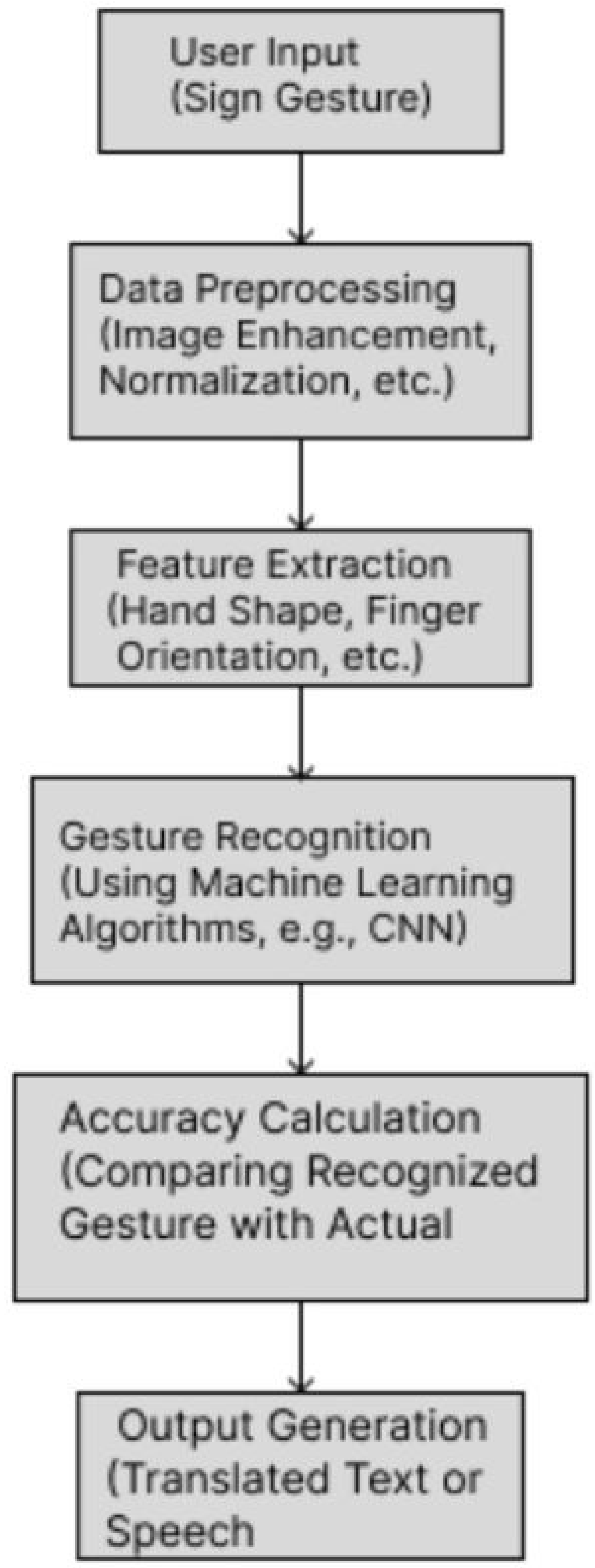

Fig. 4.2 Sign language recognition Flow Diagram

1. **User Input:** In this step, the system takes input in the form of sign gestures made by the user.

2. **Data Preprocessing:** The input data is preprocessed in this step to enhance the image quality, normalize it, and perform other necessary enhancements to prepare it for further processing.

3. **Feature Extraction:** Features relevant to the hand shape, finger orientation, and other relevant aspects are extracted from the preprocessed data.

4. **Gesture Recognition:** Machine learning algorithms, such as Convolutional Neural Networks (CNN), are used in this step to recognize the gesture based on the extracted features.

5. **Accuracy Calculation:** The recognized gesture is compared with the actual gesture to calculate the accuracy of the recognition process.

6. **Output Generation:** Finally, the system generates an output, which could be translated text or speech based on the recognized gesture.

4.3. BLOCK DIAGRAM OF SOFTWARE

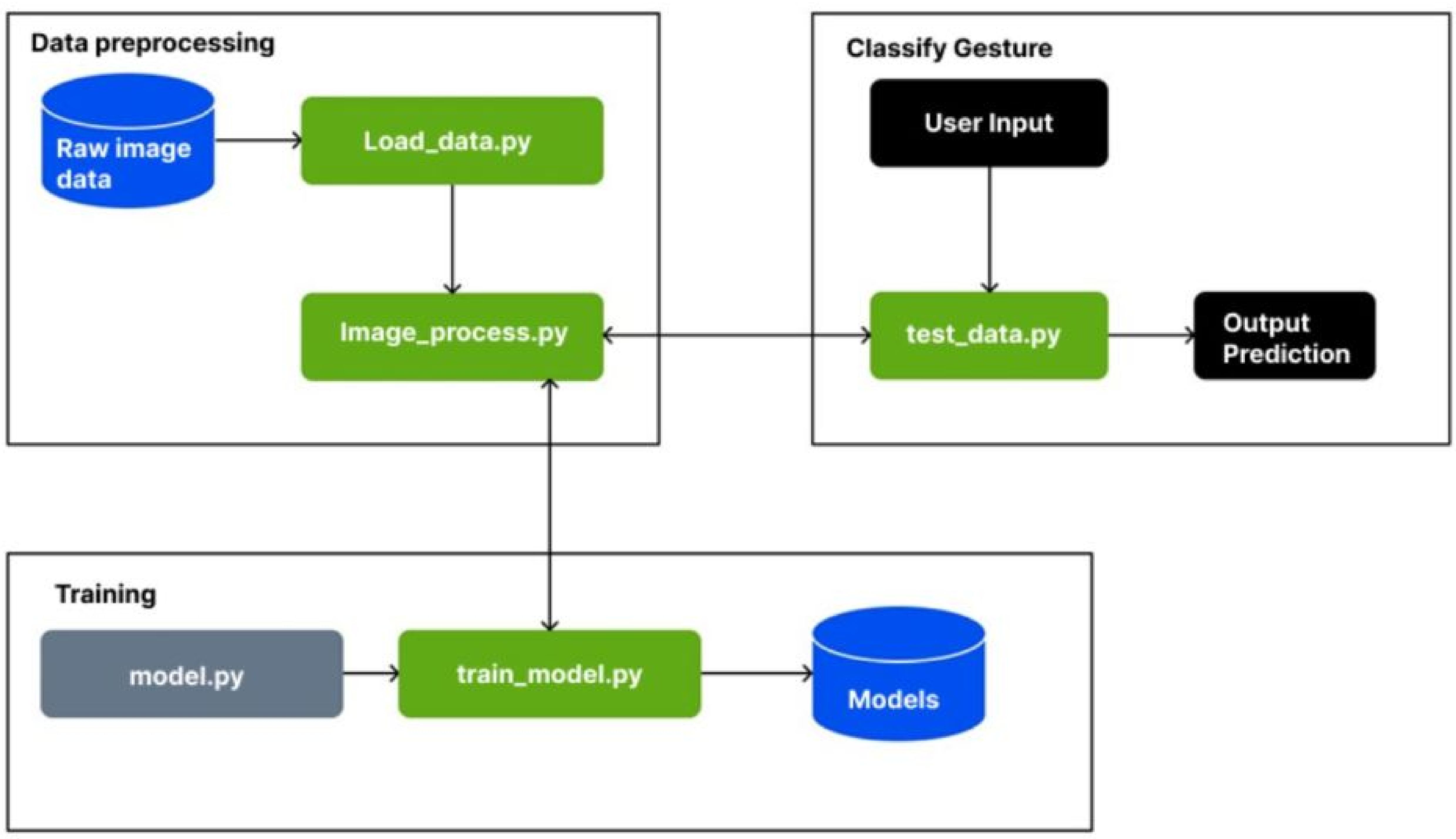

Fig. 4.3 Block Diagram of software

The diagram shows a workflow for gesture classification. The process starts with raw image data, which is loaded and processed, then the model is trained on this processed data to create a model that can predict the gesture. When a user inputs a new image, the system uses the trained model to classify the gesture and output the prediction.

5 PROJECT PLAN

5.1 PROJECT RESOURCES

Here are some of the key resources required for a Sign language recognition using deep learning project:

1. Hardware: The hardware resources required for a system include laptops, memory and other related hardware.
2. Software: The software resources required for a system include software development tools, programming languages, algorithms. The software should also be integrated with the hardware components to ensure that the system works seamlessly.
3. Team members: A system project team should include individuals with diverse skill sets, such as software developers, hardware engineers, data analysts, project managers, and quality assurance specialists. The size of the team will depend on the scope and complexity of the project.
4. Data: The system will need access to data such as data packets, destination IP, Timestamps.
5. Legal and regulatory support: The project may require support from legal and regulatory experts to ensure compliance with local laws and regulations, obtain necessary permits and licenses, and manage any legal issues that may arise.
6. Funding: The project will require funding to cover the cost of hardware, software, infrastructure, salaries, legal and regulatory fees, and other related expenses.

Overall, a successful sign language recognition project will require a combination of hardware, software, team members, data, infrastructure, legal and regulatory support, and funding. It is essential to carefully plan and allocate resources to ensure that the project is completed on time, within budget, and meets the objectives.

5.2 RISK MANAGEMENT

5.2.1 RISK IDENTIFICATION

Overfitting Risk:

The search results indicate that one of the challenges in sign language recognition using deep learning is the risk of overfitting, which can lead to poor generalization of the models.

Overfitting occurs when the model learns the training data too well, resulting in poor performance on new, unseen data. This is a common issue in deep learning and needs to be addressed through appropriate model design and training techniques.

Ambiguity in Gesture Recognition:

The project faces the risk of ambiguity in recognizing and distinguishing between similar sign language gestures.

This can lead to errors in translation and communication, as the system may confuse certain signs with others, impacting the overall accuracy and reliability of the sign language recognition.

Performance Limitations:

The search results suggest that the project may encounter performance issues, where the system's real-time recognition capabilities may not be consistently optimal.

This could impact the seamless and responsive communication that the project aims to achieve, potentially limiting the system's practical deployment and usability.

Scalability and Adaptability Challenges:

As the project aims to support a wide range of sign language gestures and potentially expand to multiple sign language dialects or languages, there is a risk of scalability and adaptability challenges.

Ensuring the system can accommodate a growing dataset and adapt to diverse sign language requirements without compromising performance is a key risk factor.

Data Quality and Diversity:

The quality and diversity of the training data used for the deep learning models can pose a risk to the overall performance and generalization of the sign language recognition system.

Insufficient or biased data may limit the models' ability to accurately recognize a wide range of sign language gestures, leading to suboptimal performance.

5.2.2 RISK ANALYSIS

The key risks identified in the project include the risk of overfitting, where the deep learning models may learn the training data too well, leading to poor generalization and performance on new, unseen data. This requires careful regularization techniques and model tuning to mitigate. Another risk is the ambiguity in recognizing and distinguishing between similar sign language gestures, which can result in errors in translation and communication. Enhancing the models' ability to differentiate subtle differences in hand movements and positions is crucial to address this. The project also faces the risk of performance limitations, particularly in real-time recognition scenarios, affecting the system's responsiveness and accuracy. Ensuring optimal system performance and efficiency is key to mitigate this. Additionally, there are scalability and adaptability challenges as the project aims to support a wide range of sign language gestures and potentially expand to multiple sign languages or dialects, requiring the system to accommodate a growing dataset and diverse requirements without compromising performance or usability. Finally, the risk of insufficient or biased training data leading to suboptimal recognition accuracy must be addressed by ensuring a diverse and high-quality dataset. By addressing these key risks, the project can enhance the reliability, accuracy, and effectiveness of the sign language recognition system.

5.2.3 Overview of Risk Mitigation, Monitoring, Management

Risk Mitigation Strategies:

Data Quality Assurance: Ensure the quality, diversity, and representativeness of the training data to avoid biases and improve model performance

Model Interpretability: Use interpretable models and techniques to enhance transparency and understandability of the system's decisions

Robustness Testing: Conduct extensive testing to evaluate the model's performance under various scenarios, including noisy environments and different lighting conditions

Ethical Considerations: Address ethical concerns related to privacy, fairness, and inclusivity in the design and deployment of the system

Continuous Monitoring: Implement monitoring mechanisms to track the system's performance in real-time and detect any anomalies or drift in model behaviour.

Risk Management and Monitoring:

Regular Evaluation: Continuously assess the system's accuracy, reliability, and performance metrics to identify potential risks and areas for improvement

Feedback Mechanisms: Incorporate feedback loops to gather user feedback and improve the system's usability and effectiveness

Security Measures: Implement robust security protocols to protect sensitive data and prevent unauthorized access to the system

Compliance: Ensure compliance with data protection regulations and standards to mitigate legal risks associated with handling personal information

5.3 PROJECT TASK SET

The project task set for a sign language detection project may include the following tasks:

1. Project planning and scoping
2. System design and development
3. Procurement and installation
4. Testing and quality assurance
5. Training and certification of personnel
6. Implementation and deployment
7. Monitoring and maintenance
8. Risk management and mitigation
9. Performance evaluation and reporting.

6 PROJECT IMPLEMENTATION

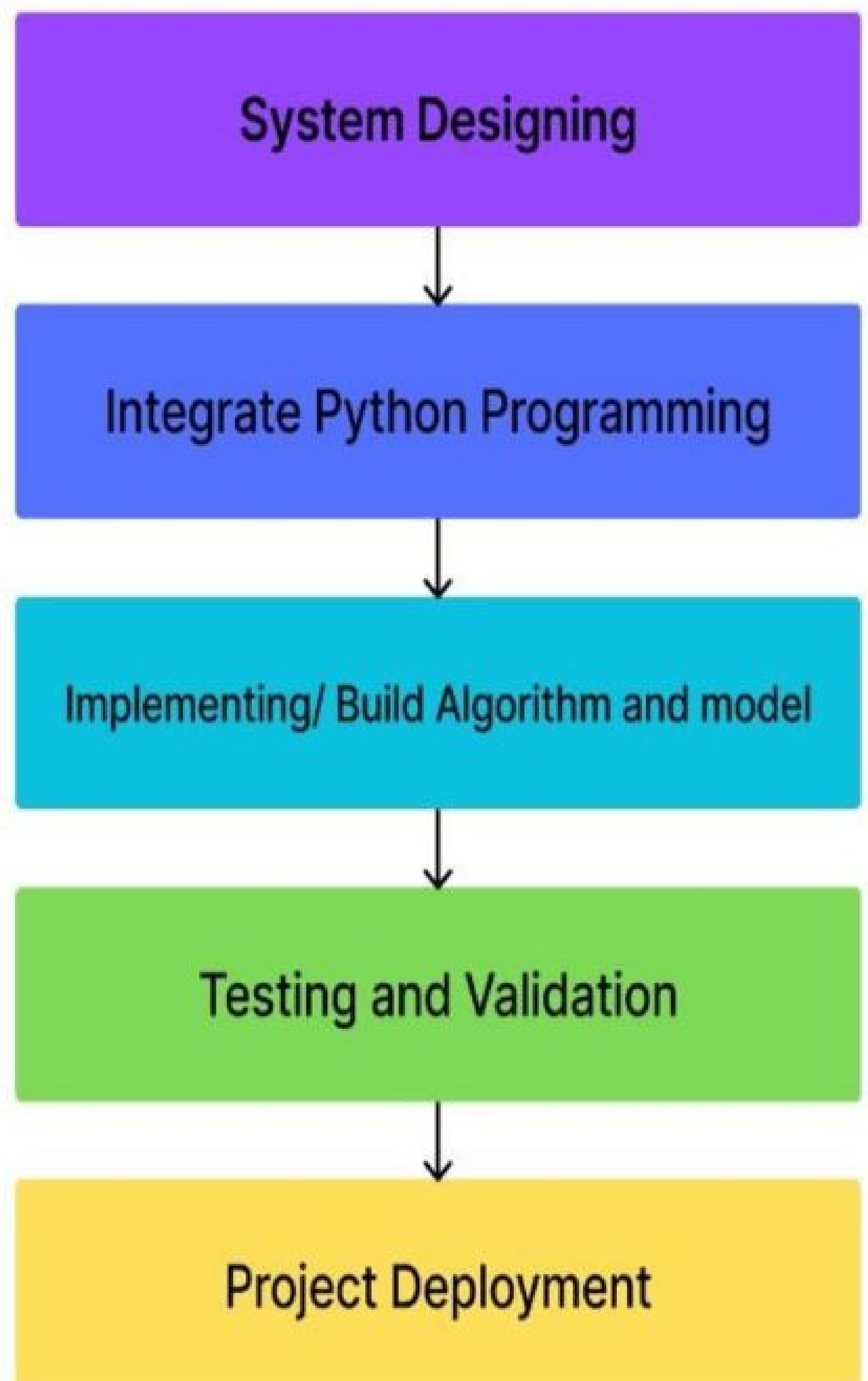

Fig 6.1 diagram of project implementation

6.1 TOOLS AND TECHNOLOGIES USED

- Programming Languages: Python
- Frameworks: media pipe, TensorFlow, open cv
- Development Tools: Visual Studio

6.2 ALGORITHM DETAILS

Deep Learning Models:

- Convolutional Neural Networks (CNNs) for feature extraction and classification of sign language gestures.
- Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models for recognizing and translating sign language gestures from video frames

6.2.1 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) have been widely used in sign language recognition systems due to their ability to effectively learn and extract features from image data. Here are the key details of how CNNs are applied in sign language recognition using deep learning:

Architecture

A typical CNN architecture for sign language recognition consists of the following layers:

Input layer: Takes in the image of the hand gesture

Convolutional layers: Apply filters to extract low-level features like edges and high-level features like hand shapes

Pooling layers: Reduce the spatial size of the feature maps to decrease computational complexity

Fully connected layers: Classify the hand gesture based on the extracted features

The convolutional layers are the core of the CNN, applying a set of learnable filters to the input image. Each filter is convolved across the width and height of the input, computing the dot product between the filter and the input patch, producing a 2D activation map of that filter.

Mathematical Formulation

Let the input image be represented as I(x,y) and the filter as K(x,y). The convolution operation is defined as:

$$(I * K)(x,y) = \sum\sum I(x-i, y-j) * K(i,j)$$

Where * represents the convolution operation. The output of the convolution is an activation map, which is then passed through a non-linear activation function like ReLU.

The pooling layer reduces the spatial dimensions of the feature maps by applying a max or average operation over a local neighbourhood. This helps in reducing the number of parameters and computation in the network.

The fully connected layers at the end of the CNN act as a classifier, taking the high-level features from the convolutional layers and outputting the predicted sign language gesture.

Training

The CNN is trained end-to-end using labeled sign language gesture images. The weights of the filters are initialized randomly and updated using backpropagation to minimize the loss between the predicted and true labels.

Data augmentation techniques like flipping, rotation, and scaling are often used to increase the diversity of the training data and improve generalization.

Applications

CNNs have been successfully applied to recognize both static hand shapes and dynamic hand gestures in sign language. They have achieved state-of-the-art performance on benchmark sign language recognition datasets.

Some key applications include:

Fingerspelling recognition for American Sign Language (ASL)

Isolated sign recognition for various sign languages

Continuous sign language recognition for sentence-level communication

In summary, CNNs provide a powerful framework for learning robust features from sign language gesture images and classifying them with high accuracy. Their end-to-end trainable nature and ability to handle complex spatial patterns make them well-suited for sign language recognition tasks.

6.2.1 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks are a type of Recurrent Neural Network (RNN) that have been widely used in sign language recognition systems due to their ability to effectively learn and classify sequential data.

Architecture

The LSTM architecture consists of memory cells, input gates, output gates, and forget gates. The memory cells store information over long periods of time, while the gates control the flow of information into and out of the cells.

Mathematical Formulation

The LSTM architecture can be formulated mathematically as follows:

Memory Cell: The memory cell state at time t is represented as Ct. It is updated based on the previous cell state Ct-1, the input gate it, the forget gate ft, and the output gate ot.

$$Ct = ft * Ct - 1 + it * tanh(Wc * Xt + bc)$$

Input Gate: The input gate it controls the flow of new information into the memory cell. It is computed based on the input Xt, the previous hidden state ht-1, and the input gate weights Wi and bi.

$$it = sigmoid(Wi * Xt + bi)$$

Forget Gate: The forget gate ft determines what information to discard from the previous memory cell state. It is computed based on the input Xt, the previous hidden state ht-1, and the forget gate weights Wf and bf.

$$ft = sigmoid(Wf * Xt + bf)$$

Output Gate: The output gate ot controls the output of the LSTM cell. It is computed based on the input Xt, the previous hidden state ht-1, and the output gate weights Wo and bo.

$$ot = sigmoid(Wo * Xt + bo)$$

Hidden State: The hidden state ht is computed based on the memory cell state Ct and the output gate ot.

$$ht = ot * tanh(Ct)$$

Training

The LSTM network is trained using backpropagation through time (BPTT) to minimize the loss between the predicted and true labels.

Applications

LSTM networks have been successfully applied to recognize both static hand shapes and dynamic hand gestures in sign language. They have achieved state-of-the-art performance on benchmark sign language recognition datasets.

Some key applications include:

Fingerspelling recognition for Indian Sign Language (ISL)

Isolated sign recognition for various sign languages

Continuous sign language recognition for sentence-level communication

In summary, LSTM networks provide a powerful framework for learning robust features from sequential sign language gesture data and classifying them with high accuracy. Their ability to handle long-term dependencies and capture temporal patterns makes them well-suited for sign language recognition tasks

Hybrid Approach:

- The project utilizes a hybrid approach, combining LSTM and GRU models to capture the semantic dependencies in sign language gestures more effectively

Computer Vision Techniques:

- Hand tracking using Media Pipe to enhance the accuracy and robustness of the sign language recognition process.
- Dual-camera setup (head-mounted and chest-mounted) to provide different perspectives and address ambiguity in gesture recognition

Regularization and Optimization:

- Dropout layers to reduce overfitting and improve the model's generalization ability
- Kernel regularize (L2 regularization) to prevent overfitting

Dataset and Training:

- Utilization of the Sign Language MNIST dataset and custom datasets for training and evaluating the deep learning models
- Splitting the dataset into training and validation sets to enable effective model training and evaluation

Real-Time Processing:

- The system is designed to perform real-time sign language recognition and translation, enabling seamless communication

Iterative Improvement:

- Acknowledgment of the need for further analysis and enhancements, such as collecting more quality data, exploring different CNN architectures, and integrating additional features (e.g., orthography correctors).

7. Result

7.1.Data collection (making dataset)

Fig 7.1 data collection

- This is the first step, where we collect the frames, in short we are making the dataset.
- We have made 30 frames for each alphabet.
- The model collects all the 30 frames for each alphabet and store it in a folder.
- collecting data for a sign recognition system, the image likely captures the hand gesture for the letter "A" in American Sign Language. This image would be one of many data points used to train the system to accurately recognize and interpret sign language gestures

7.2.Data model training using Media pipe

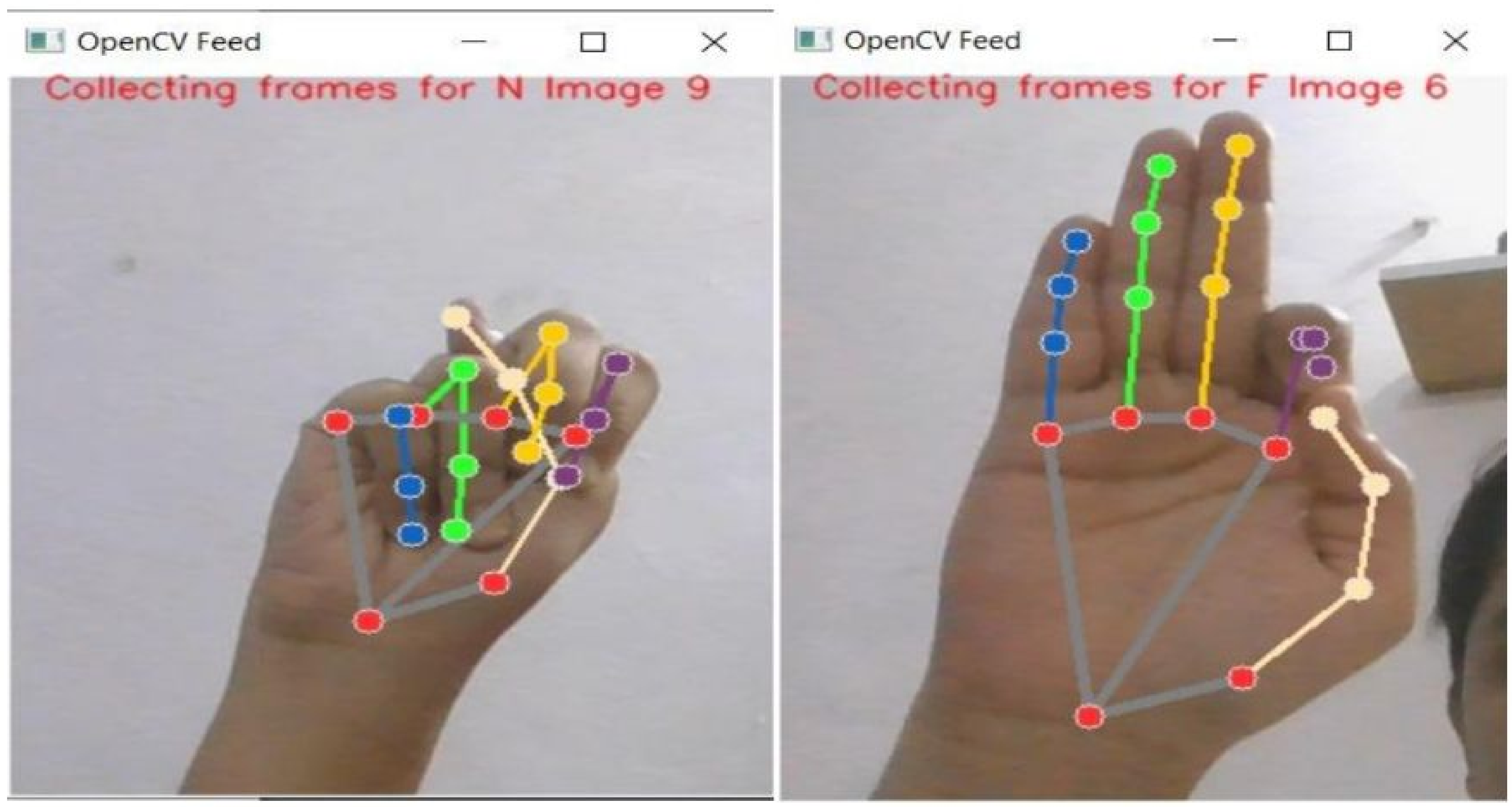

Fig 7.2 data model training using media pipe

- In this step the model is training itself before execution.
- The frames that we have made in previous steps are being collected and processed in this step.
- The model first collects all the frames i.e. dataset for further execution.
- The image shows a hand with its fingers extended, with a number of points marked on the hand. The points are connected by lines, and the image appears to be a frame from a video used to train a computer model for sign language recognition. The title of the window indicates that the frame is the 9th one captured. The different colors of the lines and points probably represent different features of the hand being tracked by the system for sign language recognition.

7.3. Sign Detection

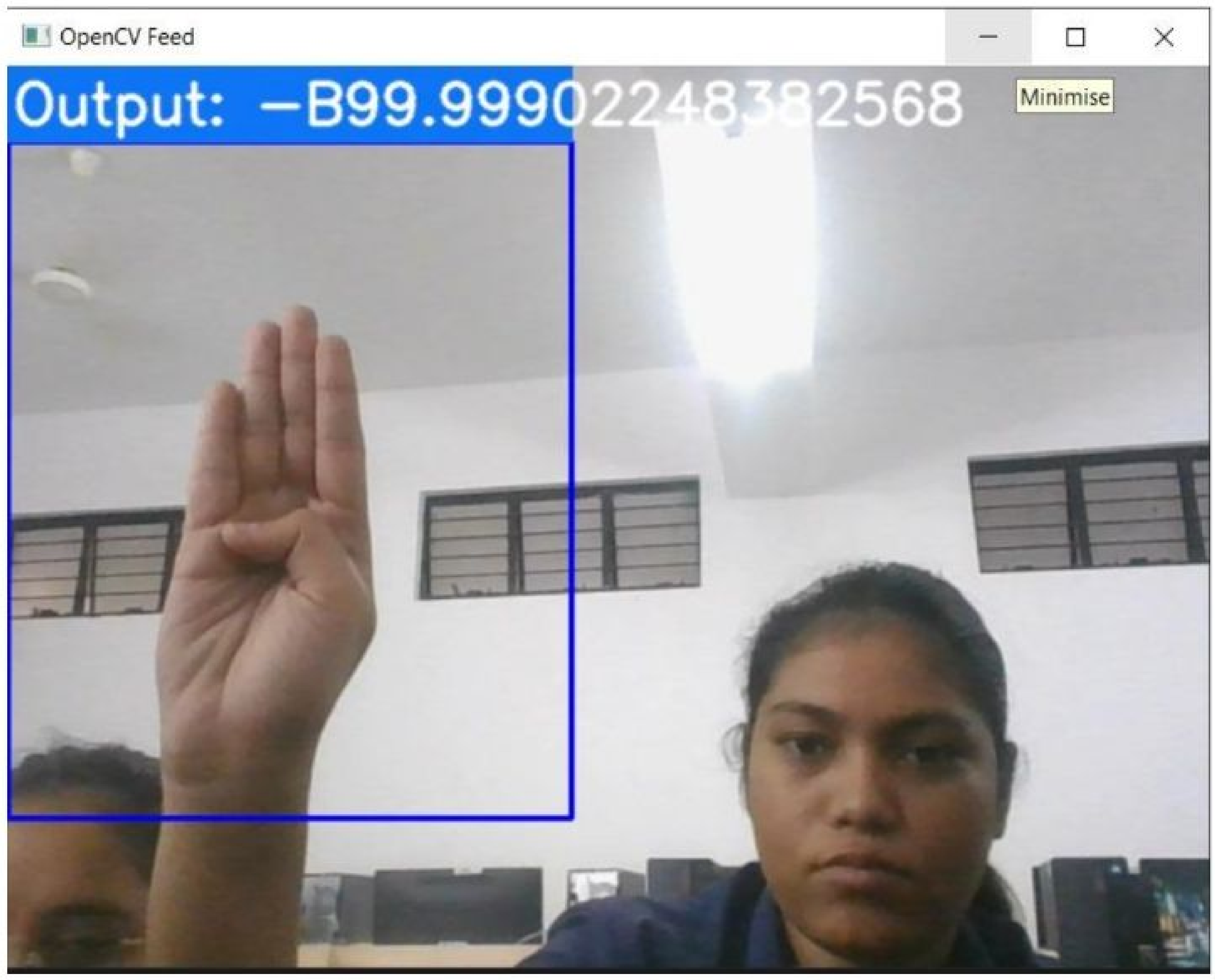

Fig 7.3 Sign/Gesture detection

- This is the stage where our model is detecting the alphabets.
- It also show the accuracy at which the alphabet is being detected.
- We just have to do a gesture in the blue frame and the alphabet will be detected.
- We can even see the alphabet that is being recognized alongside of accuracy.
- We can see which alphabet is being detected in terminal as well.
- The image shows a person performing a sign language gesture, likely being captured by a sign language recognition system. The system has recognized the gesture with an accuracy of -B99.99902248382568. This suggests that the recognition was not successful, as accuracy scores typically range from 0 to 1, with 1 indicating perfect accuracy. The negative value and very large magnitude indicate a significant error in the recognition process.

8. CONCLUSION

8.1 CONCLUSION

Based on the information gathered, the "Sign Language Recognition using Deep Learning" project demonstrates significant potential in leveraging advanced technologies to enhance communication and accessibility for individuals with hearing impairments. By employing deep learning models such as Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU), coupled with computer vision techniques like hand tracking and dual-camera setups, the project aims to accurately recognize and translate sign language gestures in real-time. The system's iterative approach to data collection, model training, and performance evaluation highlights a commitment to continuous improvement. However, challenges such as overfitting, ambiguity in gesture recognition, and scalability issues pose risks that need to be carefully managed. By addressing these risks and focusing on enhancing system performance, accuracy, and adaptability, the project has the potential to make a significant impact in bridging communication barriers and promoting inclusivity for individuals with hearing impairments.

8.2 FUTURE WORK

1. Expanding the Dataset and Improving Data Quality:

- The project acknowledges the need to collect more quality data to improve the generalization capabilities of the deep learning models.

- Expanding the dataset to cover a wider range of sign language gestures, including dynamic signs and facial expressions, can enhance the system's recognition abilities.

2. Exploring Different CNN Architectures:

- The researchers suggest exploring different Convolutional Neural Network (CNN) architectures, beyond the InceptionResNetV2 used in the initial prototype, to potentially improve the feature extraction and classification performance.

3. Integrating Orthography Correctors and Word Predictors:

- Incorporating additional features like orthography correctors or word predictors can help address the ambiguity in gesture recognition and improve the overall accuracy and reliability of the sign language translation system.

4. Redesigning the Vision System:

- The project acknowledges the need to redesign the vision system, potentially exploring alternative camera setups or computer vision techniques, to further enhance the accuracy and robustness of the sign language recognition process.

5. Enabling Cross-Language Sign Language Recognition:

- Investigating the potential for the system to support multiple sign language dialects or languages can expand the accessibility and usability of the sign language recognition technology.

6. Integrating Text-to-Speech or Speech-to-Text Conversion:

- Combining the sign language recognition system with text-to-speech or speech-to-text conversion capabilities can enable seamless two-way communication, further improving accessibility for individuals with hearing impairments.

7. Real-World Deployment and User Feedback:

- Deploying the sign language recognition system in various real-world settings, such as educational institutions, workplaces, and public spaces, and gathering user feedback can provide valuable insights for further improvements.

8.3 APPLICATIONS

1. Improving Communication and Accessibility for the Deaf and Hard-of-Hearing Community:

- The primary application of this project is to bridge the communication gap between individuals with hearing impairments and those who do not know sign language.
- By translating sign language gestures into text or speech, the system aims to enhance accessibility and enable seamless communication for the deaf and hard-of-hearing community.

2. Deployment in Various Settings:

- The project envisions deploying the sign language recognition system in different settings, such as educational institutions, workplaces, and public spaces, to improve inclusivity and accessibility for individuals with hearing impairments.

3. Assistive Technology for the Deaf and Mute:

- The project's goal is to develop an assistive technology that can help deaf and mute individuals communicate more effectively with those who do not know sign language.

4. Potential for Cross-Language Sign Language Recognition:

- The project suggests the possibility of expanding the system to support multiple sign language dialects or languages, further enhancing its reach and applicability.

5. Integration with Text-to-Speech or Speech-to-Text Conversion:

- Combining the sign language recognition system with text-to-speech or speech-to-text conversion capabilities can enable two-way communication, improving accessibility for individuals with hearing impairments.

9. REFERENCES

1. M. Elmezain, A. Al-Hamadi, and B. Michaelis, Real-Time Capable System for Hand Gesture Recognition Using Hidden Markov Models in Stereo Color Image Sequences, The Journal of W S C G'08, Vol. 16(1), pp. 65-72, 2008

2. Prateek Bhatia (2019) Sign Language Recognition Systems: A Decade Systematic Literature Reviewhttps://www.researchgate.net/pu blication/353571514

3. ADEYANJU, I.A., BELLO, O.O. and ADEGBOYE, M.A.(2021),"Machine learning methods for sign language recognition",https://doi.org/10.1016/j.iswa .2021.200056

4. Md.Zahirul Islama and Md.Shahadat Hossain, "Static Hand Gesture Recognition " 2019 IEEE International Conference paper, doi: 10.1109/ICIEV.2019.8858563.

5. Mukesh Kumar Makwana, "Sign language recognition using convolutional neural networks." 2014.

6. Jaward, M. H., Cheok, M. J., & Omar, Z. (2020). A review of hand gesture and sign language recognition techniques. In 2020 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS) (pp. 1-6). IEEE.

7. Bello, S., Shinde, A., Sharma, K., & Shinde, A. (2023). SIGN LANGUAGE RECOGNITION SYSTEM USING MACHINE LEARNING. IJIRMPS, 11(6).

8. Godoy-Trujillo, P., Flores-Bosmediano, E., Carrascal-Garcia, J., Otero-Potosi, S., Benitez-Pereira, H., & Peluffo-Ordonez, D. H. (2018). Sign Language Recognition Based on Intelligent Glove Using Machine Learning Techniques. In 2018 IEEE 3rd Ecuador Technical Chapters Meeting (ETCM) (pp. 1-6). IEEE.

9. Bantupalli, K., & Xie, Y. (2018). American Sign Language Recognition using Deep Learning and Computer Vision. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 4896-4899). IEEE.
10. Dutta, K. K., & Bellary, S. A. S. (2018). Machine Learning Techniques for Indian Sign Language Recognition. In 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC) (pp. 333-336). IEEE.

11. Kakde, M., Nakrani, M. G., Rawate, A. M. (2016). A Review Paper on Sign Language Recognition System For Deaf And Dumb People using Image Processing. International Journal of Engineering Research & Technology, 5(3), 590-592.

12. Vijay, P. K., Suhas, N. N., Chandrashekhar, C. S., Dhananjay, D. K. (Year). Recent Developments in Sign Language Recognition: A Review. International Journal on Advanced Computer Engineering and Communication Technology, 1(2), xx-xx.

13. Moreira Almeida, S., Guimarães, F., Arturo Ramírez, J. (2014). Extraction in Brazilian Sign Language Recognition based on phonological structure and using RGB-D sensors. Expert Systems with Applications, 41, 7259-7271.

14. Amin, M. A., Yan, H. (2007). Sign Language Finger Alphabet Recognition from Gabor-PCA Representation of Hand Gestures. Machine Learning and Cybernetics, 4, 2218-2223.

15. Shotton, J., Fitzgibbon, A., Cook, M. (2011). Real-time Human Pose Recognition in Parts from Single Depth images. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1297-1304.

16. El zaar, A., Benaya, N., El allati, A. (Year). Sign Language Recognition: High Performance Deep Learning Architecture Based on Convolutional Neural Network. E3S Web of Conferences

17. Kothadiya, D., Bhatt, C., Sapariya, K., Patel, K., Gil-González, A.-B., Corchado, J.M. (2022). Deepsign: Sign Language Detection and Recognition Using Deep Learning. MDPI

18. Rodríguez Villarroel, J. P., Ponce de León Espinoza, N., Arteaga Sabja, W. (Year). Multilayer and Convolutional Neural Networks for Bolivian Sign Language Recognition: An Empirical Evaluation. SciELO.

19. Krismono Triwijoyo, B., Rahmani Karnaen, L. Y., Bumigora, A. A. (Year). Deep Learning Approach For Sign Language Recognition. eprints UAD

20. De Coster, M., Herreweghe, M.V., Dambre, J. (Year). Deep Learning for Sign Language Recognition: Current Techniques, Benchmarks, and Open Issues. ResearchGate