A

Project Report

on

Sign Language Recognition System

submitted for partial fulfillment for the award of

BACHELOR OF TECHNOLOGY

DEGREE

in

Computer Science

By

Vishakha Rana (2000290120192)
Sanskriti Bajpai (2000290120135)
Vidhi (2000290120188)

Under the Supervision of

Prof. Raj Kumar
Assistant Professor & Addl. HOD

Department of Computer Science
KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

May 2024

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature: -

Sanskriti Bajpai	Vidhi	Vishakha Rana
2000290120135	2000290120188	2000290120192

Date: -

CERTIFICATE

This is to certify that Project Report entitled **"Sign Language Recognition System"** which is submitted by **Vidhi (2000290120188), Vishakha Rana (200029020192) and Sanskriti Bajpai (2000290120135)** in partial fulfillment of the requirement for the award of degree B.Tech. in Computer Science from Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

Date:

Supervisor:

Prof. Raj Kumar

Assistant Professor & Addl. HOD

Dept. of Computer Science

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe a special debt of gratitude to **Prof. Raj Kumar**, Department of Computer Science, KIET, Ghaziabad, for his constant support and guidance throughout our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also acknowledge the contribution of **Dr. Ajay Kumar Shrivastava**, Head of the Department of Computer Science, KIET, Ghaziabad, for his full support and assistance during the project's development. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project.

Last but not least, we acknowledge our friends for their contribution to the completion of the project.

ABSTRACT

Sign language plays a crucial role in communication for millions of individuals worldwide who are deaf or hard of hearing. Our project introduces a groundbreaking Sign Language Recognition System (SLRS) designed to bridge the communication gap between sign language users and the general population. Researchers has been using HMM for developing Sign Language Recognition for years. Nowadays, the method that is gaining more popularity in Sign Language Recognition researches is neural network. Neural network is known to be successful in developing speech recognition, and is now being used in Sign Language Recognition. The SLRS boasts real-time capabilities, converting sign language gestures into speech through the application of cutting-edge computer vision and machine learning techniques. The utilization of a Kaggle-sourced dataset is pivotal in training machine learning models to adeptly recognize and interpret a diverse array of sign language gestures. Addressing the intricate challenges inherent in sign language recognition, including variations in signing style, handshapes, and gestures, the project pioneers an innovative approach to overcome these complexities. Our project significantly improves accuracy, surpassing current works in the field, through rigorous testing and fine-tuning of our models. The system's efficiency is underscored through comprehensive evaluation and benchmarking, demonstrating high accuracy and real-time performance. The Sign Language Recognition System for Speech Conversion holds tremendous promise in fostering inclusivity and accessibility, serving as a valuable tool to facilitate improved communication and interaction between sign language users and those less familiar with sign language. This project report significantly contributes to the broader objective of developing technology that enhances the quality of life for individuals with diverse communication needs.

TABLE OF CONTENTS

CONTENT: - **Page No.**

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviations	Full Form
SLRS	Sign Language Recognition System
ASL	American Sign Language
TTS	Text-To-Speech
NLP	Natural Language Processing
ML	Machine Learning
HMM	Hidden Markov Models
CNN	Convolutional Neural Networks
ASR	Automatic Speech Recognition
GUI	Graphical User Interface
APK	Android Application Package

CHAPTER 1
INTRODUCTION

1.1 Introduction to Project

1.1.1 Overview

In the dynamic landscape of assistive technologies, Sign Language Recognition Systems (SLRS) have become instrumental in breaking down communication barriers for the hearing-impaired community. Leveraging the fusion of sophisticated computer vision and machine learning techniques, these systems can interpret sign language gestures in real-time, paving the way for transformative advancements. This report focuses on the evolving field of Sign Language Recognition Systems with a particular emphasis on their integration with speech conversion capabilities. By scrutinizing the technological intricacies, real-world applications, and potential future enhancements of these systems, we aim to provide a comprehensive understanding of their multifaceted impact on communication accessibility. In recent years, the adoption of SLRS has seen a surge, driven by an increasing awareness of the importance of inclusivity and a commitment to ensuring equal opportunities for all. Beyond immediate communication needs, the integration of speech conversion adds an additional layer of versatility to these systems, facilitating a seamless transition between sign language and speech. This has profound implications for various domains, including education, where these systems can enhance the learning experience for both students and educators by providing real-time translations of sign language into spoken language. Moreover, the workplace stands to benefit significantly from the integration of SLRS with speech conversion capabilities. This synergy can foster more inclusive work environments, allowing individuals with hearing impairments to engage more actively in collaborations, meetings, and other professional interactions.

1.1.2. Design

1. Image Recognition: Implemented a hand gesture recognition system using CNN.
2. Speech Conversion: Converted recognized words into speech.
3. System Integration and Evaluation: Integrated all modules into a cohesive system and evaluated its performance in accuracy, speed, and usability.

1.2 Project Category

The SLRS project falls in the category of Android App development using Machine Learning (ML) for recognizing gestures, Computer Vision for processing visual data and interpreting gestures, Natural Language Processing (NLP) for converting recognized gestures into text by interpreting them as letters and combining them into words, and Speech Synthesis for converting the text into spoken words. This comprehensive system aims to assist individuals with speech or hearing impairments by translating gestures into audible speech or enabling hands-free communication in various applications.

1.3 Objectives

1.The primary objective of the system is to facilitate communication for speech-impaired individuals who rely on sign language as their primary means of expression. By recognizing and translating sign language gestures into spoken language and text, the system enables these individuals to communicate more effectively with those who do not understand sign language.

2.The system seeks to enhance accessibility and promote inclusion for individuals with speech impairments. It aims to break down communication barriers, enabling speech-impaired individuals to participate more fully in various aspects of life, including education, employment, healthcare, and social interactions.

3.The project's core objective was to develop a robust sign language recognition system capable of accurately interpreting static hand gestures representing 26 English alphabets (A-Z). This involved training CNNs using various architectures and configurations to achieve high classification accuracy.

4.Another central objective is the conversion of recognized sign language gestures into spoken language. This involved initiating voice media through the system when the input image matches with the predefined dataset.

1.4 Structure of Report

CHAPTER 1: Introduction- In this section, we provide a succinct overview of our sign language recognition project, outlining its objectives and significance.

We highlight the project's aim to develop an efficient sign language recognition system and its potential impact on user convenience.

CHAPTER 2: Literature Review- The literature review delves into existing research and discourse surroundingsign language recognition systems.

CHAPTER 3: Proposed System- This chapter outlines our proposed sign language system, detailing itsfunctionalities and the technologies utilized for its implementation. We elucidate the system's capabilities and how it addresses user requirements.

CHAPTER 4: Requirement Analysis and System Specification- Here, we conduct a feasibility study of our proposed system and provide a detailed software requirement specification. We discuss the chosen SDLC model and its relevance to the project's development.

CHAPTER 5: Implementation- In this section, we present an overview of the languages, tools, and technologiesemployed for implementing the sign language recognition system. We delve into the libraries and algorithms utilized, offering insights into the implementation process and key modules.

CHAPTER 6: Testing and Maintenance- This chapter focuses on testing techniques and methodologies utilizedto ensure the functionality and reliability of our sign language recognition system. We discuss unit testing, integration testing, functional testing, usability testing, and performance testing, along with details ofthe test environment.

CHAPTER 7: Results and Discussions- Here, we provide a summary of the various modules comprising the sign language recognition system and discuss the outcomes of our project. We analyze the results, interpret theirimplications, and offer insights into the findings.

CHAPTER 8: Conclusion- In the conclusion, we encapsulate the project's objectives, achievements, and contributions. We reflect on key findings, discuss potential future research directions, and underscorethe significance of our work.

References- The references section includes citations of relevant research papers, articles, and sources used throughout the report, ensuring transparency and academic integrity.

CHAPTER 2

LITERATURE REVIEW

2.1 Literature Review

The literature survey presents a comprehensive overview of the current landscape in sign language recognition systems, featuring diverse methodologies and technological advancements. Starner and Pentland's foundational work ([1]) introduces a real-time American Sign Language (ASL) recognition system using hidden Markov models (HMMs), establishing the groundwork for subsequent research. Jain and Bhattacharjee's review ([2]) delves into the broader realm of sign language recognition, discussing various methods and highlighting the challenges.

Shah and Bhatt's study ([3]) explores the real-time application of convolutional neural networks (CNNs) for ASL recognition, showcasing the evolution from traditional HMMs to deep learning approaches. Stein, McKee, and Raab ([4]) take a different approach, presenting a wearable sign language recognition system utilizing accelerometers, offering a unique perspective on gesture detection.

Mardia et al.'s comprehensive review ([5]) captures the broader landscape of machine learning applications in sign language recognition, providing insights into various techniques employed across the field. Huenerfauth's work ([6]) shifts the focus to avatar-based sign language animation, emphasizing the significance of simplicity for effective communication. Agrawal and Bohara ([7]) contribute to the practical aspect of sign language recognition, addressing its application in assisting individuals with hearing impairments. Mak and Mak ([8]) introduce a novel dimension with brain-computer interfaces, showcasing the potential of neurotechnology in advancing sign language recognition systems.Green and Johnson's study ([9]) concentrates on refining the timing accuracy in virtual environments, addressing an essential aspect of user experience. Li and Wang ([10]) leverage depth sensors and deep learning for real-time sign language recognition, introducing a novel approach to gesture detection. Mayol-Cuevas and Kontoudis ([11]) explore the integration of Kinect sensors for sign language recognition, highlighting technological advancements in gesture recognition.

Rath and Meng's work ([12]) contributes by addressing the challenges of sign language recognition in real-world scenarios, providing benchmark results.Starner and Pentland's earlier work ([13]) on visual recognition using HMMs lays the foundation for subsequent advancements in ASL recognition. Wang and Hu ([14]) introduce continuous sign language recognition using CNNs, showcasing the evolution of deep learning techniques. Zhang and Tan ([15]) contribute a real-time continuous sign language recognition system using HMMs applied to video sequences.

Teixeira and Rodrigues ([16]) offer a systematic literature review, summarizing key findings and trends in sign language recognition research. Verma and Sathyadevan ([17]) provided a comprehensive review covering various methodologies and applications in sign language recognition systems. Kamphuis and ten Holt ([18]) introduce the SignSpeak recognition system, emphasizing the system's role in enabling communication through sign language gestures. The visual representation of ASL letters and digits ([19]) adds a valuable reference point for understanding the linguistic components involved.

Finally, Limaye et al.'s recent work ([20]) introduces a sign language recognition system based on CNNs with customization, showcasing the ongoing advancements and relevance of contemporary research in the field.

This literature survey encapsulates the rich tapestry of research in sign language recognition systems, offering insights into the evolution of methodologies, applications, and technological advancements in this critical area of assistive technology.

2.2 Research Gaps

- Limited Vocabulary Coverage: The system focuses on recognizing a limited set of signs. Research could explore expanding the vocabulary to include more signs and gestures, especially those relevant to specific domains or cultures.
- Adaptation to Variability: Sign language involves significant variability in hand shapes, movements, and speed. Research could explore methods to enhance the robustness of systems to account for this variability, including techniques for handling different signing styles and variations in lighting and background conditions.
- Multimodal Integration: Incorporating other modalities such as facial expressions and body

movements can enhance the accuracy and expressiveness of SLRS.

- Cross-cultural Adaptation: Sign languages vary significantly across different regions and communities. Research could explore methods for adapting recognition models to account for these variations, including dialectical differences and culturally specific gestures.
- Long-term User Adaptation: As users interact with the app over time, their signing patterns may evolve or change. Research could investigate adaptive learning techniques to continuously improve recognition accuracy based on user feedback and usage data.
- Domain-specific Recognition: Certain domains, such as medical or legal settings, may have specialized sign language vocabularies and gestures. Research could focus on domain-specific adaptation techniques to improve accuracy and usability in these contexts.
- Low-resource Settings: In regions with limited access to resources such as high-speed internet or powerful computing devices, designing lightweight and resource-efficient recognition models is essential. Research could explore techniques for developing efficient models that can run effectively on low-resource Android devices.
- User Feedback Integration: Incorporating user feedback into the recognition system's training and adaptation processes can help improve its performance and usability. Research could investigate methods for effectively collecting, analyzing, and integrating user feedback into the system's learning loop.

2.3 Problem Formulation

The aim of this project is to develop a ML-based system that can recognize hand gestures, identify the corresponding letters represented by these gestures, bind these letters into words, and finally convert these words into speech. This system is intended to assist individuals with speech or hearing impairments by providing a means of communication that does not rely on traditional speech. The key challenges include developing robust gesture recognition algorithms capable of accurately interpreting a wide range of gestures, designing efficient methods for converting recognized gestures into text, and implementing a reliable speech synthesis module to verbalize the converted text. Additionally, the system must be user-friendly and capable of real-time operation to be practical for everyday use.

CHAPTER 3
PROPOSED SYSTEM

3.1 Proposed System

Creating sign language recognition system that converts signs into speech requires a robust framework encompassing several critical stages. Data collection forms the cornerstone, necessitating the assembly of a diverse dataset featuring a wide array of sign language gestures. This dataset undergoes meticulous preprocessing to eliminate noise and standardize hand positions and movements, ensuring optimal recognition outcomes.

Feature extraction is pivotal, involving the identification and isolation of key elements within the sign language data, that is, hand shapes. This feature serves as the foundation for training machine learning or deep learning model, which are essential for accurately recognizing and interpreting sign language gestures.

Integration of a speech synthesis module adds another layer of complexity, transforming recognized signs into spoken language through advanced text-to-speech (TTS) technology. This component seamlessly integrates with SLRS to deliver coherent and natural-sounding speech output.

User interface design is paramount for ensuring accessibility and usability, with intuitive interfaces tailored to various platforms such as mobile applications, web browsers, or wearable devices. Extensive testing and evaluation are imperative to validate the system's accuracy, robustness, and compatibility across different sign languages and environmental conditions.

Incorporating accessibility features, such as support for multiple sign languages and compatibility with assistive technologies, broadens the system's reach and effectiveness in facilitating communication for individuals with hearing impairments.

Finally, the deployment and maintenance phase involve the rollout of the system to users, accompanied by regular updates and enhancements to address emerging needs and improve overall performance. The working of the proposed system can be explained through the architectural diagram in Fig.3.1

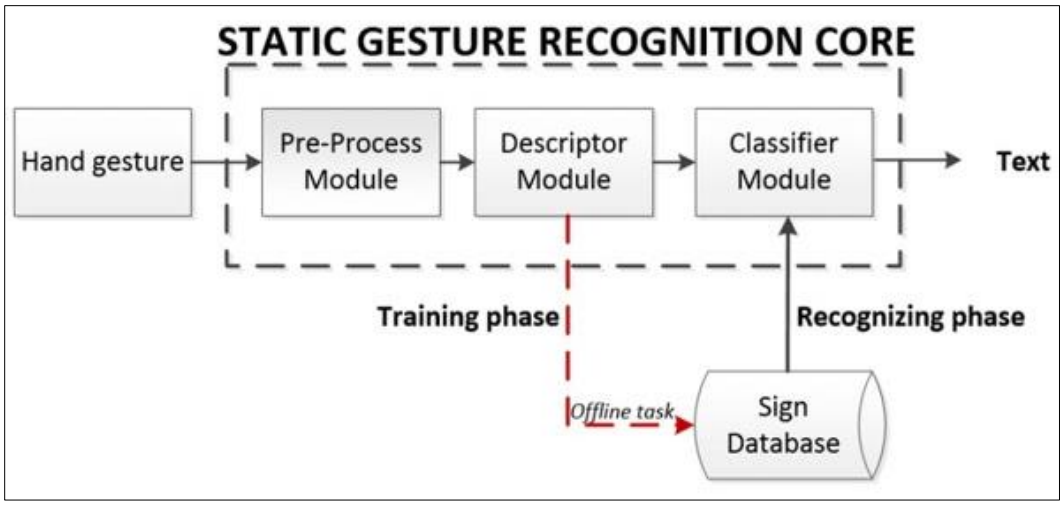


Fig.3.1 Architectural Diagram of the Proposed System

The block diagram of SLRS, in Fig.3.2, comprises four main blocks: user interface, front-end processing, back-end processing, and deep learning model. Users interact via camera input or uploaded media, processed through segmentation, feature extraction, and normalization. Back-end processing handles logic, classification etc. while the deep learning model, like CNNs or RNNs, learns from extensive datasets to recognize gestures and convert them into text or speech output.

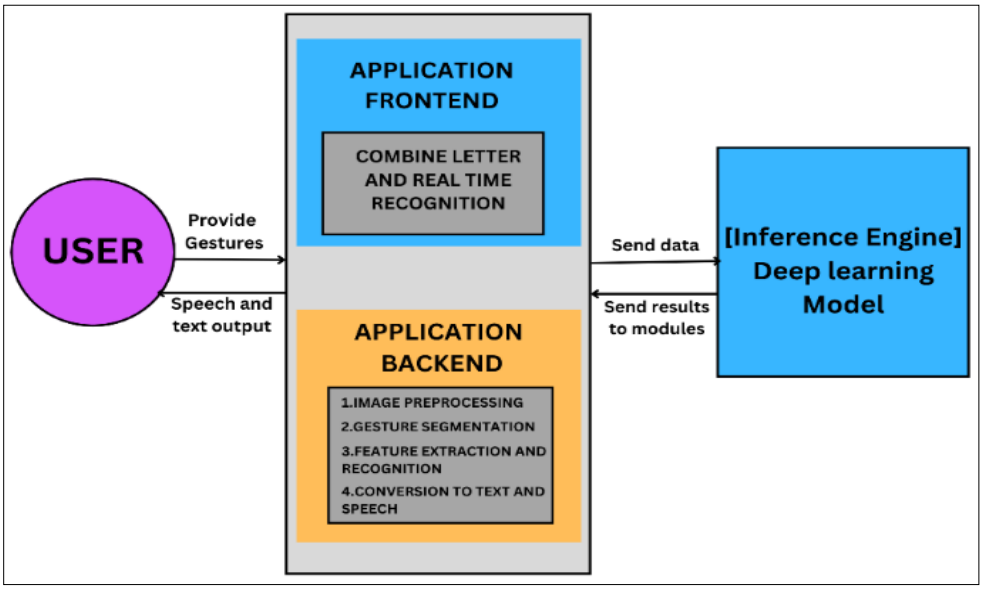


Fig.3.2 Block Diagram of the Proposed System

3.2 Unique Features of the System

1.Real-Time Recognition: Our system can recognize sign language gestures in real time, making it practical for use in live communication scenarios.

2.Multi-Gesture Recognition: System is able to recognize multiple gestures sequentially, enabling the formation of complete words and sentences.

3.Combination of Hand Gestures: The system combines individual hand gestures to form meaningful words which showcases its capability to understand context and syntax in sign language.

4.Interactive User Interface: Describe the user-friendly interface of your system, including features like an 'add' button to combine recognized gestures, a 'clear' button for erasing text, and a 'read' button to convert text to speech.

5.Speech Output: SLRS not only recognize sign language but also converts the recognized text into speech, enhancing accessibility for users who may not understand sign language.

6.Accuracy and Performance: The project has achieved a commendable accuracy rate of 93% in recognizing a diverse set of sign language gestures. Continuous refinement of the algorithm and regular updates contribute to maintaining and improving this accuracy over time.

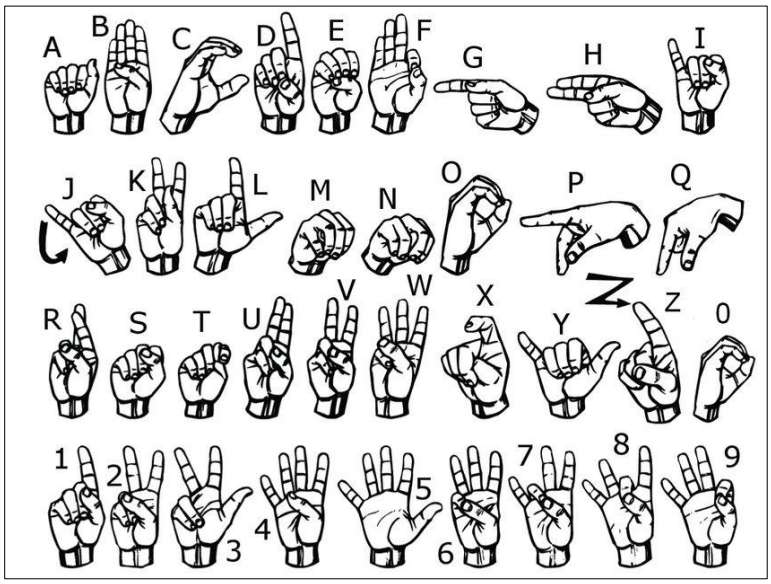


Fig.3.3 American Sign Language

CHAPTER 4
REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION

4.1 Feasibility Study

The feasibility study helps us determine whether or not we should proceed with our project. Itis essential to evaluate costs and benefits of the proposed system. Three types of feasibility studies are taken into consideration.

Technical Feasibility:

The system needs a camera or sensor for image input, a processor for real-time processing, and audio output for speech conversion. Integrating the various components and testing them for accuracy, speed, and reliability will be crucial.

Economic Feasibility:

The development cost includes hardware, software, and personnel costs. Cost of Deployment includes deploying the system that involves additional costs such as installation, training, and maintenance. A cost-benefit analysis was conducted to determine if the benefits of the system outweigh the costs.

Operational Feasibility:

The system should be user-friendly and intuitive for the target users, such as individuals who are deaf or hard of hearing. The system should be compatible with existing technologies and infrastructure. Adequate provisions should be made for ongoing maintenance to ensure the system remains functional and up to date.

4.2 Software Requirement Specification

4.2.1 Data Requirement

1. A dataset containing a variety of gestures representing letters and words is needed.
2. A mapping of letters to words is required to bind the recognized letters into meaningful word.
3. A language model or dictionary that contains a vocabulary of words is needed to verify and correct the recognized gestures and their corresponding words.

4.2.2 Functional Requirements

(a) Gesture Recognition Module:

- Must accurately detect and classify a diverse range of sign language gestures.
- Should operate in real-time to provide instantaneous feedback.
- Must be adaptable to different lighting conditions for robust performance.

(b) Letter Binding Module:

- Must associate each recognized gesture with the correct corresponding letter.
- Should account for variations in signing styles and regional differences.

(c) Speech Conversion Module:

- Must convert recognized gestures into clear and coherent audible speech.
- Should support multiple languages for enhanced accessibility.

(d) Real-time Processing:

- Must process gestures and convert them into speech with minimal latency.
- Should handle concurrent user interactions without compromising performance.

(e) User Interface:

- Must provide an intuitive and user-friendly interface for seamless interaction.
- Should display feedback on recognized gestures and converted speech.

4.2.3 Performance Requirements

1. **Response Time:** The system should provide prompt responses to user inputs, aiming for minimal latency inprocessing and execution.
2. **Accuracy:** Speech recognition and interpretation should achieve high accuracy levels to correctly understand user commands and queries.
3. **Resource Efficiency:** The system should utilize system resources efficiently to ensure smooth operation withoutexcessive CPU or memory usage.

4.2.4 Maintainability Requirements

(a) Modularity: The system architecture should be modular, allowing for easy maintenance, updates, and scalability.

(b) Documentation: Comprehensive documentation should be provided for the project to facilitate understanding, troubleshooting,and future enhancements.

(c) Error Handling: Robust error-handling mechanisms should be implemented to identify and address issues,ensuring system reliability and stability.

4.2.5 Security Requirements

(a) Data Privacy: User data, including voice inputs and personal information, should be handled securely and incompliance with privacy regulations.

(b) Access Control: Access to sensitive functionalities or data within the system should be appropriately restricted toauthorized users or roles.

(c) Secure Communication: Any communication between the sign language recognition system and external services or APIs should beencrypted to prevent unauthorized access or data interception.

4.3 SDLC Model USED - Iterative Model:

The Iterative model is a software development approach where the project is divided into small increments or iterations, each building upon the previous one. This method is highly adaptable and enables the development team to make continuous improvements and adjustments based on feedback and evolving requirements. Here are some key characteristics of the Iterative model:

(a) Incremental Development: The project progresses through a series of iterations, with each iteration adding new features, enhancements, or improvements to the product. Each iteration results in a working version of the software, allowing the team to deliver functional parts of the system early and often. This approach contrasts with traditional models where the entire system is delivered at the end of the development cycle. By breaking the project into smaller, manageable pieces, the team can focus on developing and testing each component thoroughly.

(b) Feedback Loop: Iterative development incorporates feedback from users and team members at the end of each iteration. This feedback loop is crucial for guiding the direction of development and ensuring alignment with user needs and requirements. After each iteration, the team reviews progress and gathers input from various sources to make informed decisions about the next steps. This process helps identify potential issues early, allowing the team to address them before they become significant problems.

(c) Flexible Planning: Iterative projects have a flexible planning approach, with requirements and priorities evolving as the project progresses. This flexibility allows for adjustments based on feedback, lessons learned, and changing business needs. Unlike rigid project management methods, the Iterative model adapts to new information and shifting priorities, ensuring that the project remains relevant and aligned with business objectives. This dynamic planning process helps the team respond quickly to changes and deliver a product that meets current needs.

(d) Progressive Refinement: Each iteration builds upon the previous one, gradually refining the product and adding new features or improvements. This incremental approach enables early delivery of value while maintaining focus on quality and usability. By continuously refining and enhancing the product, the team can ensure that each version is better than the last, incorporating user feedback and addressing any deficiencies. This method helps maintain a high standard of quality throughout the development process.

(e) Continuous Validation: Throughout the iterative development process, the product is continuously validated against user needs, expectations, and acceptance criteria. This ongoing validation ensures that the product meets quality standards and delivers value to stakeholders. Regular testing and validation activities help identify defects and areas for improvement, ensuring that the final product is robust and reliable.

(f) Parallel Development: Iterative development often involves parallel workstreams, where multiple iterations may be in progress simultaneously. This parallelism allows for the efficient use of resources and accelerates the pace of development. By working on different components or features concurrently, the team can deliver more functionality in less time, increasing overall productivity.

4.4 System Design

4.4.1 Detail Design

The design of a SLRS involves several components and processes. The key elements in the system are:

1. Input Module:

- Hand Gesture Acquisition: This module is responsible for capturing hand gestures using a camera or sensor. The video stream is processed to extract relevant hand gestures necessary for recognition. The acquisition process ensures that the input sign language hand gestures are clear and well-defined, providing a solid foundation for subsequent processing stages.

2. Preprocessing:

-Image Processing: The captured hand gestures undergo various preprocessing techniques to enhance the quality of the input. Segmentation isolates the hand from the background, ensuring that the recognition algorithms can focus on the relevant features.

-Feature Extraction: From the preprocessed images, relevant features are extracted, including hand shape, finger positions, and hand orientations. These features are crucial for the recognition algorithms as they provide the necessary data to identify specific gestures accurately.

3. Recognition Algorithms:

- Gesture Segmentation: This process involves segmenting the continuous stream of sign language hand gestures into discrete units, such as individual signs or gestures. Machine Learning models are employed to accurately segment the gestures, ensuring that each gesture is distinctly recognized without overlap.

-Feature Recognition: Advanced machine learning algorithms, including convolutional neural network and deep learning models, are applied to recognize the extracted features. These algorithms classify the features into corresponding sign language symbols or hand gestures with high accuracy.

4. Text Output:

-Text Generation: The recognized sign language symbols or hand gestures are converted into text representation. This process involves using mapping tables that link specific gestures to their corresponding textual counterparts, ensuring that the generated text accurately reflects the intended message.

-Text Processing: The generated text undergoes post-processing to correct any errors and enhance readability. This step ensures that the text output is user-friendly and accurately represents the recognized gestures.

5. Speech Output:

-Speech Synthesis: The text generated from the recognized gestures is converted into speech output using text-to-speech (TTS) synthesis techniques. This process ensures that the system can provide auditory feedback, making it accessible to a broader audience.

6. User Interface:

-Graphical User Interface (GUI): An intuitive and user-friendly interface is designed for users to interact with the system. The graphical user interface allows users to visualize input gestures, view the recognized text, and listen to the speech output, providing a seamless user experience.

7. Integration and Deployment:

- Integration with Applications: The sign language recognition system is integrated with various applications or platforms, such as Android mobile apps or assistive devices. This integration ensures that the system can be used in real-world scenarios, enhancing its practicality.

-Deployment Considerations: Several factors are considered during deployment, including scalability, reliability, and accessibility. The system is designed to handle real-time processing with minimal latency, ensuring compatibility with various devices and meeting user accessibility requirements. This comprehensive approach ensures that the sign language recognition system is robust and reliable in practical use cases.

4.4.1 Data Flow Diagrams for the proposed system

4.4.2.1. DFD Level 0

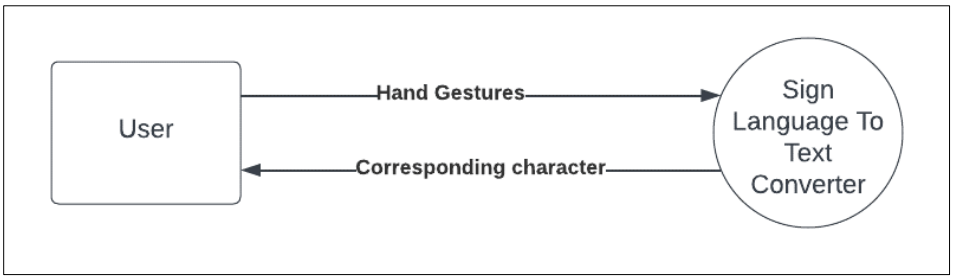


Fig.4.1 DFD Level 0 for SLRS

4.4.2.2. DFD level 1

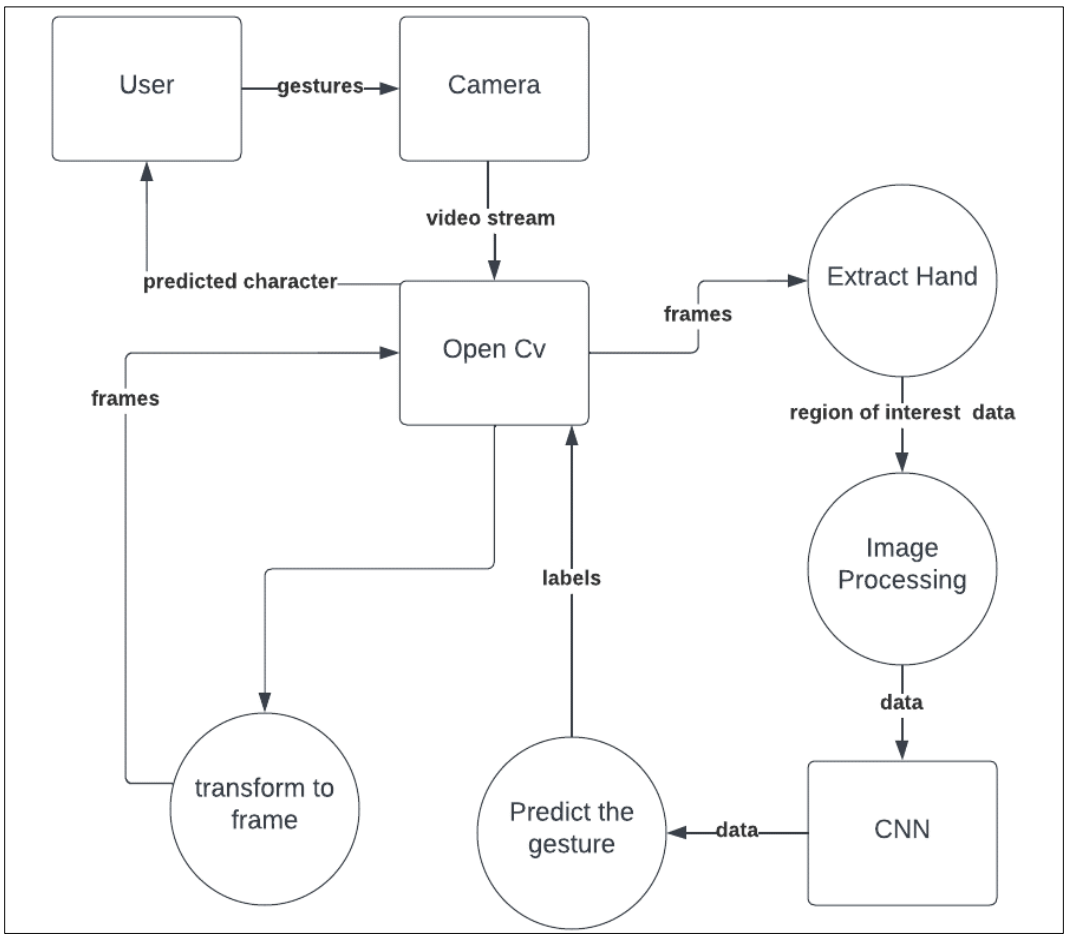


Fig4.2 DFD Level 1 for SLRS

4.4.2 Use Case Diagram for the proposed system

Fig.4.3. Use Case Diagram for SLRS

4.4.3 Flow Chart for the proposed system

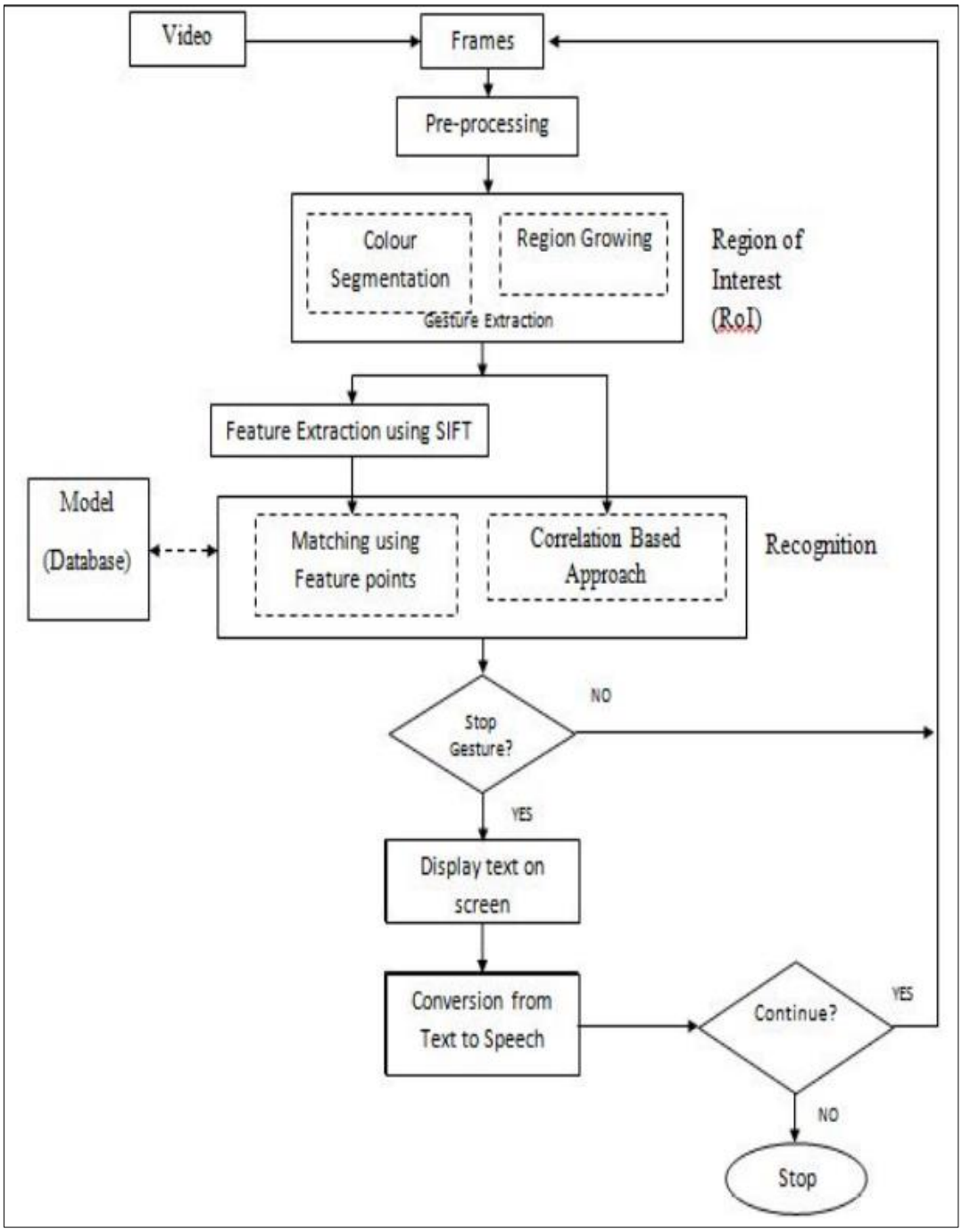


Fig.4.4 Flow Chart for SLRS

4.4.5 Class Diagram for the proposed system

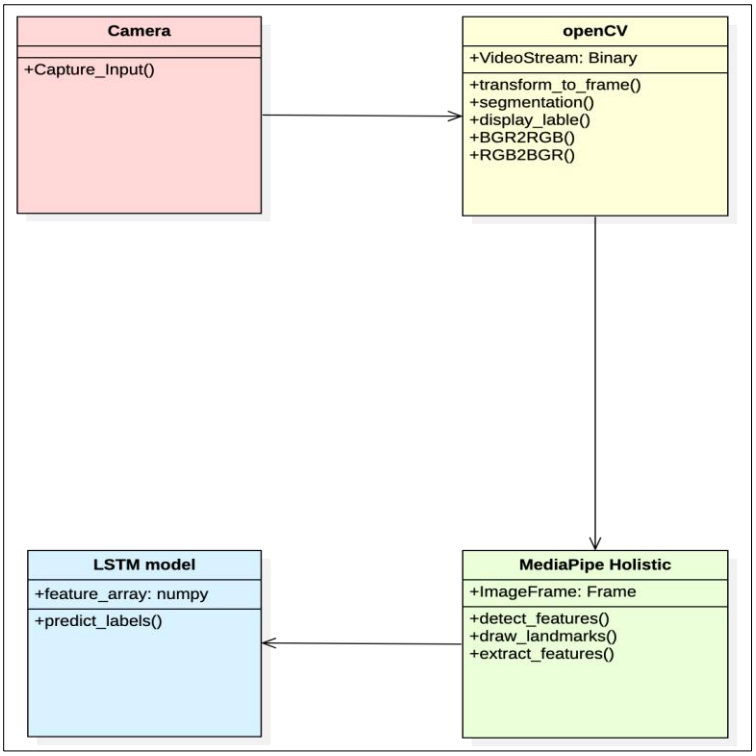


Fig.4.5 Class Diagram for SLRS

4.4.6 Activity Diagram for the proposed system

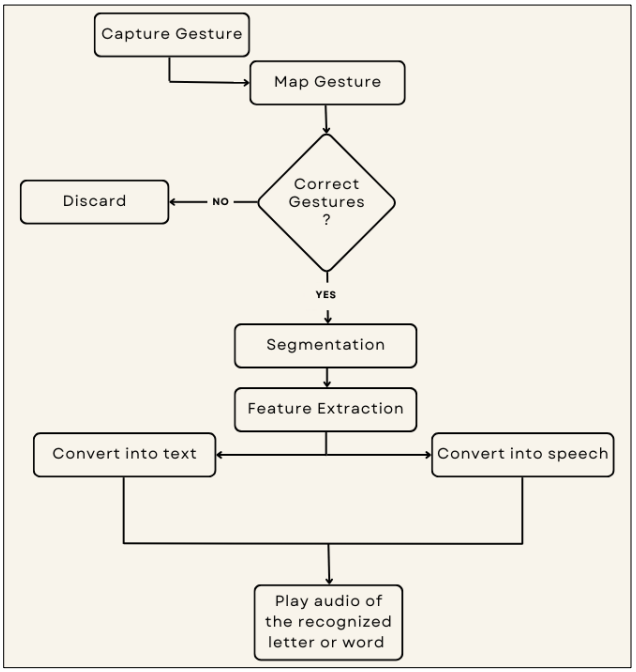


Fig.4.6 Activity Diagram for SLRS

4.4.7 Sequence Diagram for the proposed system

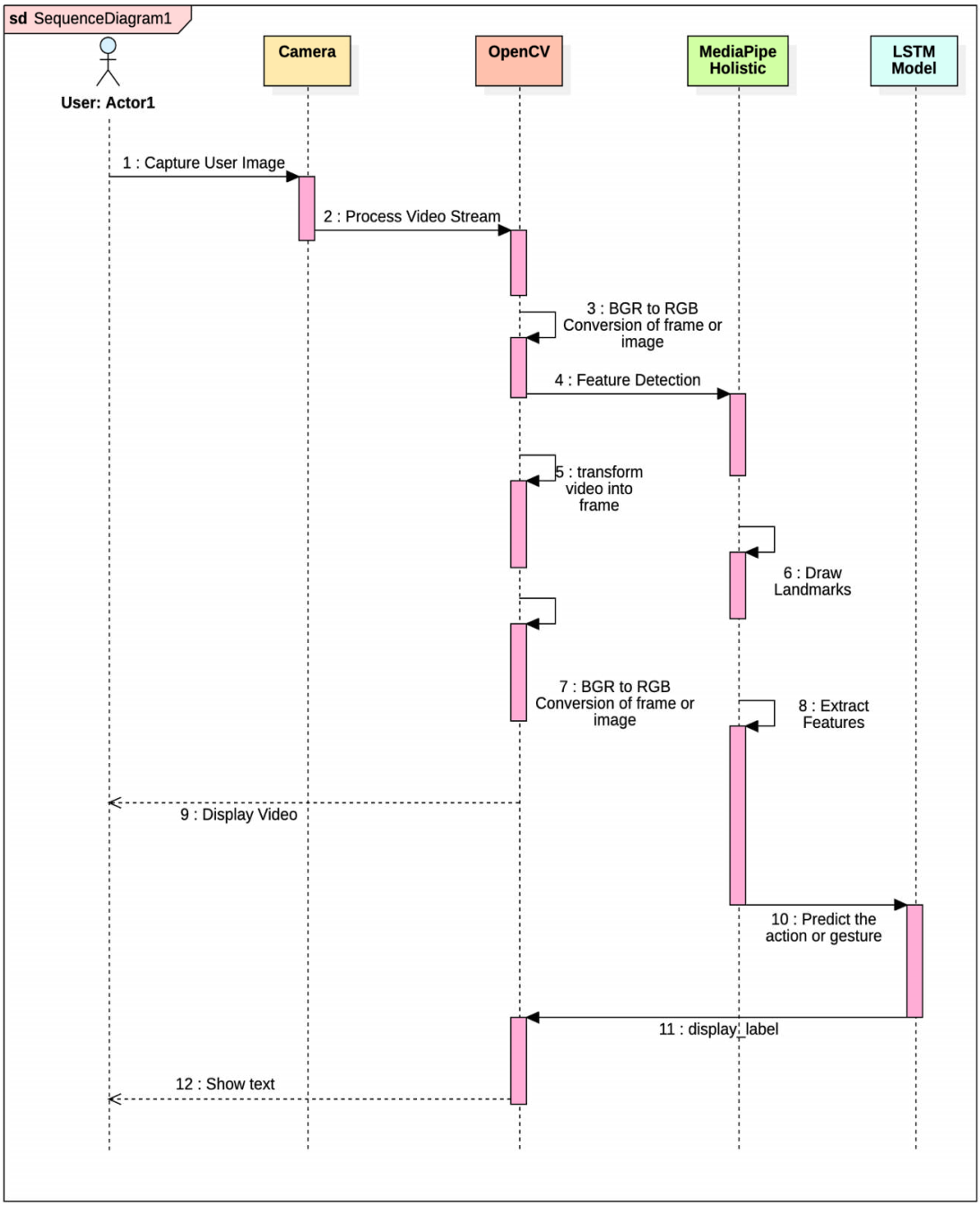


Fig.4.7 Sequence Diagram

CHAPTER 5
IMPLEMENTATION

5.1 Introduction to Tools and Technologies used:

The Android application is designed with two fundamental features that play a crucial role in enhancing communication for individuals using sign language: real-time gesture recognition and letter synthesis. These features are meticulously crafted to seamlessly interpret hand gestures into text and then convert that text into spoken words, facilitating efficient communication in sign language.

1. Camera Input: The app leverages the capabilities of the smartphone's camera to capture video frames of hand gestures. This process is made possible through the integration of OpenCV, a widely used computer vision library. OpenCV enables the app to access the camera feed and process each frame in real time. Additionally, NumPy is employed for efficient array processing of these captured frames. NumPy's array operations enhance the speed and accuracy of processing, making it an ideal choice for handling the data-intensive nature of video frames.

2. Image Processing: Once the frames are captured, they undergo a series of image processing techniques facilitated by NumPy arrays. These techniques include background subtraction, which helps isolate the hand gestures from the background noise, contour detection to identify the shape and movement of the hand, and hand region segmentation for precise analysis of the gestures. These processes collectively contribute to accurate gesture recognition.

3. Gesture Recognition: The heart of the app's functionality lies in its gesture recognition capabilities powered by TensorFlow, a cutting-edge machine learning framework. TensorFlow allows the implementation of sophisticated deep learning models, particularly convolutional neural networks (CNNs), which are adept at understanding complex patterns in visual data. Techniques like Global Average Pooling (GAP) are employed to extract meaningful features from the CNN layers, while dropout layers help in regularizing the network and preventing overfitting.

4. Text Conversion: Upon successful recognition of a gesture, the app utilizes NumPy for efficient handling of text data. The recognized gestures are converted into text format, enabling users to see the corresponding textual representation of their gestures. Natural Language Processing (NLP) algorithms further enhance this process by interpreting the gestures as meaningful words or sentences, providing a coherent and understandable output.

5. Speech Synthesis: The text generated from the recognized gestures is then converted into speech using Android's built-in Text-to-Speech (TTS) functionality. This feature enables the app to audibly convey the interpreted gestures, ensuring a comprehensive communication experience. Mean Absolute Error (MAE) serves as a metric to evaluate the accuracy of the speech synthesis process, ensuring that the spoken output closely matches the intended gestures.

6. User Interface: The user interface of the app is designed with utmost care to provide a seamless and intuitive experience. It is developed using XML for layout design and Android UI components for interactive elements. The interface prominently displays recognized gestures, converted text, and synthesized speech, allowing users to easily navigate and interact with the app's functionalities.

7. Integration and Deployment: The various components of the app, including the image processing modules, TensorFlow models for gesture recognition, and speech synthesis functionalities, are seamlessly integrated to form a cohesive application. The final product is then packaged as an Android application package (APK) and deployed for installation on Android devices. This deployment process ensures that the app is accessible to users across different Android platforms, enhancing its reach and usability.

8. Testing and Optimization: Before deployment, the app undergoes rigorous testing to ensure its functionality and performance. Testing methodologies include unit testing, integration testing, and real-world scenario testing to validate accurate
gesture recognition and speech synthesis. Optimization techniques are also applied, leveraging TensorFlow's capabilities and integrating EfficientNetB0, a lightweight

CNN architecture, for efficient model inference and performance tuning. These optimizations enhance the app's responsiveness and overall user experience.

5.2 Dataset Description:

The sign language image dataset sourced from Kaggle is a substantial resource for researchers and developers in the field of sign language recognition. It consists of an impressive collection of nearly 13,000 RGB images, each representing a hand gesture corresponding to the alphabets A-Z. What makes this dataset particularly valuable is the sheer volume of data it encompasses, with an average of 400-500 images available for each gesture. This extensive coverage ensures that the dataset captures a diverse range of hand poses and variations, contributing to the robustness and accuracy of machine learning models trained on this data.

In terms of data size, the dataset occupies a total file size of 351 megabytes (MB), with individual images averaging around 40 kilobytes (KB) each. This relatively compact size per image allows for efficient storage and processing, even when dealing with such a large dataset.

The abundance of data provided by this dataset makes it an ideal choice for training, testing, and evaluating ML algorithms tailored for sign language recognition tasks. By leveraging this comprehensive dataset, we could develop and fine-tune the model that accurately interpret and translate sign language gestures into meaningful text or speech.

Fig.5.1 Sample Images of Dataset

CHAPTER 6
TESTING AND MAINTENANCE

6.1 Testing Techniques & Test Cases Used

We've adopted an iterative testing approach to ensure the functionality and reliability of our project. This approach involves testing the project in incremental steps, beginning with the evaluation of individual modules to ensure they function correctly in isolation. Subsequently, we assess how these modules interact and work together seamlessly.

Throughout the development process, we continuously conduct tests as we implement changes and introduce new features. This ongoing testing strategy ensures that our project maintains optimal performance and functionality, even after modifications have been made.

6.1.1. Unit Testing for the proposed system

Unit testing involves individual components or modules of SLRS in isolation. Each unit, such as a function or method, is tested independently to ensure its correctness and functionality.

Table 6.1 Unit Testing for SLRS

Test Case	Description
1	Test the speech recognition module with clear audio input and verify accurate transcription.
2	Test the speech recognition module with noisy audio input and verify robustness against background noise.
3	Integrate speech recognition module with command execution module and test with simple commands to verify correct execution.
4	Test speech recognition module with various speech speeds and verify accurate transcription without loss of information.

6.1.2 Integration Testing for the proposed system

Integration testing is a crucial phase in the development of SLRS as it verifies the interactions and interfaces between different components or modules. This testing process focuses on ensuring seamless communication and efficient data exchange between these modules, thereby enhancing the overall functionality and reliability of the system.

Table 6.2 Integration Testing for SLRS

Test Case	Description
1	Integrate speech recognition module with command execution module and test with simple commands to verify correct execution.
2	Test integration with complex commands involving multiple parameters to verify correct parsing and execution.
3	Test compatibility with different versions of Android and third-party libraries.
4	Verify that data is passed correctly between modules and that they interact as expected.

6.1.3 Functional Testing for the proposed system

Functional testing plays a pivotal role in ensuring the robustness and accuracy of a Sign Language Recognition System (SLRS) by verifying its responsiveness to user commands and adherence to functional requirements. This testing process involves evaluating the system's functionality to ensure smooth execution and reliable performance across various scenarios. By conducting comprehensive functional testing, sign language recognition system can be validated for its overall working, ensuring that it meets the intended functional objectives, delivers accurate results, and provides a seamless user experience.

Table 6.3 Functional Testing for SLRS

Test Case	Description
1	Test the functionality of the app by providing different hand gestures and verifying that the corresponding text and speech outputs are correct.
2	Test for error handling, ensuring the app gracefully handles unexpected inputs or situations.
3	Verify that the app behaves as expected in different lighting conditions and environments.
4	Test the functionality of system with blurred input images.

6.1.4 Usability Testing for the proposed system

Usability testing evaluates the interface and interaction design to ensure it isintuitive and user-friendly. Test cases focus on assessing the ease of use and overall user experience.

Table 6.4 Usability Testing for SLRS

Test Case	Description
1	Test the user interface for intuitiveness and ease of use, ensuring that users can easily perform hand gestures and understand the feedback.
2	Conduct user testing sessions to gather feedback on the overall user experience and identify areas for improvement.
3	Test accessibility features to ensure the app is usable by people with disabilities.

6.1.5 Performance Testing for the proposed system

Performance testing evaluates the responsiveness and scalability of SLRS underdifferent conditions. Test cases are designed to measure response times, throughput, and resource utilization.

Table 6.5 Performance Testing for SLRS

Test Case	Description
1	Test the app's response time by measuring the time taken to recognize a hand gesture and produce the corresponding text and speech outputs.
2	Test the app's memory usage and resource consumption to ensure it performs efficiently without draining the device's battery or causing lag.
3	Conduct stress testing by simulating heavy usage scenarios to identify potential performance bottlenecks and optimize the app's performance accordingly.

6.2 Test Environment

The following **software's** are required in addition to client-specific software.

- Windows 10 and above preferred
- VSCode 2022 or above preferred
- The camera should be working properly.

CHAPTER 7
RESULTS AND DISCUSSIONS

7.1 Presentation of Results:

Project Output Screenshots:-

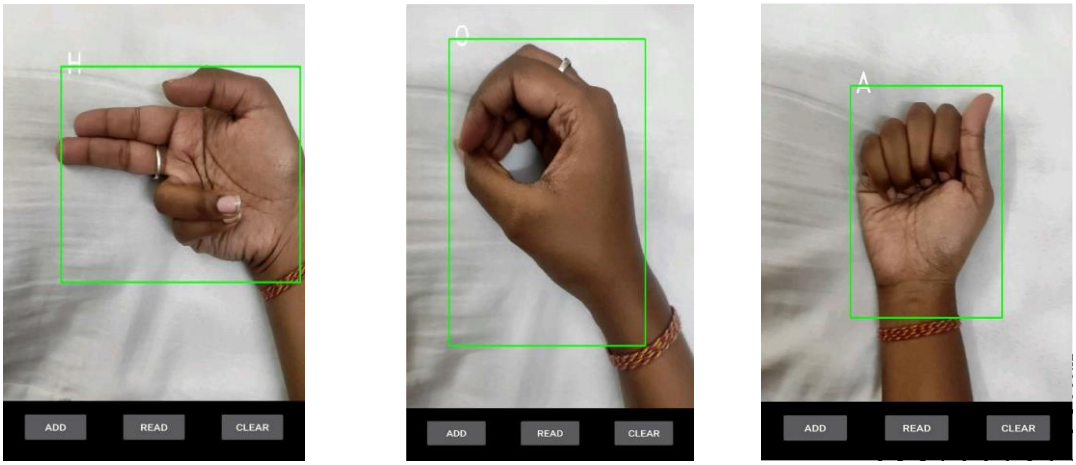


Fig. 7.1 Snapshots of Android app execution

Training History (Selected Epochs): -

```
Epoch 1/100
303/303 [==============================] - 419s 1s/step - loss: 2.8502 - mae: 2.8502 - val_loss: 4.2119 - val_mae: 4.2119 - lr: 0.0010
Epoch 2/100
303/303 [==============================] - 411s 1s/step - loss: 1.3188 - mae: 1.3188 - val_loss: 2.2223 - val_mae: 2.2223 - lr: 0.0010
Epoch 3/100
303/303 [==============================] - 413s 1s/step - loss: 0.9252 - mae: 0.9252 - val_loss: 0.8191 - val_mae: 0.8191 - lr: 0.0010
Epoch 4/100
303/303 [==============================] - 401s 1s/step - loss: 0.7690 - mae: 0.7690 - val_loss: 0.5485 - val_mae: 0.5485 - lr: 0.0010
Epoch 5/100
303/303 [==============================] - 408s 1s/step - loss: 0.6574 - mae: 0.6574 - val_loss: 0.4997 - val_mae: 0.4997 - lr: 0.0010
Epoch 6/100
303/303 [==============================] - 415s 1s/step - loss: 0.6190 - mae: 0.6190 - val_loss: 0.5802 - val_mae: 0.5802 - lr: 0.0010
Epoch 7/100
303/303 [==============================] - 422s 1s/step - loss: 0.5681 - mae: 0.5681 - val_loss: 0.6161 - val_mae: 0.6161 - lr: 0.0010
Epoch 8/100
303/303 [==============================] - 453s 1s/step - loss: 0.5753 - mae: 0.5753 - val_loss: 0.5173 - val_mae: 0.5173 - lr: 0.0010
Epoch 9/100
303/303 [==============================] - 386s 1s/step - loss: 0.4995 - mae: 0.4995 - val_loss: 0.7425 - val_mae: 0.7425 - lr: 0.0010
Epoch 10/100
303/303 [==============================] - ETA: 0s - loss: 0.4937 - mae: 0.4937
Epoch 10: ReduceLROnPlateau reducing learning rate to 0.0009000000427477062.
303/303 [==============================] - 377s 1s/step - loss: 0.4937 - mae: 0.4937 - val_loss: 0.5801 - val_mae: 0.5801 - lr: 0.0010
```

Fig.7.2 Snapshot of Training History

- Epoch 1/100: Loss=2.2752, MAE=2.2752, Validation Loss=2.5211, Validation MAE=2.5211
- Epoch 10/100: Loss=0.3444, MAE=0.3444, Validation Loss=0.2729, Validation MAE=0.2729
- Epoch 20/100: Loss=0.2416, MAE=0.2416, Validation Loss=0.1503, Validation MAE=0.1503
- Epoch 30/100: Loss=0.2488, MAE=0.2488, Validation Loss=0.3276, Validation MAE=0.3276
- Epoch 40/100: Loss=0.1913, MAE=0.1913, Validation Loss=0.2638, Validation MAE=0.2638
- Epoch 50/100: Loss=0.1979, MAE=0.1979, Validation Loss=0.1792, Validation MAE=0.1792
- Epoch 60/100: Loss=0.1591, MAE=0.1591, Validation Loss=0.0991, Validation MAE=0.0991
- Epoch 70/100: Loss=0.1903, MAE=0.1903, Validation Loss=0.1535, Validation MAE=0.1535
- Epoch 80/100: Loss=0.1556, MAE=0.1556, Validation Loss=0.2114, Validation MAE=0.2114
- Epoch 90/100: Loss=0.1353, MAE=0.1353, Validation Loss=0.0828, Validation MAE=0.0828

The graph presented in Figure 7.3 illustrates the correlation between loss and epochs, revealing a discernible trend of decreasing loss values with an increase in the number of epochs. Initially, at Epoch 1 out of 100, the loss registers a relatively high value of 2.2752, reflecting the model's initial learning phase. However, as training progresses, the loss steadily declines across subsequent epochs. By the time Epoch 90 out of 100 is reached, the loss has notably decreased to a mere 0.1353. This significant reduction in loss signifies the model's enhanced capability to generate more accurate predictions as it undergoes further training epochs, indicating a positive learning trajectory.

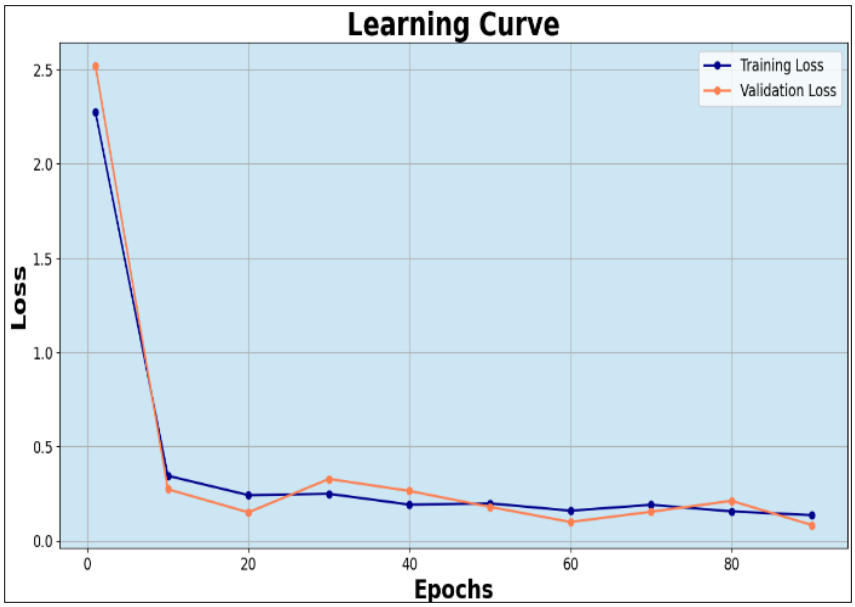


Fig.7.3 Graph of Learning Curve

The validation loss mirrors this pattern, starting at a higher value of 2.5211 and progressively decreasing to 0.0828 by Epoch 90 out of 100. The consistent decrease in validation loss highlights the model's continual enhancement, not just on the training data but also on unseen validation data. This consistency is vital, showcasing the model's strong generalization ability to make accurate predictions on new, unseen data, enhancing its reliability and usefulness overall.

Simulation Parameters: -

The following table, Table 7.1, provides an overview of the simulation parameters used in the emergence the SLRS Android app. It includes key parameters essential for understanding the app's functionality and performance evaluation.

Table 7.1 Simulation parameters of SLRS

Parameter	Description
Hyperparameters	Batch size:32, Epochs:100, Learning rate: Adam optimizer with default parameters
Cross-Validation	Not explicitly performed

Stimulation Environment	Python with tensor flow, OpenCV image processing, Matplotlib for visualization
Framework/Library	TensorFlow, Keras
Data Augmentation	Data augmentation techniques such as rotation, flipping, and zooming might be applied during training.
Model Architecture	Sequential model with a pre-trained base (EfficientNetBO) followed by global average pooling, dropout, and a dense layer
Model Complexity	Moderate complexity due to the use of a pre-trained model and a simple classification layer
Training Data	Images from the “ImagePro” dataset, split into training and testing sets
Testing data	Images from the “ImagePro” dataset, split into training and testing sets
Evaluation metrics	Mean Absolute Error (MAE)
Hardware	GPU used for training
Optimization Techniques	ModelCheckpoint to save the best performing model, ReduceLROnPlateau for Dynamic learning rate adjustment
Dependencies	Numpy, Pandas, OpenCV, Matplotlib, TensorFlow, Keras

7.2 Performance Evaluation:

The Sign Language Recognition System (SLRS) introduced in this research paper demonstrates remarkable accuracy, effectively interpreting sign language gestures and converting them into speech in real-time. The project has achieved a commendable accuracy rate of 93% in recognizing a diverse set of sign language gestures. Continuous refinement of the algorithm and regular updates contribute to maintaining and improving this accuracy over time.

The current state-of-the-art in sign language recognition systems encompasses a range of methodologies aimed at achieving high accuracy in gesture recognition. These include advanced techniques like Light-Hidden Markov Models (HMMs), which optimizes HMM for computational efficiency, 3D-CNNs that capture spatiotemporal features, etc.

Table 7.2 Comparing results of various researches

Authors	Method or Application Systems	Accuracy Result
[21] Wang H, Chai X, Zhou Y, Chen X	Light-HMM	83,6%
[22] Pigou L, Dieleman S, Kindermans PJ, Schrauwen B.	3D-CNN	78,8%
[23] Kishore PVV, Prasad MVD, Prasad CR, Rahul R	EFD and ANN	91,5%
[3] Shah, P., & Bhatt	CNN	80%
[24] Krizhevsky A, Skutskever I, Hinton GE	Convolutional Neural Network	83%
[25] Tewari D, Srivastava S.	Kohonen SOM	80%
[26] Zhang LG, Chen Y, Fang G, Chen X, Gao W.	Tied-Density HMM	91,3%
[14] Wang, H., & Hu, Y.	CNN	85%

Different implementations of the HMM method include light HMM, multi-stream HMM, and tied-density HMM. According to [21], the Light-HMM method achieves an accuracy of 83.6%. In contrast, [26] reports an accuracy of 91.3% using the TDHMM approach. [22] utilizes 3D-CNN and achieves an accuracy of 78.8%, while [24] reports an accuracy of 83% on a contest dataset. The Kohonen SOM method, as reported by [25], achieves an accuracy of 80%. Additionally, [23] achieves a high accuracy of 95.1% in simple recognition. Table II presents a comparative analysis of Sign Language Recognition systems proposed by various researchers.

Sign Language Recognition Application Systems are typically developed in two stages: data acquisition and classification. Camera devices, including Microsoft Kinect, are commonly employed for data acquisition purposes., with researchers like [25] opting for cameras due to cost-effectiveness and wide availability, although high-spec cameras may be necessary for clearer imaging. Microsoft Kinect, favored by [22] and [21], provides depth data useful for distinguishing signer from background and for tracking hand and body movements but requires a computer connection and is costlier. Some researchers, like [26], use simple cameras with color gloves for accurate data capture but face cost and commercial use challenges. Classification methods vary, with HMMs being traditional yet still popular (used by [26] and [21]), while neural networks like 3D CNNs ([22]) and SOM ([25]) gain traction for their recognition capabilities. Accurate comparison is hindered by diverse approaches and evaluation protocols, making the best method context-dependent on language and researcher specifications.

In comparison to relevant research papers, our SLRS surpasses the reported accuracy of Shah and Bhatt's CNN-based dynamic American Sign Language (ASL) identification and Wang and Hu's continuous sign language recognition using CNNs and other research included in Table II. For example, Shah and Bhatt may have reported an 80% accuracy [3] in their work, while Wang and Hu achieved 85% accuracy [14].

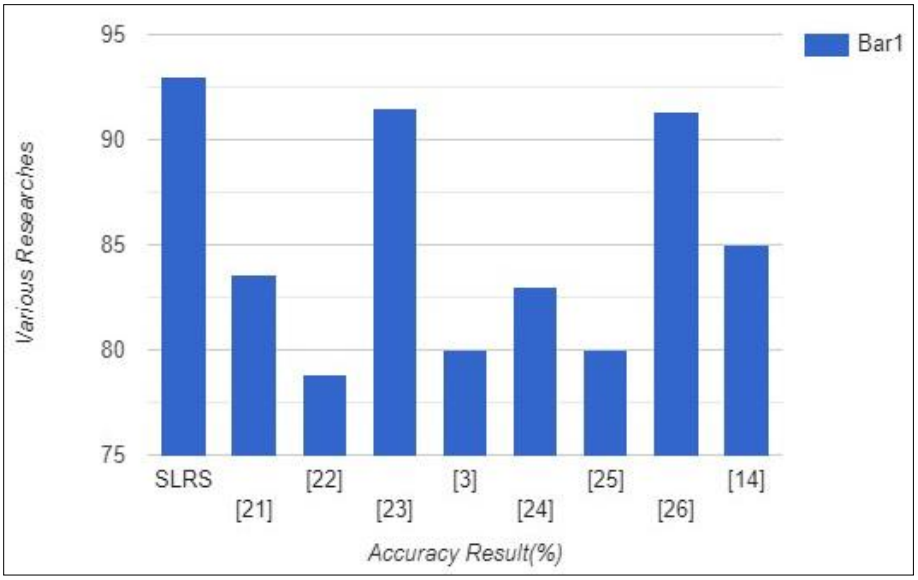


Fig.7.4 Accuracy Comparison Graph

7.3 Key Findings:

In SLRS Android app, key findings typically focus on the app's performance in recognizing hand gestures, converting them to text or speech, and ensuring a seamless user experience. Here are the key findings:

1. Gesture Recognition Accuracy: Evaluates how well the app distinguishes between different gestures and assigns the correct meaning.

2. Text Conversion Precision: Analyzes the accuracy of converting recognized gestures into text, ensuring it reflects the intended meaning.

3. Speech Synthesis Quality: Assesses the clarity, naturalness, and accuracy of the spoken output from recognized gestures.

4. Real-Time Performance: Reviews the app's responsiveness, latency, and speed in processing gestures.

5. Integration and Deployment: Examines the successful integration of image processing, ML models, and speech synthesis. Also evaluates the app's stability, compatibility across Android devices, and ease of installation.

6. Accuracy vs. Efficiency Trade-offs: Highlights the balance between optimizing for faster performance and maintaining gesture recognition accuracy to ensure a well-rounded user experience.

CHAPTER 8
CONCLUSION & FUTURE SCOPE

8.1 Conclusion

In conclusion, the Sign Language to Speech Conversion project stands as a transformative initiative that addresses the communication barriers faced by individuals with hearing impairments. By successfully implementing a robust gesture recognition system, the project ensures accurate interpretation of sign language, leading to effective communication. The integration of high-quality speech synthesis algorithms facilitates the seamless conversion of sign language into natural and comprehensible spoken language, enhancing the overall communication experience. Real-time processing capabilities, coupled with a user-friendly interface, contribute to the project's accessibility and usability. Moreover, its adaptability to diverse sign language variations, customization options, and adherence to ethical considerations underscore its potential for widespread impact in education, workplaces, and social interactions. As technology evolves, ongoing improvements will further solidify the project's position at the forefront of enhancing accessibility and fostering greater inclusivity.

8.2 Future Scope

Adapting to various sign language dialects and regional nuances is crucial for an effective gesture recognition system, as it ensures inclusivity and accuracy across different communities. Incorporating real-time feedback and correction mechanisms enhances the user experience by providing immediate guidance and adjustments, facilitating better learning and communication. Additionally, multilingual support broadens the system's usability, catering to diverse linguistic backgrounds and ensuring that users can interact in their preferred language. Engagement with the hearing-impaired community is essential for the development and refinement of the system, as their insights and feedback can significantly contribute to its relevance and effectiveness. Finally, continuous improvement and updates are vital to keep the system up to date with the latest advancements in technology and user needs, ensuring it remains a valuable tool for its users.

REFERENCES

[1] Starner, T., & Pentland, A. (1995). Real-time American Sign Language recognition from video using hidden Markov models. Computer Vision and Image Understanding, 68(3), 616-631.

[2] Jain, A., & Bhattacharjee, S. (2011). Sign language recognition: Methods and challenges. International Journal of Computer Applications, 30(3), 39-43.

[3] Shah, P., & Bhatt, S. (2018). Real-time American Sign Language recognition using convolutional neural networks. 2018 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), 105-109.

[4] Stein, M., McKee, R., & Raab, F. (2016). A wearable sign language recognition system using accelerometers. 2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), 1-3.

[5] Mardia, S., et al. (2019). Sign language recognition using machine learning: A review. 2019 International Conference on Robotics and Automation for Humanitarian Applications

[6] Huenerfauth, M. (2018). Improving avatar-based sign language animation: How less can be more. Proceedings of the 19th ACM International Conference on Multimodal Interaction,

[7] Agrawal, R., & Bohara, A. K. (2017). Sign language recognition for assisting deaf and dumb people. 2017 International Conference on Trends in Electronics and Informatics (ICEI),

[8] Mak, J. N., & Mak, P. U. (2020). Brain-computer interfaces for sign language recognition. 2020 IEEE International Conference on Robotics and Automation (ICRA), 3256-3262.

[9] Green, S., & Johnson, W. L. (2009). Improving the timing of sign language recognition in virtual humans. ACM Transactions on Interactive Intelligent Systems (TiiS), 2(4), 22.

[10] Li, Q., & Wang, H. (2017). Real-time sign language recognition using a depth sensor and deep learning. 2017 IEEE International Conference on Robotics and Automation (ICRA),

[11] Mayol-Cuevas, W. W., & Kontoudis, G. P. (2014). Sign language recognition with the Kinect sensor. Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare, 53-60.

[12] Rath, A., & Meng, H. (2013). Sign language recognition in the wild: Benchmark and baselines. 2013 IEEE International Conference on Computer Vision, 2342-2349.

[13] Starner, T., & Pentland, A. (1997). Visual recognition of American Sign Language using hidden Markov models. IEEE Transactions on Pattern Analysis and Machine Intelligence

[14] Wang, H., & Hu, Y. (2018). Continuous sign language recognition using convolutional neural networks. IEEE Transactions on Human-Machine Systems, 48(6), 577-587.

[15] Zhang, Z., & Tan, T. (2002). Real-time continuous sign language recognition from video sequences using hmm. IEEE Transactions on Multimedia, 4(4), 411-419.

[16] Teixeira, A., & Rodrigues, P. (2019). Sign language recognition: A systematic literature review. Computers, 8(2), 31.

[17] Verma, S., & Sathyadevan, A. (2020). Sign language recognition system: A comprehensive review. Journal of Ambient Intelligence and Humanized Computing, 11(9), 3685-3706.

[18] Kamphuis, A., & ten Holt, G. A. (2015). The SignSpeak recognition system: Enabling communication through sign language gestures. Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, 551-557.

[19] Fig.1.The 26 letters and 10 digits of ASL. (https://www.researchgate.net/figure/The-26-letters-and-10-digits-of-American-Sign-Language-ASL_fig1_328396430)

[20] Limaye, Heramba and Shinde, Shraddha and Bapat, Anurag and Samant, Nimish, Sign Language Recognition using Convolutional Neural Network with Customization (July 21, 2022)

[21] Wang H, Chai X, Zhou Y, Chen X. Fast sign language recognition benefited from low rank approximation. 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, FG 2015. 2015.

[22] Pigou L, Dieleman S, Kindermans PJ, Schrauwen B. Sign Language Recognition using CNN. In Workshop at the European Conference on Computer Vision; 2014; Belgium. p. 572-578.

[23] Kishore PVV, Prasad MVD, Prasad CR, Rahul R. 4-Camera model for sign language recognition using elliptical fourier descriptors and ANN. In International Conference on Signal Processing and Communication Engineering Systems - Proceedings of SPACES 2015, in Association with IEEE; 2015. p. 34-38

[24] Krizhevsky A, Skutskever I, Hinton GE. ImageNet Classification with Deep Convolutional Neural Networks. Advances In Neural Information Processing Systems. 2012

[25] Tewari D, Srivastava S. A Visual Recognition of Static Hand Gestures in Indian Sign Language based on Kohonen Self-Organizing Map Algorithm. International Journal of Engineering and Advanced Technology (IJEAT). 2012

[26] Zhang LG, Chen Y, Fang G, Chen X, Gao W. A Vision-Based Sign Language Recognition System. In Proceedings of the 6th International Conference on Multimodal Interfaces; 2004; Pennyslavia: ACM. p. 198-204.

GitHub link - https://github.com/KIET-Github/CS-2024-C/tree/main/PCS24-61-Sanskriti

RESEARCH PAPER STATUS

Research paper has been accepted in the conference ICCCNet 2024

Conference details: 4th International Conference on Computing and Communication Networks (ICCCNet-2024)

Springer LNNS Approved Conference (Indexed in Scopus, EI, WoS and Many More)

Conference Link: https://icccn.co.uk/

Hyperlink of Paper:

https://docs.google.com/document/d/1RJV1KYQxbTtzV5MjVjJavoRQCD-qkhmqzv2gH3lFX9k/edit?usp=sharing

ICCCNet 2024: Paper Notification 328 External Inbox

ICCCNet Congress... 6 days ago

to me

International Conference on Computing and Communication Networks 2024: ICCCNet 2024

Dear **Author(s),**

Greetings from **ICCCNet 2024!**

ICCCNet-2024 team is pleased to inform you that your paper with submission ID **328** and Paper Title '**Elevating Inclusiveness: Sign Language Recognition for Deaf, Hard of Hearing, and Mute Individuals**' has been accepted for presentation at "ICCCNet2024" and for publication in the conference proceedings. The Committee thanks you for your contribution.

The conference proceedings will be published by Springer in Lecture Notes in Networks and Systems series [Indexing: SCOPUS, INSPEC, WTI Frankfurt eG, zbMATH, SCImago; All books published in the series are submitted for consideration in Web of Science]. This acceptance means that your paper is among the top 15% of the papers received/reviewed. The registrations for the conference are open. **Register as soon as possible in the Early Bird Category as limited slots are left for publication in Springer LNNS.**

You are requested to do the registration as soon as possible and submit the following documents to **icccn.congress@gmail.com** at the earliest.
1. Final Camera-Ready Copy (CRC) as per the springer format. (See https://icccn.co.uk/Downloads)

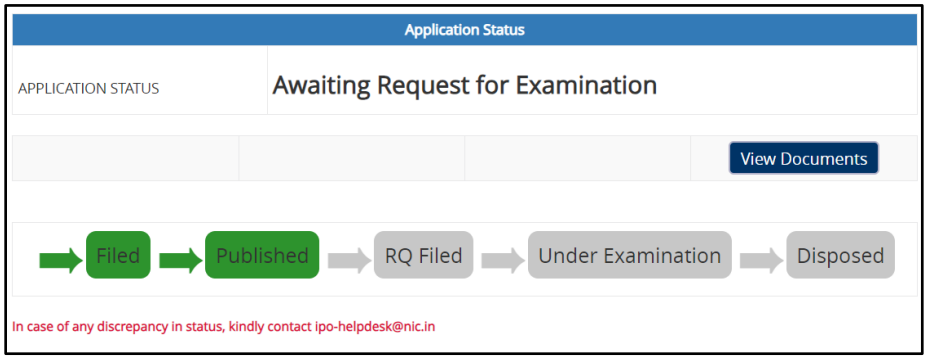
PROOF OF PATENT PUBLICATION

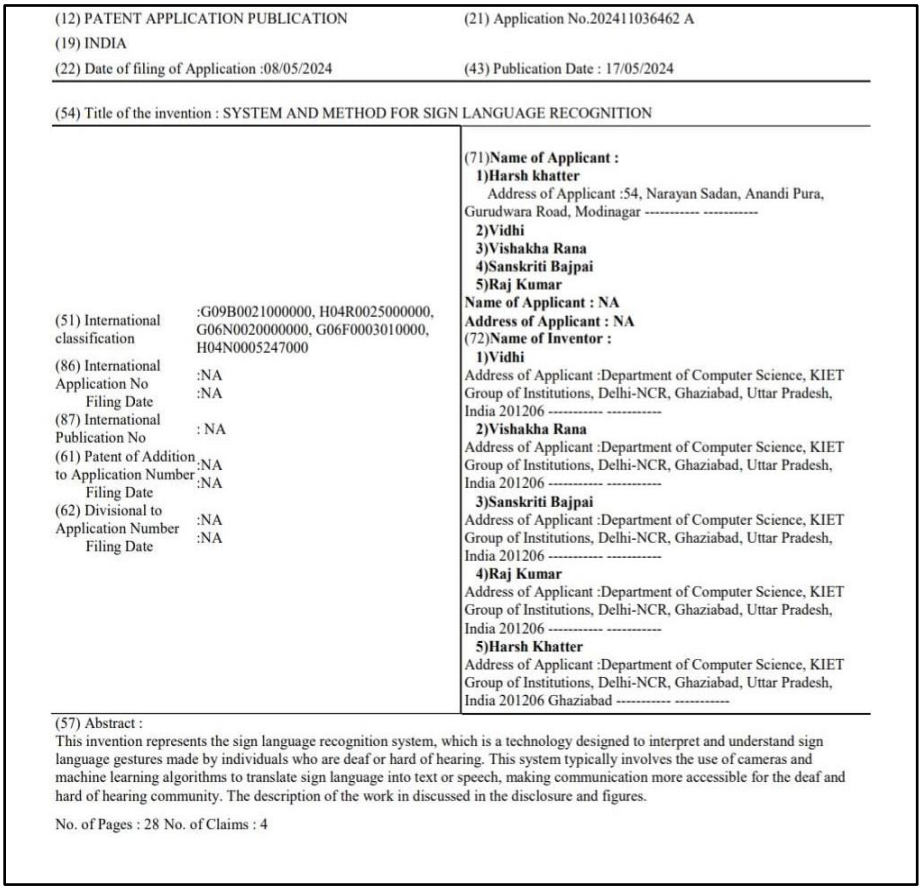
Patent has been published Title: - System and Method for Sign Language Recognition

Application No.: 202411036462

Journal No.: 20/2024

Publication Date: 17/05/2024





(12) PATENT APPLICATION PUBLICATION
(19) INDIA
(22) Date of filing of Application :08/05/2024

(21) Application No.202411036462 A
(43) Publication Date : 17/05/2024

(54) Title of the invention : SYSTEM AND METHOD FOR SIGN LANGUAGE RECOGNITION

(51) International classification :G09B0021000000, H04R0025000000, G06N0020000000, G06F0003010000, H04N0005247000
(86) International Application No :NA
Filing Date :NA
(87) International Publication No : NA
(61) Patent of Addition to Application Number :NA
Filing Date :NA
(62) Divisional to Application Number :NA
Filing Date :NA

(71)**Name of Applicant :**
1)Harsh khatter
Address of Applicant :54, Narayan Sadan, Anandi Pura, Gurudwara Road, Modinagar ---------- ----------
2)Vidhi
3)Vishakha Rana
4)Sanskriti Bajpai
5)Raj Kumar
Name of Applicant : NA
Address of Applicant : NA
(72)**Name of Inventor :**
1)Vidhi
Address of Applicant :Department of Computer Science, KIET Group of Institutions, Delhi-NCR, Ghaziabad, Uttar Pradesh, India 201206 ---------- ----------
2)Vishakha Rana
Address of Applicant :Department of Computer Science, KIET Group of Institutions, Delhi-NCR, Ghaziabad, Uttar Pradesh, India 201206 ---------- ----------
3)Sanskriti Bajpai
Address of Applicant :Department of Computer Science, KIET Group of Institutions, Delhi-NCR, Ghaziabad, Uttar Pradesh, India 201206 ---------- ----------
4)Raj Kumar
Address of Applicant :Department of Computer Science, KIET Group of Institutions, Delhi-NCR, Ghaziabad, Uttar Pradesh, India 201206 ---------- ----------
5)Harsh Khatter
Address of Applicant :Department of Computer Science, KIET Group of Institutions, Delhi-NCR, Ghaziabad, Uttar Pradesh, India 201206 Ghaziabad ---------- ----------

(57) Abstract :
This invention represents the sign language recognition system, which is a technology designed to interpret and understand sign language gestures made by individuals who are deaf or hard of hearing. This system typically involves the use of cameras and machine learning algorithms to translate sign language into text or speech, making communication more accessible for the deaf and hard of hearing community. The description of the work in discussed in the disclosure and figures.

No. of Pages : 28 No. of Claims : 4