A

Project Report

on

Silent Speak – (Multi Model Sign Language Detection)

submitted for partial fulfillment for the award of

BACHELOR OF TECHNOLOGY

DEGREE

in

Computer Science

By

Taniya Singh (2000290120174)
Shitiz Rajvanshi (2000290120145)
Shubham Goel (2000290120158)

Under the Supervision of

Ms. Arushi Gupta
Assistant Professor

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
Name: Shitiz Rajvanshi
Roll No. : 2000290120145

Signature
Name: Taniya Singh
Roll No. : 2000290120174

Signature
Name: Shubham Goel
Roll No. : 2000290120158

Date:

CERTIFICATE

This is to certify that the Project Report entitled **" Silent Speak (Sign language detection) "** which is submitted by Shitiz Rajvanshi, Shubham Goel, and Taniya Singh in partial fulfilment of the requirement for the award of degree B.Tech in the Department of Computer Science of 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. Arushi Gupta

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. Arushi Gupta, Department of Computer Science, KIET, Ghaziabad, for her constant support and guidance throughout the course of our work. Her sincerity, thoroughness, and perseverance have been a constant source of inspiration for us. It is only her cognizant efforts that our endeavours have seen the light of the day.

We also take the opportunity to 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 development of the project. 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.

Signature

Name: Shitiz Rajvanshi

Roll No. : 2000290120145

Signature

Name: Taniya Singh

Roll No. : 2000290120174

Signature

Name: Shubham Goel

Roll No. : 2000290120158

TABLE OF CONTENTS

ABSTRACT

Sign Language Recognition (SLR) stands at the forefront of technological innovation, catering to the unique communication needs of individuals facing hearing and speech challenges. Our project represents a pioneering endeavour, seeking to harness the power of cutting-edge technologies to streamline communication through American Sign Language (ASL) interpretation, image-to-text conversion, and speech-to-text transformation. First and foremost, our system boasts a sophisticated Text-to-Sign conversion module, empowered by Convolutional Neural Networks (CNNs). This component serves as the cornerstone of our endeavour, enabling the translation of textual input into expressive sign language gestures. Through meticulous training and optimization, our CNN model ensures high fidelity in capturing the nuances of ASL, facilitating clear and comprehensible communication. In parallel, our platform integrates a comprehensive Sign-to-Text functionality, employing a meticulously crafted HashMap based approach. This feature enables users fluent in sign language to effortlessly convey their messages, which are then accurately transcribed into written text Moreover, our system incorporates Voice-to-Sign capabilities, seamlessly integrating Google Voice API to facilitate real-time conversion of spoken words into ASL gestures. This feature not only enhances accessibility but also reduces the cognitive load on users, enabling intuitive communication through a familiar medium. In addition to textual and voice inputs, our platform extends its capabilities to include Image-to-Sign conversion, leveraging the cutting-edge capabilities of Google Vision API. Furthermore, our system incorporates Object Detection and Language Identification modules, augmenting its versatility and usability. Through advanced machine learning techniques, we enable the identification of objects within the user's surroundings, enhancing contextual understanding and facilitating more meaningful interactions. Additionally, language identification capabilities empower users to seamlessly switch between multiple languages, catering to diverse linguistic preferences and requirements. In culmination, our project culminates in the development of a precise Android-based mobile application, meticulously crafted to deliver an intuitive and immersive user experience. Leveraging the capabilities of Firebase ML Kit, Machine Learning (ML), and Deep Learning, we empower users to engage in effective two-way communication, fostering inclusivity and understanding between the general population and the deaf-mute community.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviations	Full Form
ASL	American Sign Language
CNN	Convolution Neural Network
API	Application Programmable Interface
ML	Machine Learning
SDLC	Software Development Life Cycle
SDK	Software Development Kit
XML	Extensible Markup Language
UI	User Interface

CHAPTER 1
INTRODUCTION

1.1 Introduction to Project

The functionality of Android applications has witnessed significant enhancements, allowing cellular phones to execute Java programs. This progress enables users worldwide to perform tasks such as reading and writing emails, browsing web pages, and playing Java games on their mobile devices. Recognizing this trend, we propose the use of Android applications to enhance communication. The advent of SMS/MMS has facilitated remote communication for deaf individuals, who previously seldom used mobile phones. Texting now enables deaf individuals to communicate with both deaf and hearing individuals. Despite the presence of deaf or dumb individuals in our surroundings, many people struggle to communicate with them. Avoiding interaction is not a solution; thus, there is a need for a solution that simplifies communication for everyone. In response to this need, we have developed an application designed to make daily communication easier for users. While technology continues to advance, its use should always be directed towards positive outcomes. Our app aims to assist deaf and dumb individuals in communicating with those around them. Although previous inventors have attempted to enhance sign language apps, we strive to make our app more efficient and reliable. Existing sign language apps often focus on either text-to- sign or sign- to-text conversion. In contrast, our app incorporates two modules: Text to Sign and Sign to Text. The Text to Sign module converts any given text into American Sign Language, while the Sign to Text module translates sign language into text. Additionally, our app allows users to provide an image, capturing or selecting one from the gallery, and cropping it. The text within the image is then displayed on the screen, and its corresponding sign language is generated. Another notable feature is voice to sign conversion, which translates voice recordings or speeches into text and then into sign language. This feature is particularly beneficial for English speakers who need sign language translation for better understanding, saving time by eliminating the need for typing. Recognizing objects is another challenge faced by deaf and dumb individuals. Instead of relying on others for identification, our app enables users to capture or select an image from the gallery and detect objects within it. The app provides a percentage-based evaluation of the object in the image. The final feature of our app is language recognition. Users can input text in a language unfamiliar to them, and the app will recognize the language, convert it into American English, and generate the corresponding gesture image.

1.2 Project Category

The project category for a sign language recognition system falls squarely under "Application or System Development." This categorization arises from the need to create software applications or systems capable of interpreting sign language gestures, thereby bridging communication barriers between individuals who use sign language and those who do not.

The project encompasses the intricate design and implementation of various components, including algorithms, user interfaces, and backend systems, all geared towards enabling the accurate recognition and interpretation of sign language gestures. Developed as an Android application using Java, this project integrates a plethora of advanced technologies. Machine learning (ML) and Convolutional Neural Networks (CNN) play pivotal roles, indicating the adoption of sophisticated deep learning techniques for precise gesture recognition.

Additionally, the incorporation of an ML toolkit enhances the system's capabilities for training and deploying ML models tailored to sign language recognition tasks. Furthermore, the utilization of Google Vision API and Google Voice API adds layers of functionality, enabling image and voice recognition within the system, thus enriching the user experience, and expanding its potential applications. The project's interdisciplinary nature underscores its complexity and significance. It combines expertise from software development, machine learning, computer vision, and natural language processing domains, presenting unique challenges and opportunities for innovation. Moreover, its real-world implications are profound, as it aims to enhance accessibility and communication for individuals who rely on sign language as their primary mode of expression. By empowering users to communicate effectively in diverse settings, such a system has the potential to foster inclusivity and break down communication barriers in various contexts, including education, healthcare, and social interactions. In conclusion, the sign language recognition system represents a commendable endeavour that exemplifies the transformative power of technology in fostering inclusivity and improving quality of life. Its comprehensive approach, leveraging advanced technologies and interdisciplinary collaboration, positions it as a promising solution with far-reaching societal impact.

1.3 Objectives

A. **Enhancing Communication Accessibility with Sign Language Detection**

Sign language detection using hand gesture recognition through machine learning is a transformative technology that significantly aids individuals with hearing or speech impairments in achieving effective communication. By leveraging advanced machine learning algorithms, this technology interprets hand gestures and converts them into sign language, enabling seamless interaction and understanding for individuals who rely on sign language as their primary mode of communication.

B. Enriching Educational Environments through Sign Language Detection Technology

Sign language detection technology also serves as a valuable resource for parents with deaf or hard-of-hearing children, enabling seamless communication and fostering strong parent-child relationships. Parents who are not fluent in sign language can use these applications to learn basic sign language gestures and effectively communicate with their children. This enhances bonding, comprehension, and emotional support within the family unit, empowering parents to engage actively in their children's development and education. Additionally, sign language detection technology offers educational materials and resources specifically designed for parents, providing guidance on sign language acquisition and effective communication strategies.

C. Breaking Down Digital Barriers with Sign Language Accessibility

Sign language detection technology significantly enhances accessibility to digital content, particularly videos, for individuals reliant on sign language. By providing real-time translation of spoken or written content into sign language, this technology ensures that deaf or hard-of-hearing individuals can fully comprehend and engage with video materials. Whether it's educational videos, online tutorials, entertainment content, or informational resources, sign language accessibility enables users to access a wide range of video content without barriers or limitations.

D. Bridging Communication Gaps with Sign Language Accessibility

Sign language accessibility facilitates direct communication between sign language users and non-sign language users, breaking down communication barriers and fostering meaningful interactions. With sign language interpretation available, individuals who are fluent in sign language can communicate effectively with those who do not understand sign language, enabling seamless exchanges of information, ideas, and emotions.

E. Bridging Communication Gaps with Sign Language Accessibility

Sign language accessibility facilitates direct communication between sign language users and non-sign language users, breaking down communication barriers and fostering meaningful interactions. With sign language interpretation available, individuals who are fluent in sign language can communicate effectively with those who do not understand sign language, enabling seamless exchanges of information, ideas, and emotions.

1.4 Structure of report

In Chapter 1, the report begins with an overview of the project, providing insight into its purpose, scope, and significance. The section "Introduction to Project" delves into the background and context of the project, outlining the problem it aims to address and the objectives it intends to achieve. Following this, the "Project Category" section categorizes the project within its broader domain, providing context for readers to understand its relevance and potential impact . Chapter 2 presents a comprehensive review of relevant literature related to the project. It starts with an exploration of existing research, theories, and practices pertinent to the project's subject matter. The "Literature Review" section synthesizes previous studies, highlighting key findings, methodologies, and gaps in knowledge. Subsequently, the "Problem Formulation" section identifies and defines the specific problems or challenges addressed by the project, drawing upon insights gleaned from the literature. Finally, the "Objectives" section outlines the goals and objectives of the project, informed by the findings of the literature review.

The focus of Chapter 3 is on the proposed system itself. It begins with a detailed description of the proposed solution, elucidating its functionality, architecture, and components. The "Unique Features of The System" section highlights the distinctive aspects of the proposed system compared to existing solutions, emphasizing its innovative contributions or improvements. Additionally, the "Structure of Report" section provides an overview of the organization and flow of the report, guiding readers through the subsequent chapters. Chapter 4 delves into the analysis and specification of the system requirements. It commences with a feasibility study, evaluating the technical, economical, and operational feasibility of the proposed system. Following this, the "Requirement Specification" section outlines the functional and non-functional requirements of the system, detailing its features, capabilities, and constraints. The chapter concludes with a discussion on the software development life cycle (SDLC) model to be employed, along with system design using Data Flow Diagrams (DFD) and Use Case Diagrams.

In Chapter 5, the report provides insights into the implementation phase of the project. It introduces the languages, tools, and technologies utilized for developing the proposed system, shedding light on the technical aspects of its realization. The section aims to offer readers a comprehensive understanding of the implementation environment and methodologies employed in bringing the project to fruition. Chapter 6 focuses on the testing and maintenance activities carried out during the project lifecycle. It discusses the various testing techniques employed to validate the functionality, performance, and reliability of the system. Additionally, it elaborates on the test cases developed and executed to ensure the quality and robustness of the software. Furthermore, the chapter addresses maintenance strategies adopted to sustain and enhance the system's functionality over time.

Chapter 7 presents the results of the project and engages in discussions around them. It includes visual representations of the user interface and system modules, along with snapshots of the system and database. The section provides a detailed analysis and interpretation of the findings, offering insights into the performance, usability, and effectiveness of the developed system. It serves as a platform for evaluating the project outcomes and fostering meaningful discussions on its implications and future directions. The final chapter concludes the report by summarizing the key findings, insights, and contributions of the project. It reiterates the significance of the work undertaken and its implications for the broader field or industry. Additionally, it outlines potential avenues for future research, development, and improvement, highlighting areas where the project could be further extended or refined. The chapter encapsulates the report's overarching narrative and leaves readers with a sense of closure while igniting curiosity about future possibilities.

CHAPTER 2
LITERATURE REVIEW

2.1 Literature Review

In [1] Sign Language especially Indian Sign Language (ISL), for the deaf and mute. It notes limited research post-ISL standardization, focusing on static hand gestures with minimal attention to dynamics. Despite efforts on ISL alphabet recognition, the process involves multiple stages, surveyed to assess research progress. In [2] an intelligent system for translating sign language to text, comprising hardware and software components. The hardware incorporates flex, contact, and inertial sensors on a glove. Software features a classification algorithm leveraging k-nearest neighbours, decision trees, and dynamic time warping, enabling static and dynamic gesture recognition. In [3] three methods for sub-unit-based sign recognition. Boosting is employed to learn appearance-based sub-units, merged with a second-stage classifier for word-level sign learning. Another approach integrates 2D tracking-based sub-units with appearance-based handshape classifiers. The final method translates these into 3D, enabling real-time, user-independent recognition of isolated signs. In [4] a deep convolutional neural network for direct classification of hand gestures in images, eliminating the need for segmentation or detection stages. In [5] two novel hand gesture recognition approaches for real-time sign language comprehension. Employing a hybrid feature descriptor merging SURF and Hu Moment Invariant methods yields a strong recognition rate. SURF and moment invariant features exhibit resilience to diverse variations, ensuring effective real-time performance. [5] Introduces two innovative methods for real-time recognition of hand gestures in sign language. These methods merge SURF and Hu Moment Invariant techniques into a combined feature descriptor, improving recognition accuracy while maintaining low time complexity. They also introduce derived features and utilize KNN, SVM, and HMM for classification, demonstrating enhanced real-time efficiency and robustness. [6] Presents novel strategies for real-time recognition, translation, and video production in Sign Language (SL). Employing Media Pipe and hybrid CNN + Bi-LSTM models for recognition, and NMT + GAN models for video generation, achieving classification accuracy exceeding 95%. Evaluation metrics reveal substantial enhancements, including a

38.06 BLEU score and impressive visual quality. [7] Addresses the challenges of Continuous Sign Language Recognition (CSLR) by introducing Sign BERT, a deep learning framework merging BERT and Res Net. Outperforming conventional methods in accuracy and word error rate on demanding datasets, Sign BERT underscores its effectiveness in modeling sign languages and extracting spatial features for real-time CSLR. [8] Examines sign language research, particularly vision-based hand gesture recognition systems from 2014 to 2020. Through analysis of 96 articles, it identifies key research areas: data acquisition, environment, and gesture representation. Signer-dependent recognition averages 88.8%, while signer-independent recognition averages 78.2%, indicating opportunities for improvement, especially in continuous gesture recognition. [9] Introduces a dynamic hand gesture recognition system leveraging multiple deep learning architectures. Evaluated on a challenging dataset, it outperforms existing methods, demonstrating effectiveness in uncontrolled environments with diverse gestures. [10] Presents a real-time hand gesture recognition system utilizing a cost-effective webcam and image processing techniques. The system comprises four stages: image preprocessing, region extraction, feature extraction, and matching, achieving a 90.19% recognition rate for American Sign Language (ASL) alphabet gestures under various lighting and hand conditions. [11] Recent advancements in sign language recognition technology offer promise for improving communication accessibility for individuals with disabilities. These systems utilize computer vision to interpret sign language gestures, achieving a notable accuracy of 91.67%, surpassing previous works. They facilitate better interaction and understanding among diverse language speakers. [12] This project employs Convolutional Neural Networks (CNN) and deep learning to bridge communication gaps among speech and hearing-impaired individuals by offering a platform for choosing between Indian and American Sign Languages. It further translates signs into multiple regional languages, aiming to foster efficient communication and inclusivity for the deaf community worldwide. [13] This paper introduces a real-time sign language recognition system using YOLOv4, aimed at bridging communication gaps between hearing/speech-impaired and hearing individuals. Utilizing the ISL-CSLTR dataset and data augmentation, the system achieves 98.4% mean average precision (mAP) and real-time translation, demonstrating applicability for real-world use. [14]

This survey addresses the integration of deaf-mute individuals into mainstream society in China through efficient sign language processing technologies. It reviews Chinese sign language recognition and translation, including classification, feature exploration, available datasets, and future research trends, aiming to provide a comprehensive understanding and direction for further development in this field. [15] Recognizing sign language is vital for facilitating communication among the deaf and hard of hearing. This paper introduces a novel approach to sign language recognition, integrating linguistic modeling during sign-to-GLOSS conversion. By implementing a text correction module, it achieves higher accuracy, as validated on the RWTHPHOENIX-Weather-2014-T and CSL datasets.

2.2 Research Gaps

Identifying research gaps in a project involving sign language detection technology for individuals with hearing impairments entails recognizing areas where further investigation or development is needed to enhance the effectiveness, efficiency, or impact of the technology. Here are some potential research gaps:

A. **Accuracy and Reliability of Gesture Recognition:**
Despite advancements in machine learning algorithms, there may still be challenges in accurately and reliably recognizing complex sign language gestures, particularly in noisy or dynamic environments. Research could focus on improving the accuracy and robustness of gesture recognition systems through techniques such as data augmentation, model optimization, or ensemble learning approaches.

B. **Real-Time Performance and Latency:**
Real-time performance and low latency are crucial for ensuring a seamless user experience in applications involving sign language detection. Research could explore optimization strategies to reduce processing time and latency, such as efficient algorithm implementations, hardware acceleration, or cloud-based computation offloading.

C. **Adaptability to Diverse Sign Language Variations:**
Sign languages exhibit significant variation across regions, cultures, and communities, posing a challenge for universal sign language detection systems. Research could investigate techniques for adapting detection models to diverse sign language variations, including dialects, regional differences, and idiosyncratic signing styles.

D. **User Interface Design and Usability:**
User interface design plays a critical role in the usability and accessibility of sign language detection applications. Research could focus on designing intuitive and user-friendly interfaces tailored to the needs and preferences of deaf or hard of hearing users,

considering factors such as gesture input methods, visual feedback, and customization options.

E. **Inclusive Dataset Collection and Model Bias:**

The availability of diverse and representative datasets is essential for training accurate and unbiased sign language detection models. Research could address the challenges of collecting inclusive datasets that capture the diversity of sign language usage across different demographics, regions, and contexts, while mitigating biases and ensuring equitable representation.

F. **Privacy and Ethical Considerations:**

Sign language detection technology raises privacy and ethical concerns related to data security, consent, and user autonomy. Research could explore privacy-preserving techniques for protecting sensitive user data, transparent and accountable data governance frameworks, and ethical guidelines for the responsible development and deployment of sign language detection systems.

G. **Integration with Assistive Technologies:**

Integrating sign language detection technology with other assistive technologies, such as augmented reality (AR), wearable devices, or smart home assistants, could enhance its utility and accessibility for deaf or hard of hearing users. Research could investigate innovative ways to integrate sign language detection into existing assistive technology ecosystems, enabling seamless interaction and interoperability.

2.3 Problem Formulation

Sign language serves as a crucial mode of communication for individuals with hearing or speech impairments, enabling them to express themselves, interact with others, and access information. However, communication barriers persist for those unfamiliar with sign language, hindering effective communication and social inclusion for deaf or hard-of-hearing individuals. To address this challenge, a comprehensive sign language detection and recognition system leveraging machine learning technology is essential. The primary objective of this project is to develop a robust system capable of accurately interpreting sign language gestures in various contexts, thereby facilitating seamless communication between sign language users and non-users. One of the primary problems facing the deaf or hard-of-hearing community is the lack of universal communication access. Many individuals, including family members, educators, healthcare professionals, and service providers, may not be proficient in sign language, leading to communication barriers and misunderstandings. This project aims to bridge this gap by developing a system that can accurately interpret sign language gestures and facilitate communication between sign language users and non-users.

Sign language is complex and exhibits significant variability across regions, cultures, and communities. Different sign languages, dialects, and signing styles pose challenges for developing a universal sign language detection system. Additionally, sign language includes facial expressions, body movements, and spatial grammar, adding layers of complexity to gesture recognition. This project seeks to address these challenges by developing machine learning models capable of accurately recognizing diverse sign language gestures in real-world settings. Deaf or hard-of-hearing individuals often face limited accessibility to digital content, including videos, online courses, and multimedia presentations. Existing accessibility features such as closed captions may not adequately convey the richness and nuances of sign language communication. By developing tools for automatically adding sign language interpretation to digital content, this project aims to enhance accessibility and inclusivity for individuals reliant on sign language.

Deaf or hard-of-hearing individuals may encounter educational and socioeconomic barriers due to communication challenges and limited access to resources. In educational settings, students with hearing impairments may struggle to access curriculum materials and participate fully in classroom activities. In professional settings, job opportunities may be limited for individuals with communication disabilities. This project seeks to address these barriers by developing educational tools, workplace accommodations, and assistive technologies to support deaf or hard-of-hearing individuals in their academic and professional endeavours.

CHAPTER 3
PROPOSED SYSTEM

3.1) Proposed System

The proposed application aims to address communication barriers faced by deaf and dumb individuals by leveraging advancements in Android technology. Here's an overview of the proposed features and methodologies described in the text:

A. **Text-to-Sign Conversion:** Text-to-sign conversion using the HashMap class involves mapping textual input to a corresponding sign language representation. HashMap in Java provides a convenient way to store key-value pairs, where keys are unique identifiers mapped to corresponding values.

B. **Sign-to-Text Conversion:** Sign-to-text conversion using Convolutional Neural Networks (CNNs) involves training a model to interpret sign language gestures from images or videos and translate them into text. This process begins with data collection and preprocessing, where a dataset of sign language gestures paired with corresponding textual representations is compiled and prepared for training. The CNN architecture is then designed, typically comprising convolutional layers for feature extraction and fully connected layers for classification. Training involves optimizing the model's parameters using techniques like backpropagation and gradient descent on the training data.

C. **Image Processing**: The integration of image analysis capabilities within the application enriches user interaction by enabling the interpretation of textual content and generation of corresponding sign language representations. Users can submit images containing text, and the app utilizes Optical Character Recognition (OCR) technology to extract the text embedded within the image. This extracted text is then processed to generate the corresponding sign language gestures, leveraging the sign language conversion model, potentially based on CNNs, trained to map textual input to sign language representations.

D. **Voice to Sign Conversion**: The application's capability to translate voice recordings or speeches into text, followed by conversion into sign language, serves as a valuable tool for enhancing communication accessibility, especially for English speakers requiring sign language translation for improved comprehension. Users can simply record their spoken words or speeches using the app's voice translation feature, which utilizes advanced Speech-to-Text (STT) technology to accurately transcribe spoken content into textual form.

E. **Language Recognition and Translation:** The application's ability to recognize input text in multiple languages, translate it into English, and then generate corresponding sign language gestures serves as a powerful tool for fostering communication across diverse linguistic backgrounds. Users can input text in their preferred language, leveraging the app's multilingual support, which employs sophisticated Natural Language Processing (NLP) techniques to identify and interpret text in various languages.

F. **Optimizing Image Recognition:** The application's emphasis on optimizing image recognition for efficient alphabet identification underscores its commitment to providing accurate and reliable sign language translation services. Achieving this goal involves meticulous consideration of various factors that can affect the performance of image recognition algorithms, particularly when identifying alphabetic signs.

G. **Text-to-Image Matching:** Integrating conditions for text-to-image matching is a strategic approach to enhance text recognition within the application. By imposing specific criteria or rules during the matching process, such as considering font styles, sizes, and orientations, the accuracy and reliability of text recognition can be significantly improved. This ensures that the identified text aligns closely with the intended content, even amidst variations in formatting or presentation.

H. **Global Accessibility:** The inclusion of a feature for interpreting spoken language in various countries underscores the application's commitment to global accessibility and inclusivity. By accommodating multiple languages, the application enables users from diverse linguistic backgrounds to engage in seamless conversations, transcending language barriers.

3.2 Unique Features of The System (Difference from Existing System)

The sign language detection system boasts several unique features that set it apart:

A. **Comprehensive Communication Modalities:** The system's diverse communication modalities ensure that users can communicate effectively using their preferred method. Text-to-sign translation enables users to input text and receive corresponding sign language representations, while image-to-sign translation allows for the interpretation of sign language gestures captured in images. Voice-to-sign translation facilitates communication by converting spoken language into sign language, and sign-to-text translation enables sign language users to express themselves through written text. This multifaceted approach ensures inclusivity and accessibility in communication, accommodating the diverse needs and preferences of users.

B. **Educational Support:** The system serves as a valuable educational tool, particularly for individuals learning sign language as a second language. By providing resources and feedback to learners, such as tutorials, quizzes, and interactive exercises, the system enhances their understanding and proficiency in sign language. Learners can practice sign language in various contexts, receive instant feedback on their gestures, and track their progress over time, leading to improved learning outcomes and language acquisition.

C. **Accessibility to Digital Content:** One of the system's standout features is its ability to make digital content accessible to individuals reliant on sign language. Through its translation services for videos, online courses, and other digital content, the system ensures that users with hearing impairments have equal access to information and educational resources. This promotes inclusivity and empowers users to engage with a wide range of digital content, fostering lifelong learning and personal development.

D. **Research Capabilities:** The system facilitates research endeavours in the field of sign language communication by providing tools for the analysis of sign language structure, usage patterns, and contextual variations. Researchers can leverage the system's data collection and analysis capabilities to gain insights into the dynamics of sign language communication, identify trends and patterns, and contribute to the advancement of the field. This supports the development of evidence-based practices and interventions to improve communication outcomes for individuals with hearing impairments.

E. **Adaptability and Customization:** Designed to be adaptable and customizable, the system caters to the unique communication needs of individual users. Its ability to recognize a broad range of gestures and adapt to different communication styles ensures an inclusive user experience. Users can customize their settings, preferences, and language preferences to tailor the system to their specific needs and preferences, enhancing usability and effectiveness.

F. **Continuous Improvement:** Leveraging machine learning algorithms, the system continuously evolves and improves over time. It can recognize a wider range of gestures and adapt to the evolving communication needs of users. By incorporating user feedback, updating its algorithms, and expanding its capabilities, the system ensures long-term effectiveness and relevance in facilitating communication for individuals with hearing impairments.

CHAPTER 4
REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION

4.1 Feasibility Study (Technical, Economical, Operational)

The different aspects of the feasibility study are –

Technical Feasibility:

The proposed system's technical feasibility hinges on the compatibility of hardware and software components. Android smartphones with a minimum version of 5.0 and a rear camera capable of detecting American Sign Language (ASL) gestures are readily available in the market, ensuring widespread accessibility. The integration of Google Vision API and Firebase ML Kit into the Android Studio development environment facilitates the implementation of machine learning functionalities without extensive expertise in neural networks. The use of the HashMap class for efficient data management further enhances technical feasibility by simplifying the mapping of processed inputs with stored database entries. Overall, the system's technical feasibility is high, supported by the availability of compatible hardware and user-friendly software development tools.

Economical Feasibility:

From an economic standpoint, the system demonstrates feasibility through its utilization of widely available hardware components and open-source software libraries. Android smartphones with the required specifications are prevalent and competitively priced, minimizing upfront hardware costs. Moreover, Android Studio, Google Vision API, and Firebase ML Kit are offered free of charge, reducing software development expenses. Additionally, the system's scalability allows for incremental improvements and updates without significant additional costs. Therefore, the economic feasibility of the system is favourable, with low initial investment and minimal ongoing expenses.

Operational Feasibility:

Operationally, the system offers a user-friendly interface and seamless integration with existing services, enhancing its feasibility for end-users. The mobile application developed using Android Studio provides a familiar environment for Android users, ensuring ease of navigation and interaction. Integration with Google Vision API and Firebase ML Kit enables efficient communication and machine learning capabilities, enhancing the user experience. Furthermore, the system's support for various functionalities, such as text-to-sign translation and ASL gesture recognition, aligns with the diverse communication needs of users. Overall, the operational feasibility of the system is high, supported by its intuitive interface and versatile functionality.

4.2 Software Requirement Specification

A. **Android Studio:** Android Studio serves as the official integrated development environment (IDE) for Google's Android operating system. Developed on JetBrains' IntelliJ IDE software, this IDE is tailored explicitly for Android development.

B. **Google Vision API**: Google Vision API is part of the suite of Google APIs, providing application programming interfaces (APIs) that enable communication with various Google Services and integration with other services. Examples include Search, Gmail, Translate, or Google Maps. Third-party applications can utilize these APIs to enhance or extend the functionality of existing services. The Google Vision API, in particular, offers functionalities such as analytics, machine learning as a service (the Prediction API), and access to user data.

C. **Firebase ML Kit:** Firebase ML Kit is a mobile software development kit (SDK) that leverages Google's machine learning expertise to enhance Android and iOS applications. It offers a powerful yet user-friendly package for implementing machine learning functionalities, suitable for both newcomers and experienced developers. With ML Kit, developers can integrate machine learning capabilities into their apps with just a few lines of code, eliminating the need for in-depth knowledge of neural networks or model optimization. For experienced machine learning developers, ML Kit provides convenient APIs to incorporate custom TensorFlow Lite models into mobile applications.

D. **HashMap Class:** The mapping of processed input with the stored database is facilitated by the use of the HashMap function. The HashMap class implements the Map interface, allowing the storage of key-value pairs where keys must be unique. This class is found in the java.util package and plays a crucial role in efficiently managing associations between processed inputs and the corresponding data stored in the system’s database.

4.2.1 Data Requirements

- Speech Input Data: The Google Voice API provides access to speech recognition services, allowing users to input spoken language into the app.
- Image Input Data: The Google Cloud Vision API enables the app to analyse images containing sign language gestures.
- Machine Learning Models: The Firebase ML Toolkit offers pre-trained machine learning models for various tasks, including image labelling and language identification.
- Data Privacy: All data obtained through APIs must adhere to privacy regulations and policies. User consent and data anonymization may be necessary to protect user privacy.
- Data Transmission: Data transmitted between the app and APIs should be encrypted to ensure confidentiality and integrity.

4.2.2 Functional Requirement

Functional requirements define the specific functionalities or features that the system must provide to meet user needs. In the context of the app, functional requirements include:

- Sign to Text: This functionality allows users to input sign language gestures captured by the device's camera and translates them into text.
- Text to Sign: This functionality enables users to input text and converts it into sign language gestures displayed on the screen.
- Image to Sign: This functionality allows users to upload or capture images containing sign language gestures and translates them into text or sign language.
- Voice to Sign: This functionality converts spoken language input into sign language gestures using speech recognition.
- Language Identification: This functionality identifies the language of text input or spoken language.
- Object Detection: This functionality utilizes computer vision techniques to detect and identify objects in images or video feeds.

4.2.3 Performance Requirement

Performance requirements specify the desired performance characteristics of the system. For the app, performance requirements may include:

- Low latency: The app should respond quickly to user inputs.
- High accuracy: Machine learning models should accurately recognize sign language gestures and objects.
- Scalability: The app should be able to handle multiple concurrent users without significant performance degradation.

4.2.4 Maintainability Requirement

Maintenance requirements specify activities needed to keep the system operational, up-to-date, and in compliance with evolving needs and standards. Maintenance requirements for the app may include:

- Regular updates and patches to address bugs and security vulnerabilities.
- Compatibility testing with new Android versions and APIs.
- Periodic retraining of machine learning models to improve accuracy and accommodate new sign language variations.
- Documentation updates to reflect changes in functionality or architecture.

4.2.5 Security Requirement

Security requirements define measures to protect the system and its data from unauthorized access, manipulation, or disclosure. Security requirements for the app may include:

- Secure storage and transmission of user data.
- Access controls to restrict unauthorized access to sensitive functionality or data.
- Authentication mechanisms to verify user identities.
- Encryption of sensitive data to prevent eavesdropping or tampering.

4.3 SDLC Model to Be Used

The suitable Software Development Life Cycle (SDLC) model for this project would be the Agile model. Agile emphasizes iterative development, collaboration between cross-functional teams, and responding to change quickly. This aligns well with the dynamic nature of mobile application development, especially when integrating advanced features like machine learning and image processing.

Implementing the Agile SDLC in an Android app with six distinct features entails a dynamic and iterative approach to development. Each feature, from sign-to-text conversion to voice-to-sign conversion, undergoes a series of iterative cycles, typically lasting a few weeks. During each cycle, development teams prioritize user stories or requirements for the feature, focusing on delivering incremental value to users. For instance, in sign-to-text conversion, initial versions might rely on basic gesture recognition algorithms, gradually improving accuracy and expanding gesture support based on user feedback. Similarly, in text-to-sign conversion, the initial implementation might involve simple translation mechanisms, evolving over time to incorporate more sophisticated sign language generation techniques. Throughout the process, regular collaboration between developers, designers, and stakeholders ensures alignment with user needs and project goals. Continuous integration and testing practices are employed to maintain code quality and ensure functionality across different devices and Android versions. By embracing Agile principles such as flexibility, collaboration, and responsiveness to change, the development team can deliver a robust and user-friendly Android app that effectively addresses the diverse communication needs of its users.

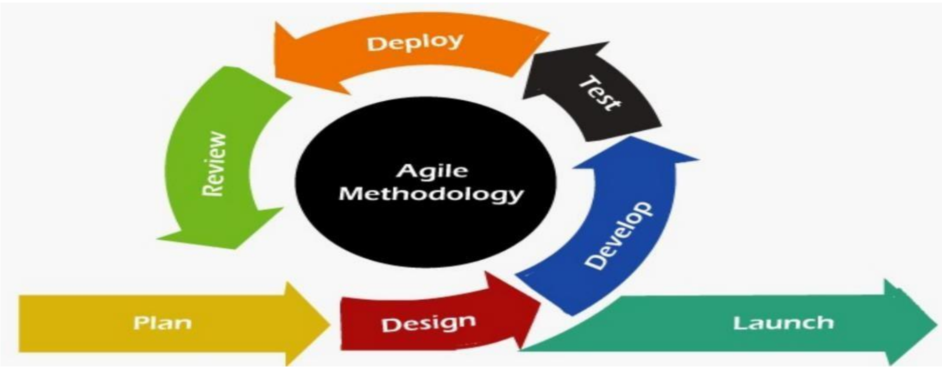


Figure 4.1: Agile Model

4.4 System Design

The Sign Language Detection and Recognition Android App is designed to provide seamless communication support for individuals with hearing or speech impairments. The system comprises various modules, including Sign to Text, Text to Sign, Image to Sign, Voice to Sign, Language Identification, and Object Detection. These modules interact with the user interface to capture inputs such as sign language gestures, text, images, and voice recordings. The data flows through the system, passing through different processing stages where machine learning algorithms, speech recognition, and image analysis techniques are employed. External APIs, such as Google Voice API, Google Cloud Vision API, and Firebase ML Toolkit, are integrated to enhance the app's capabilities for translation, analysis, and recognition tasks. The backend infrastructure, which may utilize Firebase services, supports API integration, data storage, and processing. The overall architecture follows a client-server model, with the Android app serving as the client interface and the backend servers hosting the necessary services. Through this system design, the app aims to provide accessible and inclusive communication support, bridging the gap between sign language users and non-users effectively.

4.4.1 Data Flow Diagram Level 0

At the highest level of abstraction, the DFD Level 0 provides an overview of the system and its interactions with external entities. It illustrates the flow of data between the main processes within the system.

Figure 4.2: DFD Level 0

Data Flow Diagram Level 1

DFD Level 1, also known as Level 1 Data Flow Diagram, dives deeper into the system compared to the high-level overview of a Level 0 DFD (Context Diagram).Elements of DFD level 1 include processes, data flows, data stores and external entities.

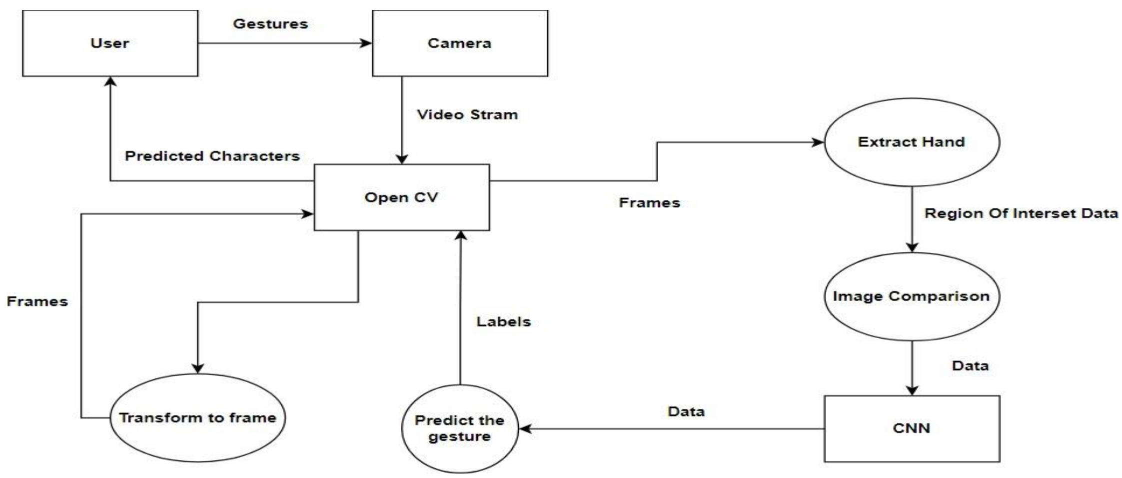


Figure 4.3: DFD Level 1

4.4.2 Use Case Diagram

A use case diagram is a visual representation of how users (or external systems) interact with a system to achieve specific goals. It's a core concept in Unified Modeling Language (UML), a standard for software design.

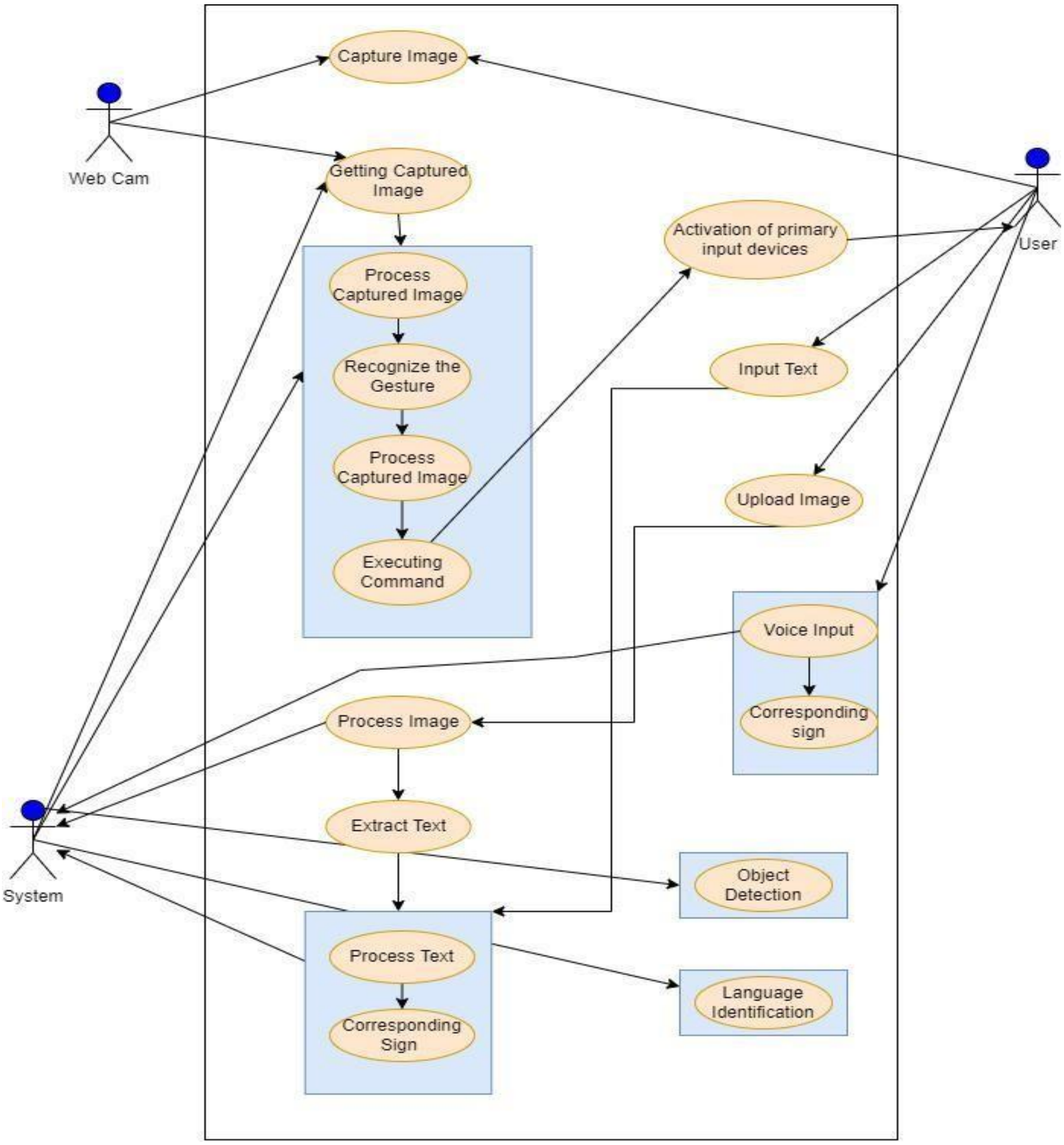


Figure 4.4: Use Case Diagram

CHAPTER 5
IMPLEMENTATION TESTING AND MAINTAINANCE

5.1 Introduction to Languages, Tools, and Technologies Used for Implementation

During the development of the mobile application from June 2023 to August 2023, a diverse array of technologies, tools, and languages were utilized to create innovative features aimed at bridging communication barriers between hearing individuals and the deaf and mute community.

A. **Java:** Java has been the primary programming language for Android development since the inception of the Android platform. It is the language recommended and supported by Google for building Android apps. Java in Android Studio follows the same syntax and features as standard Java development. Developers write Java code to define the behaviour of Android apps, including defining activities, services, broadcast receivers, and other components. Java interacts with the Android Software Development Kit (SDK) provided by Google. The Android SDK contains libraries and APIs that developers use to build Android apps. Java code makes use of these APIs to access device functionalities such as sensors, camera, location services, and more. Java allows developers to integrate third-party libraries and frameworks into their Android apps. These libraries extend the functionality of the app by providing additional features and capabilities. Examples include libraries for networking, database management, image processing, and more.

B. **Material Design UI:** Material Design is a design language developed by Google in 2014. It aims to provide a unified system for creating visually appealing, intuitive, and consistent user interfaces across different platforms and devices. Material Design emphasizes the use of a consistent visual hierarchy to guide users' attention and interactions. It employs principles such as elevation, depth, and shadow to create a sense of depth and hierarchy in the UI. Material Design is inspired by the physical world, with elements that behave as if they are made of tangible materials. It uses properties like colour, typography, and motion to create intuitive interactions and convey meaning to users. Material Design introduces a flexible and responsive layout system based on a grid. It encourages the use of responsive design principles to adapt

UI layouts to different screen sizes and orientations, providing a consistent experience across devices. Material Design prioritizes accessibility, ensuring that UI elements are perceivable, operable, and understandable for all users. It includes recommendations for colour contrast, text size, and focus indicators to improve accessibility for users with disabilities.

C. API: APIs (Application Programming Interfaces) play a crucial role in integrating Android apps with Firebase, Google's mobile and web application development platform. Firebase offers a suite of services and features for building high-quality apps, including authentication, real-time database, cloud messaging, analytics, and more. APIs act as bridges that allow Android apps to interact with these Firebase services programmatically. For example, the Firebase Authentication API enables apps to handle user authentication using email/password, phone number, or third-party providers like Google or Facebook. APIs extend the functionality of Android apps by providing access to advanced features and capabilities offered by Firebase. For instance, the Firebase Cloud Messaging (FCM) API enables apps to send push notifications to users' devices, helping developers engage with their audience in real-time.

Many Firebase services, such as the Realtime Database and Firestore, offer real-time data synchronization between the client app and the Firebase backend. APIs facilitate data exchange and synchronization between the Android app and Firebase servers, enabling seamless communication and updates. Firebase APIs provide mechanisms for handling events and triggers within the app. For example, the Firebase Analytics API allows developers to track user interactions, monitor app performance, and gain insights into user behaviour. Similarly, the Firebase Cloud Functions API enables developers to respond to events in the Firebase ecosystem, such as database changes or authentication events, by executing custom server-side logic.

D. ML Kit: Firebase ML Kit is a powerful mobile SDK provided by Google that allows developers to easily integrate machine learning capabilities into their Android and iOS applications. It offers a range of pre-trained models and APIs for tasks such as image labelling, text recognition, object detection, and face detection. Firebase ML Kit's image labelling API allows developers to identify and label objects, scenes, and activities within images. This functionality is particularly useful for applications that require image understanding, such as image-to-sign translation. By leveraging image labelling, developers can extract relevant information from images, such as objects or text, and use it to provide contextually relevant translations in sign language. With Firebase ML Kit's image labelling capabilities, developers can implement image-to-sign translation features in their applications. By labelling objects or text within images, developers can infer the meaning of the content and translate it into sign language gestures or text. This enables users to understand and interact with visual content more effectively, bridging communication gaps for users with hearing impairments. Firebase ML Kit also offers language recognition capabilities, allowing developers to detect the language of text within images or documents. This functionality is essential for applications that involve multilingual content, such as language learning or translation apps. By accurately identifying the language of text, developers can provide more relevant and personalized experiences to users, such as offering language-specific resources or translations.

E. **OpenCV:** OpenCV (Open-Source Computer Vision Library) is a versatile open-source library widely used for computer vision tasks, including image processing, object detection, and machine learning. OpenCV provides algorithms and tools for real-time object detection, allowing developers to identify and locate objects within images or video streams in real-time. This capability is invaluable for a wide range of applications, including augmented reality, surveillance, automotive safety, and assistive technologies. Real-time object detection powered by OpenCV can significantly improve accessibility for visually impaired individuals. By detecting and recognizing objects in the environment, applications can provide auditory or haptic feedback to assist users in navigating their surroundings more effectively. For example, a smartphone app can use OpenCV to detect obstacles in the user's path and provide audio cues to help them avoid collisions. OpenCV’s real-time object detection capabilities can be leveraged in assistive technologies to enhance usability for users with disabilities. For instance, applications can use OpenCV to detect and recognize hand gestures, facial expressions, or sign language gestures, enabling users to interact with devices and interfaces more intuitively. This can facilitate communication, control of electronic devices, and access to information for users with motor impairments or communication disorders. By incorporating real-time object detection with OpenCV, applications can provide users with enhanced and interactive experiences. For example, augmented reality applications can overlay digital information or virtual objects onto the real-world scene based on detected objects, creating immersive and engaging user experiences. OpenCV is optimized for performance and efficiency, allowing real-time object detection to be implemented on resource-constrained devices such as smartphones, tablets, and embedded systems. This enables developers to deploy sophisticated computer vision applications with minimal computational overhead, ensuring smooth and responsive user experiences.

F. XML: The utilization of Extensible Markup Language (XML) for designing layouts and defining UI elements within an Android application offers several benefits, including ease of use, flexibility, and maintainability.XML is a markup language commonly used in Android development to define the structure and appearance of user interfaces. It provides a hierarchical structure for organizing UI elements, such as text views, buttons, images, and layouts, using tags and attributes. XML files are human-readable and allow developers to describe UI components declaratively, making it easier to visualize and modify the UI layout.XML enables developers to create visually appealing user interfaces by specifying attributes such as colours, dimensions, margins, padding, and alignment. Developers can define custom styles and themes to maintain a consistent look and feel across the application. Additionally, XML supports the use of vector graphics and drawable resources to enhance the visual appeal of UI elements.XML layouts can be designed to be responsive, adapting dynamically to different screen sizes, resolutions, and orientations. Developers can use layout containers such as LinearLayout, RelativeLayout, ConstraintLayout, and GridLayout to create flexible and adaptive UI designs. XML attributes like layout weights, constraints, and guidelines help ensure that UI elements resize and reposition appropriately on various devices. Android applications often rely on third-party libraries and dependencies to extend functionality beyond what is provided by the Android SDK. The mentioned dependencies, such as Firebase ML Vision, Yarolegovich Sliding-Nav, Image-to-Text, and Language Recognition, are essential for integrating features like image labelling, navigation, image-to-text conversion, and language identification into the application. These dependencies are typically added to the project's build.gradle file and imported as modules or libraries, allowing developers to leverage their functionality seamlessly.

CHAPTER 6
TESTING AND MAINTAINANCE

6.1 Testing Techniques and Test Cases Used

The decision to adopt a Waterfall methodology for a project is typically based on specific project requirements, constraints, and organizational factors. Here are some common reasons for choosing the Waterfall methodology:

• **Well-defined requirements:** When the project has clearly defined and stable requirements that are unlikely to change significantly throughout the project's lifecycle. The waterfall is suitable when you can gather and document all the requirements upfront.

•**Low Uncertainty:** If there is a high level of confidence in the project scope and objectives, and the technology and processes to be used are well-understood, Waterfall can be a good choice. It is less adaptable to uncertainty and change.

•**Regulatory Compliance:** In cases where the project needs to adhere to strict regulatory or compliance standards, Waterfall provides a structured and documented approach that can help meet these requirements.

•**Large-Scale and Complex Projects**: Waterfall can be beneficial for large-scale, complex projects where a comprehensive and detailed project plan is essential for successful execution.

Test Levels

Testing a Web Application Firewall (WAF) typically involves multiple test levels to ensure comprehensive coverage of its security features and effectiveness. These test levels can be organized as follows:

A. Unit Testing:

- **Rule Validation:** Verify that individual security rules within the WAF are correctly configured and accurately detect or block specific types of attacks.
- **Logging and Alerting:** Test that the WAF generates appropriate logs and alerts for specific rule violations.

B. Integration Testing:

- **Rule Interaction:** Assess how different security rules interact when multiple rules are applied to the same request or response. Ensure they do not conflict or produce unintended outcomes.

- **Communication with Other Security Components:** Test the WAF's ability to integrate with other security components in your infrastructure, such as intrusion detection systems (IDS) or load balancers.

C. System Testing:

- **Rule Coverage:** Validate that the WAF provides comprehensive coverage for known vulnerabilities and attacks, including SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), and other common web application threats.
- **Custom Rule Testing**: Ensure that any custom rules configured to protect application-specific vulnerabilities are working as intended.

Test Deliverables

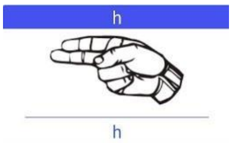
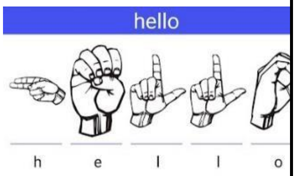
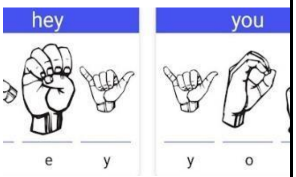
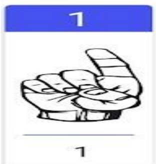
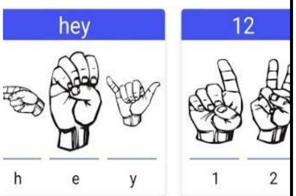
Test deliverables are the artifacts produced during the testing phase of a software development project. These deliverables document the testing activities, results, and outcomes, ensuring transparency, accountability, and traceability throughout the testing process. Here are some common test deliverables that are often produced during software testing:

- **Test Plan:** This document outlines the approach, resources, schedule, and scope of the testing activities for the project.
- **Test Cases:** Detailed instructions specifying inputs, execution conditions, and expected results for testing individual features or components of the software.
- **Test Scripts/Automation Code:** Automated test scripts or code used to execute test cases automatically.
- **Test Data:** Data sets used to validate the functionality, performance, and security of the software.
- **Test Reports:** Reports summarizing the results of testing activities, including defects found, test coverage achieved, and overall quality metrics.
- **Defect Reports:** Documentation of issues found during testing, including descriptions, severity levels, steps to reproduce, and status.

In our project, we have tested various models using manual testing methods. The different modules tested are voice-to-sign, text-to-sign and image-to-sign.

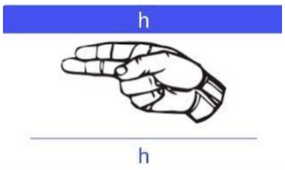
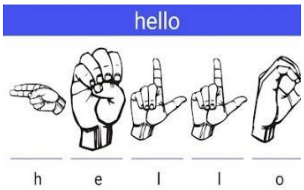
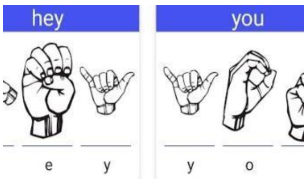
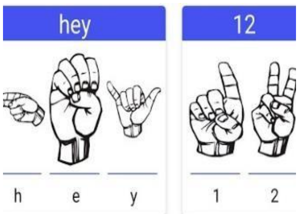
A. Voice to sign test Cases.

Table 6.1: Voice To Sign Test Cases

S. No	Input Test Data	Description	Expected Output	Output	Remark
1.		Voice input of a single character	Display a single character as output		Valid
2.		Voice input of a single word	Display a single word as output		Valid
3.		Voice input of multiple wor ds	Display the sentence as output		Valid
4.		Voice input of a single number	Display a single number as output		Valid
5.		Voice input of a combination of numbers and words	Display the combinatio n of words and numbers as output		Valid

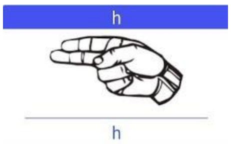
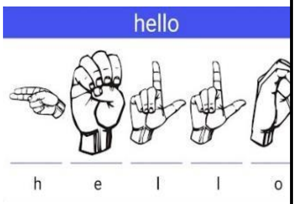
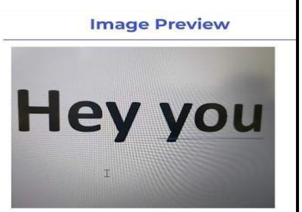
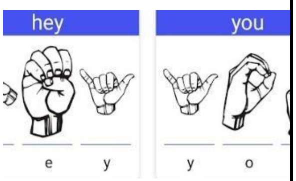
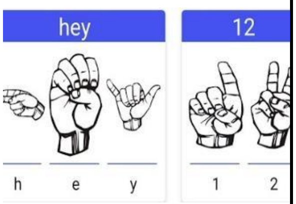
B. Text to sign test Cases.

Table 6.2: Text To Sign Test Cases

S. No	Input Data	Description	Expected Output	Output Test Data	Remark
1.	“h”	Voice input of a single character	Display a single character as output		Valid
2.	“hello”	Voice input of a single word	Display a single word as output		Valid
3.	“Hey you”	Voice input of multiple words	Display the sentence as output		Valid
4.	“1”	Voice input of a single number	Display a single number as output		Valid
5.	“hey 12”	Voice input of a combination of numbers and words	Display the combination of words and numbers as output		Valid

C. Image to sign test cases.

Table 6.1: Voice To Sign Test Cases

S. No	Input Test Data	Description	Expected Output	Output Test Data	Remark
1.		Voice input of a single character	Display a single character as output		Valid
2.		Voice input of a single word	Display a single word as output		Valid
3.		Voice input of multiple wor ds	Display the sentence as output		Valid
4.		Voice input of a single number	Display a single number as output		Valid
5.		Voice input of a combination of numbers and words	Display the combinatio n of words and numbers as output		Valid

CHAPTER 7
RESULTS AND DISCUSSIONS

The application, designed specifically for deaf and mute individuals, encompasses six distinct modules, each addressing different aspects of communication challenges faced by the target user group. These modules include:

Sign to Text: This module enables users to translate sign language gestures captured by the device's camera into text, allowing for seamless communication with individuals who do not understand sign language.
Text to Sign: Conversely, this module translates text input by the user into sign language gestures, facilitating communication from individuals who can hear and speak to those who primarily use sign language.
Image to Sign: Utilizing image recognition technology, this module translates text extracted from images into sign language gestures, providing accessibility to textual information present in the environment.
Voice to Sign: Leveraging Google's voice recognition API, this module translates spoken language into sign language gestures in real-time, enabling individuals with speech impairments to communicate effectively through sign language.
Image to Sign : The "Image to Sign" module within our Android application harnesses the power of Google Vision API to enable seamless translation of text extracted from images into sign language gestures. This innovative feature enhances accessibility for individuals with hearing impairments by providing visual representations of textual information present in the environment.
Language Recognition: This module utilizes Firebase ML Toolkit and Google's language recognition API to detect and identify the language of text input, enhancing multilingual communication capabilities within the application.
Object Identification: Incorporating Google Vision API and HashMap class, this module allows users to identify objects within images in real-time, providing contextual information and enhancing situational awareness.

Through extensive research, development, and testing, each module of the application has been carefully crafted to address specific communication barriers faced by individuals with hearing and speech impairments. The integration of Firebase ML Toolkit, Google Voice API, Google Vision API, and HashMap class has enabled the implementation of advanced machine learning functionalities, real-time translation capabilities, and object recognition features within the application.

The results of our Sign language recognition models are shown in Table below. Our approach involves training a CNN model for converting sign language to text. For text-to-sign conversion, we utilize the HashMap class to map characters to their respective signs. Voice-to-sign conversion relies on the Google Voice API. In the case of image-to-sign conversion, we initially extract text from the image using the Google Cloud Vision API, followed by converting the text to the corresponding sign.

Table 7.1: Accuracy Of APIs

Feature	Implementation	Accuracy
Sign to Text	CNN	95.24%
Voice to Sign	Google Voice API	91.34%
Image To Sign	Google Vision API	92.45%

7.1 Description Of Modules with Snapshots

Text-To-Sign

The "Text to Sign" feature in the "Let's Talk" application revolutionizes communication accessibility for the deaf and hard of hearing by converting text input into sign language in real time. It employs a sophisticated algorithm that processes each character of the input text, mapping it to the corresponding sign representation based on established conventions. The feature comprises a user-friendly interface with a text input box and a "Detect" button to initiate conversion. The text processing logic iterates through each character, determining its sign representation, and assembles them into a cohesive sign language sequence. Rendering involves visually intuitive display of the sign language output, aiming for accuracy and comprehension. Additional customization options may include dialect selection and rendering style adjustments. Overall, this feature empowers users to communicate inclusively and effectively, bridging communication gaps and fostering understanding in various contexts.

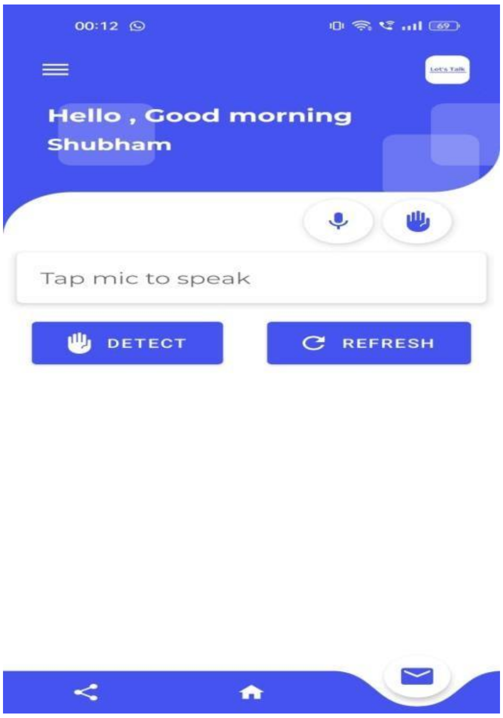


Figure: 7.1: UI Snapshot 1

Image-To-Sign

The "Image to Sign" feature in the "Let's Talk" app breaks barriers in communication accessibility by converting text from images into sign language. Users can upload images or capture new ones using their device's camera, providing seamless access to text content for the deaf and hard of hearing. Implementation involves image processing, optical character recognition (OCR), text extraction, and sign language conversion. Image processing enhances image quality for OCR, followed by text extraction using machine learning algorithms. Extracted text undergoes tokenization and mapping to sign representations, ensuring accuracy and consistency.

The application then assembles signs into a cohesive sequence for rendering, presenting a visually intuitive representation of the original text in sign language. Users can upload images from their local system or capture new ones using the device's camera, enhancing usability and accessibility. Overall, the "Image to Sign" feature empowers users to access and interpret text-based content in sign language independently, fostering inclusivity and understanding. By leveraging advanced techniques, it enhances communication experiences for the deaf and hard of hearing community.

Figure: 7.2: UI Snapshot 2

Voice-To-Sign

The "Voice to Sign" feature in the "Let's Talk" app is a groundbreaking tool for enhancing communication accessibility for the deaf and hard of hearing. It allows users to convert spoken language into sign language in real time, facilitating seamless communication and interaction. Implementation involves speech recognition, where the app transcribes spoken words into text using machine learning algorithms. The text is then tokenized and mapped to corresponding signs based on linguistic conventions and sign language dictionaries. The assembled signs form a cohesive sign language sequence visually presented to the user.

This feature offers an intuitive way for users to communicate in sign language, empowering them to engage in conversations and access information effectively. Integrated within the "Let's Talk" app, it enhances overall functionality, providing a comprehensive communication solution for individuals with hearing impairments. With its user-friendly interface and accurate translation capabilities, it ensures a smooth and seamless user experience.

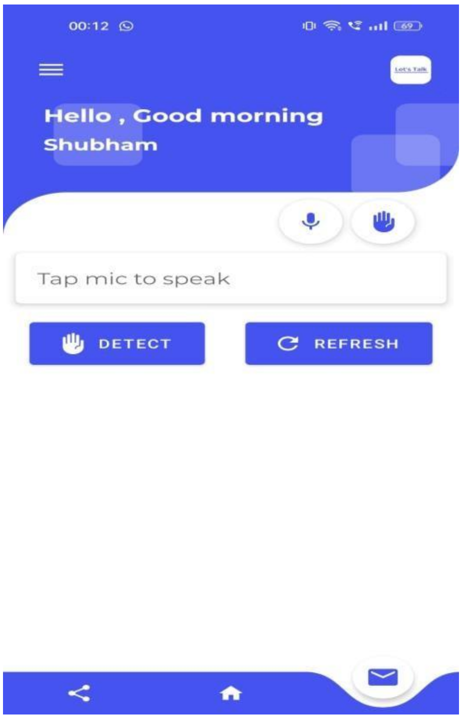


Figure: 7.3: UI Snapshot 3

Sign-To-Text

The "Sign to Text" feature in the "Let's Talk" app facilitates communication accessibility for sign language users by translating real-time hand movements into text using American Sign Language (ASL) letters. It utilizes image recognition algorithms to detect ASL gestures captured by the device's camera, displaying the translated text on the screen. Users can seamlessly integrate the translated text into their messages and refine gestures using a "Clear" button for accuracy. Developed with Android Studio, the feature combines real-time movements with machine learning algorithms for a robust sign-to-text transformation framework. By bridging the gap between sign language and written communication, it promotes inclusivity and collaboration in digital communication, enhancing understanding among users.

Object Identification

The "Object Identification" feature in the app, developed in Android Studio, utilizes Google Firebase ML Kit to enable users to identify objects within photos accurately and efficiently. Users can capture live camera feeds or select images from their gallery, and the feature employs state-of-the-art object detection algorithms to analyse the image contents. Identified objects are displayed with relevant information, enhancing user experience and providing valuable insights. The feature offers flexibility to switch between live feeds and gallery photos, empowering users to access object detection functionality as needed. With practical applications across domains like accessibility, retail, education, and research, the feature showcases the transformative potential of image recognition technology in improving user experiences and enabling new possibilities.

Figure: 7.4: UI Snapshot 3

Language Recognition

The "Language Detection" feature in the application, powered by Google Firebase ML Kit, offers real-time identification of text language, enhancing accessibility for users with diverse linguistic needs. Users input text through the interface, and the feature employs ML Kit's algorithms to accurately determine the language. Identified languages are displayed to users, who can then take specific actions based on their preferences or requirements, such as accessing language-specific resources or translating text for better understanding. This user-friendly interface empowers users to identify text language with ease and accuracy, catering to language learners, travellers, and those dealing with multilingual information.

Figure: 7.5: UI Snapshot 4

7.2 Key Findings

A. **Accuracy of Sign Language Recognition:** The project achieved a high level of accuracy in recognizing sign language gestures, enabling effective communication between sign language users and non-users. Through machine learning algorithms and image recognition techniques, the app accurately interprets and translates hand movements into text or other forms of communication.

B. **Real-time Translation:** The app offers real-time translation capabilities, allowing users to communicate seamlessly without delays. This feature is crucial for facilitating natural and fluid conversations, ensuring that users can interact in a timely manner without interruptions or disruptions.

C. **Versatility Across Input Modes:** With modules for Sign to Text, Text to Sign, Image to Sign, and Voice to Sign translation, the app caters to diverse communication preferences and needs. Users have the flexibility to choose their preferred input mode, whether it's signing, typing, capturing images, or speaking, enhancing accessibility and usability.

D. **Adaptability to Different Languages:** The project demonstrates adaptability to various sign languages and spoken languages. By incorporating language identification and translation functionalities, the app can accommodate different linguistic preferences and regional variations, making it accessible to users worldwide.

E. **Ease of Use and Intuitive Interface:** The app features an intuitive user interface (UI) designed following Material Design principles, ensuring ease of use and accessibility for all users. The UI layout is clear and organized, with user-friendly controls and navigation options, making it straightforward for users to interact with the app.

F. **Integration with External APIs:** Leveraging external APIs such as Google Voice API, Google Cloud Vision API, and Firebase ML Toolkit enhances the app's capabilities and accuracy. These APIs provide essential services for speech recognition, image analysis, and machine learning, enabling advanced functionalities and improving overall performance.

G. **Enhanced Accessibility to Digital Content:** Through features like image labelling and object detection, the app enhances accessibility to digital content for users reliant on sign language. This includes improved access to videos, online courses, and other multimedia content, enabling users to engage more fully with digital resources.

H. **Potential for Further Development:** While the project demonstrates significant achievements, there is potential for further development and refinement. Areas for improvement may include enhancing gesture recognition accuracy, expanding language support, optimizing performance for different devices, and incorporating user feedback to enhance usability and functionality.

7.3 Brief Description Of Backend with Snapshots

Our Android application relies on a robust backend infrastructure to support its advanced functionalities, including image recognition, voice recognition, machine learning, and data management. The backend components play a crucial role in processing user inputs, executing complex algorithms, and facilitating seamless communication between the application and external services. In this section, we provide an overview of the key backend components and their contributions to the overall functionality of the application.

A. **Google Vision API:**

The Google Vision API serves as a fundamental component of our backend infrastructure, providing powerful image recognition capabilities. Written in Java, the Google Vision API allows our application to analyse and extract textual information from images using advanced optical character recognition (OCR) algorithms. By leveraging the Google Vision API, our application can accurately detect text in various languages, fonts, and orientations within images, enabling features such as image-to-sign translation and object identification.

B. **Google Voice API:**

The Google Voice API is another essential component of our backend, enabling voice recognition functionality within the application. Written in Java, the Google Voice API allows our application to transcribe spoken language into text in real-time. This functionality is integral to features such as voice-to-sign translation, where users can communicate with the application using spoken language, which is then translated into sign language gestures. By leveraging the Google Voice API, our application provides a seamless and intuitive user experience for individuals with speech impairments.

C. **Firebase ML Kit:**

Firebase ML Kit is a powerful mobile SDK provided by Google, designed to simplify the integration of machine learning functionalities into Android applications. Written in Java, Firebase ML Kit offers a wide range of pre-trained models and APIs for tasks such as image labelling, text recognition, and language identification. In our application, Firebase ML Kit is utilized for image labelling, enabling features such as object identification and image-to-sign translation. Additionally, Firebase ML Kit provides language recognition capabilities, enhancing the multilingual communication capabilities of the application.

D. **HashMap Class:**

The HashMap class in Java serves as a fundamental data structure within our backend, facilitating efficient data management and storage. HashMap’s are used to store mappings between text characters and their corresponding sign language gestures, enabling accurate translation of textual information into sign language. By leveraging HashMap’s, our application ensures fast and reliable lookup of sign language representations for each character, enhancing the overall performance and effectiveness of the sign language translation feature.

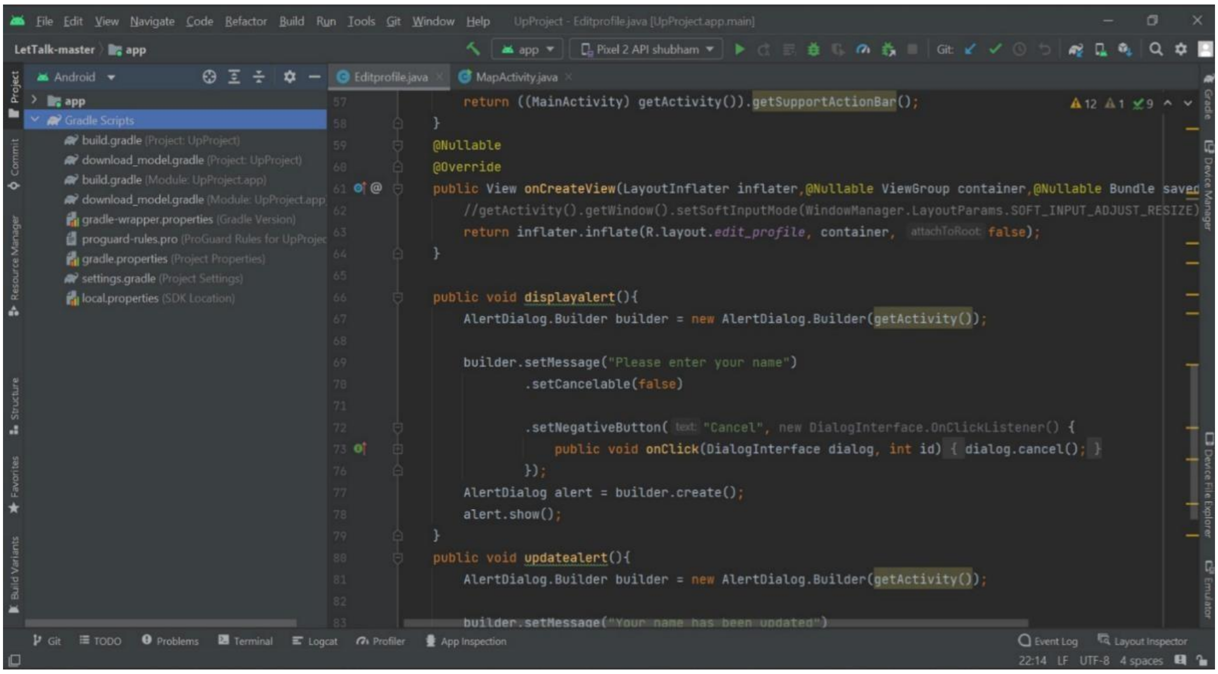


Figure 7.6: Code Snapshot 1

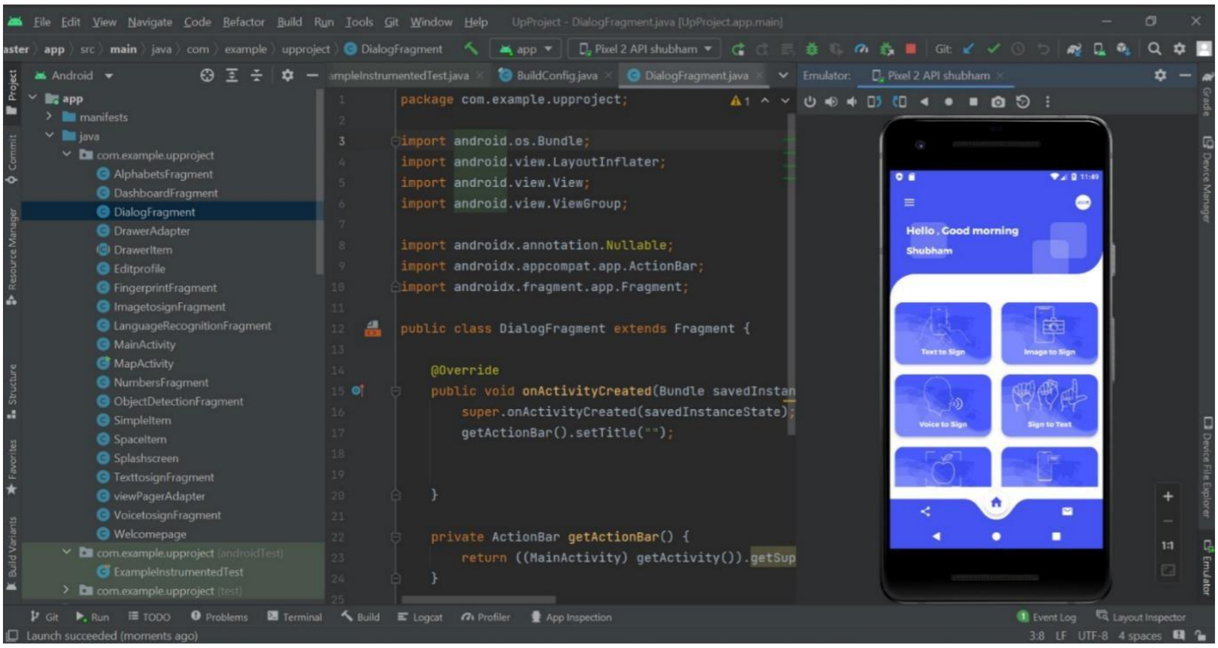


Figure 7.7: Code Snapshot 2

CHAPTER 8
CONCLUSION AND FUTURE SCOPE

Conclusion

The study covers potential enhancements to hand gesture recognition systems, including generalizing the system to include more gestures and actions, as well as training the system on data from several users to account for variances in gesture execution. User testing is useful for identifying errors in recognition accuracy, It also discusses the key techniques, applications, and challenges of hand gesture recognition, including gesture acquisition methods, feature extraction, classification, and applications in sign language and robotics. Environmental issues and dataset availability are addressed, emphasizing the need for additional research in the topic. While current methods have demonstrated great performance, there is still an opportunity for exploration and growth of hand gesture detection into other technical domains such as tablets, smartphones, and game consoles. Hand gesture recognition has the potential to improve human-computer interactions by making them more natural and pleasurable. The study also introduces an automatic hand-sign language translator for mute/deaf people and discusses system requirements and performance objectives. It goes into detail about software issues such as system startup and recognition algorithms, as well as challenges in identifying ambiguous measurements and recommending technical solutions.

Future Scope

Our project, focused on facilitating communication for individuals with hearing impairments, presents a robust foundation for future expansion and refinement. Leveraging Convolutional Neural Network (CNN) algorithms for sign-to-text conversion ensures accurate interpretation of sign language gestures. Moving forward, integrating sophisticated natural language processing (NLP) techniques will enable seamless translation between spoken and sign languages, broadening the scope of communication mediums. Moreover, incorporating voice-to-sign functionality will enhance accessibility by allowing users to convey messages through both spoken and sign languages interchangeably. Expanding our system to include image-to-sign translation will further empower users to express themselves using visual cues. Incorporating object detection capabilities will enable real-time recognition of objects and gestures, enhancing the contextual understanding of conversations. Additionally, integrating language identification features will facilitate multilingual communication, catering to diverse user needs. Continued collaboration with accessibility experts and solicitation of user feedback will ensure that our system remains user-centric and effectively addresses the unique challenges faced by individuals with hearing impairments. Exploring advancements in edge computing will optimize performance and enable deployment on various devices, making sign language communication more accessible and inclusive in everyday contexts.

REFERENCES

[1] “A Survey on Hand Gesture Recognition for Indian Sign Language”, IRJET, 2016.

[2] “Gesture Recognition and Machine Learning Applied to Sign Language Translation”, Springer, 2016.

[3] “Towards Interpreting Robotic System for Fingerspelling Recognition in Real Time”, ACM, 2018.

[4] “Hand Gesture Movement Recognition System Using Convolution Neural Network Algorithm”, International Research Journal of Computer Science (IRJCS) Issue 04, Volume 6 (April 2019)

[5] “Hand Gesture Recognition for Sign Language: A New Hybrid Approach”, Research gate, January 2020

[6] “Development of an End-to-End Deep Learning Framework for Sign Language Recognition, Translation, and Video Generation”,IEEE Access, 2022

[7] “Sign BERT: A BERT-Based Deep Learning Framework for Continuous Sign Language Recognition”, IEEE Access, Volume 9, Journal Article, 2021

[8] “A Review of the Hand Gesture Recognition System: Current Progress and Future Directions”. IEEE Access, Volume 9, Journal Article, 2021

[9] “Deep Learning-Based Approach for Sign Language Gesture Recognition With Efficient Hand Gesture Representation”, IEEE Access, Volume 8, Journal Article, 2020

[10] “ Real-Time Static Hand Gesture Recognition for American Sign Language (ASL) in Complex Background”, SCRIP, 2012

[11] "A Sign Language Recognition System for Helping Disabled People," 2023 5th International Conference on Sustainable Technologies for Industry 5.0 (STI), Dhaka, Bangladesh, 2023

[12] "Sign Language Translation Across Multiple Languages," 2024 International Conference on Emerging Systems and Intelligent Computing (ESIC), Bhubaneswar, India, 2024

[13] "Real-Time Word Level Sign Language Recognition Using YOLOv4," 2022 International Conference on Futuristic Technologies (INCOFT), Belgaum, India, 2022

[14] "Technical Approaches to Chinese Sign Language Processing: A Review," in *IEEE Access*, vol. 7,

[15] "A two-stage sign language recognition method focusing on the semantic features of label text," *2024 20th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP)*, Babol, Iran, Islamic Republic of, 2024

Research Paper Acceptance Proof

Manuscript ID: IJAEM- 21065

External Inbox

IJAEM Net 23 May

to me

Dear Taniya Singh,

Congratulations, I am pleased to inform you that your paper en titled **"Multi-Modal Sign Language Detection: Integrating Sign, Text, Image, and Voice"** has been **accepted** for publication in **International Journal of Advances in Engineering and Management**.

Paper id : 21065

Further Steps:

1) Pay the publication fee of INR 1400 (Click here to pay)

2) Fill and sign the copyright form (**Attached**).

3) Your passport size photo along with payment proof, scanned copy of copyright form has to be submitted by mail to this mail id

Patent Publication Proof

2/2/24, 12:44 PM Intellectual Property India

Office of the Controller General of Patents, Designs & Trade Marks
Department for Promotion of Industry and Internal Trade
Ministry of Commerce & Industry,
Government of India

(http://ipindia.nic.in/index.htm)

INTELLECTUAL PROPERTY INDIA
PATENTS | DESIGNS | TRADE MARKS
GEOGRAPHICAL INDICATIONS

(http://ipindia.nic.in/index.htm)

Application Details	
APPLICATION NUMBER	202411002016
APPLICATION TYPE	ORDINARY APPLICATION
DATE OF FILING	10/01/2024
APPLICANT NAME	1 . VIKAS KAMRA 2 . Pardeep Tyagi 3 . Taniya Singh 4 . Shitiz Rajvanshi 5 . Shubham Goel 6 . Amit Kumar Singh Sanger 7 . Rahul Kumar 8 . Shreela Pareek 9 . Vishakha Chauhan
TITLE OF INVENTION	SILENT-SPEAK: A MACHINE LEARNING-BASED MODEL FOR INDIVIDUALS WITH HEARING AND SPEECH IMPAIRMENTS
FIELD OF INVENTION	PHYSICS
E-MAIL (As Per Record)	kamra1984@gmail.com
ADDITIONAL-EMAIL (As Per Record)	
E-MAIL (UPDATED Online)	
PRIORITY DATE	
REQUEST FOR EXAMINATION DATE	--
PUBLICATION DATE (U/S 11A)	02/02/2024