VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANA SANGAMA", BELAGAVI-590018

TECHNICAL SEMINAR REPORT
ON

"ASL COMMUNICATION SYSTEM"

SUBMITTED IN PARTIAL FULFILLMENT FOR THE AWARD OF DEGREE OF

BACHELOR OF ENGINEERING IN

INFORMATION SCIENCE AND ENGINEERING

SUBMITTED BY

BHAVANA R

[1JB21IS021]

Under the Guidance of

Dr. Manu M N

Professor,
Dept. of ISE,
SJBIT, Bengaluru 60

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

SJB INSTITUTE OF TECHNOLOGY

BGS HEALTH AND EDUCATION CITY, KENGERI, BENGALURU-560060
KARNATAKA, INDIA.

2024 - 2025

|| Jai Sri Gurudev ||

Sri Adichunchanagiri Shikshana Trust ®

SJB INSTITUTE OF TECHNOLOGY

BGS Health & Education City, Kengeri, Bengaluru – 560 060

Department of Information Science & Engineering

CERTIFICATE

Certified that the Technical Seminar entitled "**ASL COMMUNICATION SYSTEM**" carried out by **Ms. BHAVANA R** bearing USN **1JB21IS021** is the bonafide students of **SJB INSTITUTE OF TECHNOLOGY** in partial fulfilment for the award of **BACHELOR OF ENGINEERING** in **INFORMATION SCIENCE AND ENGINEERING** of the **Visvesvaraya Technological University**, **Belagavi** during the academic year **2024-25.** It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report and deposited in the departmental library. The Technical report has been approved as it satisfies the academic requirements in respect of technical seminar prescribed for the said degree.

Signature of Guide	Signature of HOD
Dr. Manu M N	**Dr. GopalaKrishna M T**
Professor	**Professor & Head**
Dept. of ISE, SJBIT	**Dept. of ISE, SJBIT**

INTERNAL VIVA

Name of the Examiners	Signature with Date
1……………………………………	………………………
2……………………………………	………………………

ACKNOWLEDGEMENT

I would like to express our profound gratitude to **His Divine Soul Jagadguru Padma Bhushan Sri Sri Sri Dr. Balagangadharanatha Mahaswamiji** and His Holiness **Jagadguru Sri Sri Sri Dr. Nirmalanandanatha Swamiji** for providing us an opportunity to be a part of this esteemed institution.

I would also like to express our sincere thanks to **Revered Sri Sri Dr. Prakashnath Swamiji**, **Managing Director of SJB Institute of Technolog**y, for his continuous support in providing the necessary amenities to carry out this project at this esteemed institution.

I express our heartfelt gratitude to **Dr. Puttaraju**, Academic Director of the BGS and SJB Group of Institutions.

I express my gratitude to **Dr. K. V. Mahendra Prashanth**, Principal, SJB Institute of Technology, and for providing me an excellent facilities and academic ambience which have helped me in satisfactory completion of technical seminar.

I extend my sincere thanks to **Dr. GopalaKrishna M.T ,** Professor & Head, Department of Information Science and Engineering for providing me an invaluable support throughout the period of my technical seminar.

I express my truthful thanks to, Technical Seminar Coordinator, **Mrs. Poornima M**, Assistant Professor Department of Information Science and Engineering, for her valuable support.

I wish to express my heartfelt gratitude to our guide**, Dr. Manu M N,** Professor, Department of Information Science and Engineering for his/her valuable guidance, suggestions and cheerful encouragement during the entire period of my technical seminar.

Finally, I take this opportunity to extend my earnest gratitude and respect to our parents, the teaching & technical staff of the department, the library staff, and all my friends, who have directly or indirectly supported me during the period of my project work.

Regards,
BHAVANA R
[1JB21IS021]

DECLARATION

I hereby declare that the entire work embodied in this Seminar report has been carried out under the supervision of Dr. Manu M N, Professor, Dept. of Information Science and Engineering, SJB Institute of Technology in partial fulfilment for the award of "BACHELOR OF ENGINEERING" in INFORMATION SCIENCE AND ENGINEERING as prescribed by VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI during the academic year 2024-25.

BHAVANA R
[1JB21IS021]

ABSTRACT

This paper introduces an interactive and accessible desktop-based communication system designed to bridge the gap between individuals who use American Sign Language (ASL) and those who rely on spoken or written language. Recognizing the inherent challenges faced by the deaf and hard-of-hearing community—especially in real-time conversations with non-signers—the proposed solution offers a two-way translation system powered by deep learning and multimedia processing. The core of the system relies on a Convolutional Neural Network (CNN) trained on a large dataset of ASL hand gesture images (over 87,000 samples), representing the 26 alphabetic characters. The model architecture comprises three convolutional layers followed by two fully connected layers and an output layer. Preprocessing steps such as resizing to 64x64 pixels, grayscale conversion, and normalization enhance model performance and enable robustness to variation in lighting and hand orientation. Once trained, the model achieves a high validation accuracy of approximately 96%, enabling reliable gesture-to-text translation through a live webcam feed. In parallel, the system includes a reverse mode where users can input spoken words or typed text, which are then converted into ASL using pre-curated animated GIFs from a sign language database. If direct word-based GIFs are unavailable, the system dynamically switches to character-by-character animations, ensuring comprehensive coverage of the input. Both translation modes are embedded in an intuitive GUI built using Tkinter, offering features such as word suggestions, clearing functionality, and access to additional learning resources. The application operates entirely offline and is structured to work effectively in diverse environments—classrooms, customer service, and personal interaction scenarios—without the need for constant internet access or expensive hardware. The tool provides a meaningful step toward inclusive communication by combining real-time image recognition, natural language processing, and visual animation in a user-centric interface.

TABLE OF CONTENTS

List of Figures

CHAPTER 1

INTRODUCTION

1.1 Background

Communication is a fundamental human need, yet for individuals who are deaf or hard of hearing, engaging in everyday interactions can be challenging due to language barriers. Sign languages, such as American Sign Language (ASL), serve as primary modes of communication within these communities. However, the lack of widespread understanding of sign languages among the general population often leads to social isolation and limited access to essential services for those reliant on them.

Advancements in technology have opened new avenues to bridge this communication gap. In particular, the integration of computer vision and deep learning techniques has shown promise in interpreting sign language gestures. Convolutional Neural Networks (CNNs), known for their proficiency in image recognition tasks, have been employed to analyze hand gestures and translate them into textual or spoken language. This technological approach aims to facilitate more inclusive communication by enabling real-time translation of sign language.

Despite these technological strides, many existing sign language recognition systems face limitations. Some require specialized hardware, while others lack user-friendly interfaces, hindering their adoption in everyday scenarios. Moreover, the accuracy of gesture recognition can be affected by varying lighting conditions and backgrounds, posing challenges for consistent performance. Addressing these issues is crucial for developing practical solutions that can be seamlessly integrated into daily life. The current study presents a desktop-based application designed to recognize ASL gestures using a standard webcam. The system employs a CNN trained on a comprehensive dataset of ASL images, enabling it to accurately interpret hand gestures. Additionally, the application features a graphical user interface (GUI) that allows users to input text or speech, which is then translated into corresponding sign language animations. This bidirectional functionality not only assists individuals who are deaf or hard of hearing but also educates those unfamiliar with sign language, promoting mutual understanding.

1.2. Significance and Impact of ASL Sign Language Translation

1. **Significance of ASL as a Natural Language and Cultural Foundation**

- ASL is not just a communication tool but the heart of a vibrant cultural community, giving Deaf individuals a shared history, values, and means of self-expression in a visual-spatial modality that emphasizes hands, faces, and bodies.
- In the United States, over half a million people use ASL as their primary language, making it the third most commonly used language after English and Spanish.

2. **Cognitive and Educational Benefits of ASL**

- Children exposed to sign language from infancy exhibit stronger visual attention, larger vocabularies, improved language competence, and better literacy skills compared to children without early sign exposure.
- Even hearing babies who observe sign language display cognitive benefits akin to those conferred by spoken language, demonstrating ASL's universal role in linking language and thought.
- Studies indicate that learning a sign language can lead to measurable cognitive gains in hearing children, including improved memory and visual-spatial processing.

3. **Technological Innovations Driven by ASL Needs**

- Cutting-edge research in real-time sign language translation underscores the critical need for technologies that enable seamless conversation between ASL users and non-signers.
- While AI-powered ASL interpretation tools are advancing rapidly, they currently complement rather than replace human interpreters, ensuring both accuracy and cultural nuance.
- Gallaudet University's 5G-enabled helmet prototype—originally designed to display football plays via in-helmet visuals—demonstrates how ASL-inspired technologies can extend to noisy or safety-critical settings, benefiting a wide spectrum of users.

CHAPTER 2

LITERATURE SURVEY

2.1 Evolution of Sign Language Translation

The translation of sign language into text or speech has evolved significantly over the past few decades, transitioning from hardware-based systems to sophisticated software-driven solutions. The project presented in “ASL Communication System” embodies the latest stage of this evolution, demonstrating how modern deep learning and computer vision technologies can be harnessed to create efficient, accurate, and user-friendly sign language translation systems.

➢ **Early Approaches and Limitations**

- Early attempts at sign language translation primarily involved the use of sensor gloves and hardware-based solutions, where hand movements and finger bends were captured using embedded sensors. While innovative, these systems were often expensive, cumbersome, and impractical for daily use, limiting their adoption. They were also constrained in terms of vocabulary and gesture recognition, often supporting only the ASL alphabet or a small set of predefined signs.

➢ **Introduction of Image Processing Techniques**

- With the advancement of computer vision, researchers began exploring camera-based gesture recognition. This approach removed the need for wearable hardware and focused on processing hand images using techniques like background subtraction, skin color detection, and feature extraction methods such as SIFT (Scale-Invariant Feature Transform) and HOG (Histogram of Oriented Gradients). However, these methods were often computationally intensive and lacked the accuracy required for real-time applications.

➢ **Shift to Machine Learning and CNNs**

- The next significant leap came with the integration of machine learning algorithms, particularly Convolutional Neural Networks (CNNs), which offered superior performance in image classification tasks. The ASL Communication System project reflects this evolution by adopting a CNN trained on a large dataset of ASL alphabets. The architecture—comprising multiple convolutional and dense layers—enabled robust feature extraction and classification, achieving a validation accuracy of 96%.

In this project, the CNN model processes live webcam input within a defined region of interest (ROI) to recognize ASL alphabets. This represents a shift toward real-time, vision-based translation systems that are both scalable and practical for widespread use. The use of TensorFlow and Keras frameworks exemplifies the modern deep learning stack powering these applications.

➢ **Modern Implications and Future Direction**

- The document emphasizes future enhancements such as integrating continuous sign recognition, expanding to sentence-level translation, supporting multiple sign languages, and deploying the system as a mobile application. These directions mirror the current industry goals of creating fully functional, real-time sign language interpreters that operate across platforms and languages.

- In conclusion, the ASL Communication System project is a culmination of decades of research and development in sign language translation. From hardware-based recognition systems to AI-powered vision models, this evolution showcases how far the field has progressed—and how accessible and impactful such technologies can be in supporting inclusive communication.

2.2 Previous Research

1. CNN-Based Static Gesture Recognition

- **Study:** Deaf Communicator – A Python-Based ASL Tool.
- **Technique Used:** Convolutional Neural Networks (CNN).
- **Dataset:** Self-collected dataset of static ASL alphabets
- **Performance:** Achieved 95% accuracy on the test set
- **Remarks:** The model required a large amount of labeled training data. It was limited to individual alphabets, offering no support for dynamic or continuous gestures.

2. Real-Time Recognition on Embedded Platforms

- **Study:** A Real-Time System for Recognition of American Sign Language [6]
- **Technique Used:** CNN with data augmentation
- **Dataset:** Custom ASL gesture dataset
- **Performance:** Achieved 98% accuracy
- **Remarks:** The system was capable of real-time performance using a single camera. Deployed on a Raspberry Pi, it showed potential for low-cost and portable ASL recognition systems.

3. Image Processing and Feature Extraction Methods

- **Study:** Sign Language Recognition and Translation Using Image Processing Techniques.
- **Technique Used:** Background subtraction, skin color detection, SIFT, and k-Nearest Neighbor (k-NN).
- **Dataset:** Self-collected gesture dataset.
- **Performance:** 92.4% accuracy.
- **Remarks:** While effective in static gesture recognition, the system was computationally intensive and not suitable for real-time implementation.

4. Voice-to-Sign Language Translation

- **Study:** Automatic Translate Real-Time Voice to Sign Language Conversion for Deaf and Dumb People.
- **Technique Used:** Speech recognition and pretrained machine learning model.
- **Dataset:** Self-collected voice and gesture data.
- **Performance:** 94.4% accuracy.
- **Remarks:** This study focused on converting spoken input into sign language. The success of the system depended heavily on the accuracy of the speech recognition component.

5. HOG and SVM for Gesture Classification

- **Study:** Segment, Track, Extract, Recognize, and Convert Sign Language Videos to Voice/Text.
- **Technique Used:** Histogram of Oriented Gradients (HOG) and Support Vector Machine (SVM).
- **Dataset:** Self-collected sign video dataset.
- **Performance:** 96% accuracy.
- **Remarks:** The system was accurate but required high processing power, limiting its feasibility for lightweight applications.

6. Implications for the Present Study

➢ **The current project builds upon these foundational studies by:**

- Implementing a CNN-based model with a simple and intuitive GUI (Tkinter).
- Enabling bidirectional translation (Sign-to-Text and Text/Voice-to-Sign).
- Achieving 96% accuracy using a large ASL alphabet dataset.
- Enhancing real-time usability without expensive hardware or sensor-based systems.

2.3 Research Gaps and Challenges

Despite considerable progress in the field of sign language translation using deep learning and computer vision, several research gaps and practical challenges remain unaddressed. These limitations present opportunities for future improvements and innovation.

1. Limited Vocabulary and Static Gesture Recognition

- **Gap:** Most current systems, including the one in this project, focus on recognizing static gestures or alphabets (A–Z) rather than dynamic, continuous signs or full sentences.
- **Challenge:** Real-world communication requires recognition of flowing gestures, facial expressions, and contextual meaning, which static models fail to capture.

2. Lack of Support for Continuous Sign Language

- **Gap:** The majority of research, including this work, does not handle continuous signing or sentence-level grammar.
- **Challenge:** Continuous sign language involves transitions, motion blur, and coarticulation effects that are difficult to model using frame-based classification.

3. Absence of Non-Manual Cues

- **Gap:** Current CNN-based systems typically ignore non-manual cues like facial expressions, mouth movements, and body posture, which are essential components of sign languages like ASL and ISL.
- **Challenge:** Integrating multimodal data requires more advanced models and significant computational resources.

4. Dataset Limitations

- **Gap:** Many systems, including this one, are trained on the ASL alphabet dataset, which is limited in scope and lacks diverse environmental variations.
- **Challenge:** Publicly available datasets often lack diversity in lighting, backgrounds, skin tones, hand sizes, and angles—limiting model generalizability.

5. Real-Time Performance and Processing Constraints

- **Gap:** Achieving high accuracy often comes at the cost of processing speed and resource consumption.
- **Challenge:** Balancing accuracy with real-time inference on low-power devices (e.g., mobile phones) remains an engineering challenge.

6. One-Way Focus in Most Systems

- **Gap:** Many past systems are unidirectional—either sign-to-text or text-to-sign—whereas effective communication requires bidirectional translation.
- **Challenge:** Integrating both pipelines while maintaining performance and usability increases system complexity.

7. Lack of Support for Multilingual Sign Languages

- **Gap:** Most systems, including this one, focus solely on ASL, overlooking regional sign languages such as ISL (Indian Sign Language), BSL (British Sign Language), etc.
- **Challenge:** Each language has different syntax, gestures, and cultural nuances, requiring individual models and datasets.

8. User Accessibility and Interface Design

- **Gap:** Many systems focus on backend accuracy while neglecting user interface simplicity, especially for the deaf community or elderly users.
- **Challenge:** Interfaces must be intuitive, language-inclusive, and visually engaging to ensure usability across all demographics.

9. Absence of Standard Benchmarking

- **Gap:** There is no universally accepted benchmark or evaluation protocol for sign language recognition tasks.
- **Challenge:** Comparing system performance across different datasets and experimental setups becomes difficult and inconsistent.

10. Ethical and Cultural Considerations

- **Gap:** Few projects address the cultural importance of sign languages and involve the Deaf community in system design and validation.
- **Challenge:** Developing inclusive tools requires community participation, ethical data collection, and respect for Deaf culture.

CHAPTER 3

SYSTEM OVERVIEW AND DESIGN

3.1 System Overview

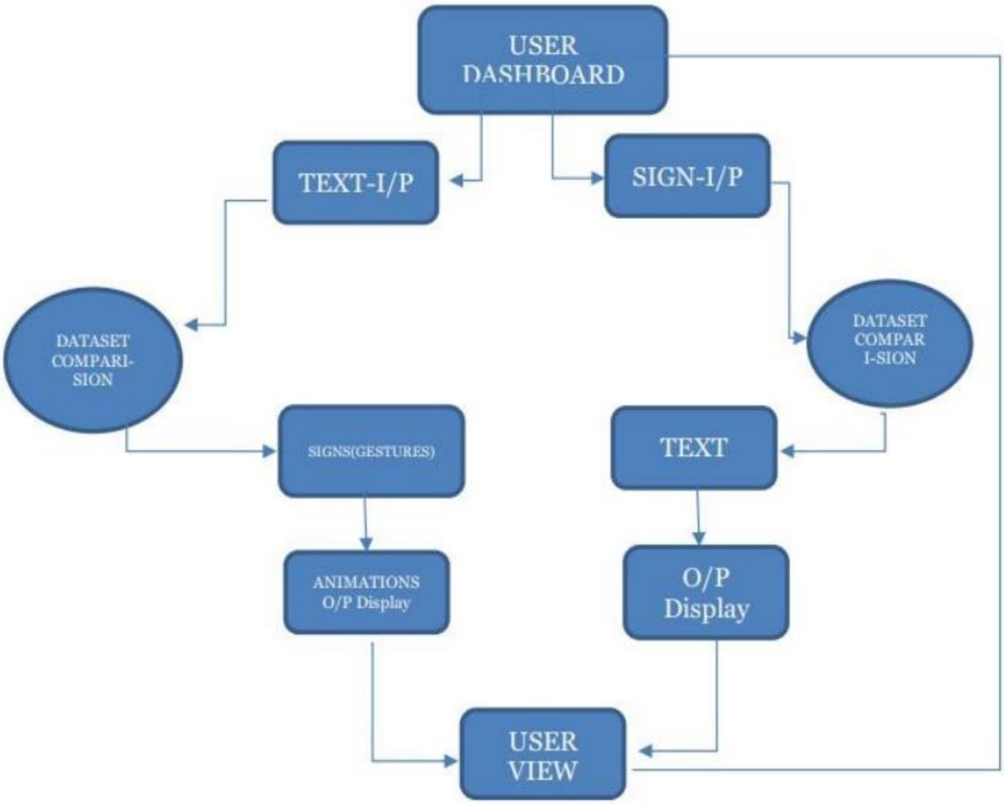


Fig 3.1 System Architecture

The system designed in this project is a bidirectional American Sign Language (ASL) communication platform that bridges the communication gap between the hearing and speech-impaired community and those unfamiliar with sign language. It performs gesture recognition through computer vision and deep learning, while also providing the ability to translate text or voice into sign language animations.

1. **System Modules** The system is divided into two primary functional modules:

a. **Gesture to Text and Voice (ASL to English):**

- This module uses a Convolutional Neural Network (CNN) to recognize ASL alphabets performed by a user in front of a webcam.
- The recognized letter is immediately converted to text and also spoken aloud using a Text-to-Speech (TTS) engine.
- This facilitates communication where the signer's hand gestures are translated into understandable speech for non-signers.

b. Text/Voice to Sign Animation (English to ASL)

- In the reverse direction, the system accepts typed text or spoken voice input.
- It then maps each letter or word to a pre-recorded ASL animation, helping the hearing/speaking person communicate visually with someone who understands only sign language.

2. Input Mechanism The system supports three types of user input:

a. Webcam for Gesture Input

- Users perform ASL gestures in front of a standard webcam.
- The webcam captures static hand gesture images in real-time.
- These images are then passed through a series of processing steps to ensure compatibility with the trained CNN model.

b. Text Input

- Users can type English letters or words into a text field in the graphical interface.
- This input is used for translation into ASL sign animations.

c. Voice Input

- Voice is captured through a microphone and converted into text using a speech recognition module.
- Once converted, it is processed in the same way as typed text

3. Image Preprocessing Pipeline: To maintain consistent model performance and reduce noise from environmental conditions, all images go through a preprocessing pipeline before classification:

- Grayscale Conversion: Reduces image complexity and model size.
- Resizing: Images are resized to a consistent size of 64x64 pixels.
- Normalization: Pixel values are scaled between 0 and 1 to improve training convergence.
- Augmentation (for training): During model training, gestures are augmented via rotation, zooming, and flipping to improve generalization across lighting and hand variations.

4. CNN-Based Gesture Recognition Model

The core recognition engine is a 2D CNN trained on a public dataset containing 87,000+ labeled ASL images, each corresponding to one of the 26 letters in the English alphabet.

- Model Architecture Includes:
- Convolutional Layers: Extract spatial features such as edges and curves.
- Activation Functions (ReLU): Add non-linearity to model complex patterns.
- Max Pooling Layers: Reduce the dimensionality while preserving key features.
- Flatten Layer: Transforms multidimensional data into 1D before feeding to dense layers.
- Dense (Fully Connected) Layers: Perform classification into one of 26 classes (A-Z).
- Softmax Output Layer: Outputs a probability distribution over possible classes.

Performance: The model achieves an accuracy of 96%, making it highly suitable for real-time applications with minimal misclassification.

5. Text-to-Speech (TTS) Conversion

- Once the CNN identifies a gesture and maps it to a letter:
- The text is passed to the TTS engine using Python’s pyttsx3 library.
- The engine converts the text to audible speech using a synthetic voice.

- This output helps the hearing person understand what the signer has expressed.

6. Text/Voice to Sign Animation Translator

For the reverse process:

- Text or voice input is analyzed, and each recognized word is matched with a pre-recorded ASL animation.
- These animations are stored locally and played within the application, simulating a person signing the words.

Fallback to Alphabet Mode:

- If a word is not available as a whole animation, the system spells it out by playing letter-by-letter sign animations.

7. Graphical User Interface (GUI)

- Developed using Python’s Tkinter framework, the GUI provides:
- Easy navigation between modules.
- Real-time display of recognized gestures.
- Option to switch between gesture recognition and text-to-sign modes.
- Buttons to record audio, type text, or activate the camera.
- Audio playback of recognized gestures.
- A window for displaying video-based ASL output.

8. Output Mechanism: The system outputs results in three forms:

a. Textual Output: Displayed in the GUI for reference after a gesture is recognized.

b. Audio Output: Generated from the recognized text using TTS.

c. Video Sign Output: For text or speech input, ASL sign animations are played from the local video dataset.

9. System Requirements and Environment

- Platform: Windows-based desktop application.
- Languages Used: Python
- Libraries/Frameworks:
- OpenCV (image capture and processing)
- TensorFlow/Keras (CNN model)
- pyttsx3 (TTS)
- SpeechRecognition (voice input)
- Tkinter (GUI)
- Hardware:
- Standard webcam
- Microphone
- No special sensors required

10. Real-World Applications

- Education: Helps teachers communicate with hearing-impaired students.
- Public Services: Enables communication in hospitals, banks, and government offices.
- Daily Interaction: Encourages seamless integration of sign language users into mainstream communication.

3.2 System Design

The project is divided into two parts:

VOICE/TEXT TO SIGN LANGUAGE CONVERSION:

- Scraping Data from Giphy using Chrome Extension
- Then filtered the gif files and added names
- Also added gif files of single alphabets

- Took Voice/Text input from the user and split into words and checked if it is present in the GIF filenames. If it is not present then use the Alphabet GIFs for making up words
- Finally Displayed it onto Tkinter App

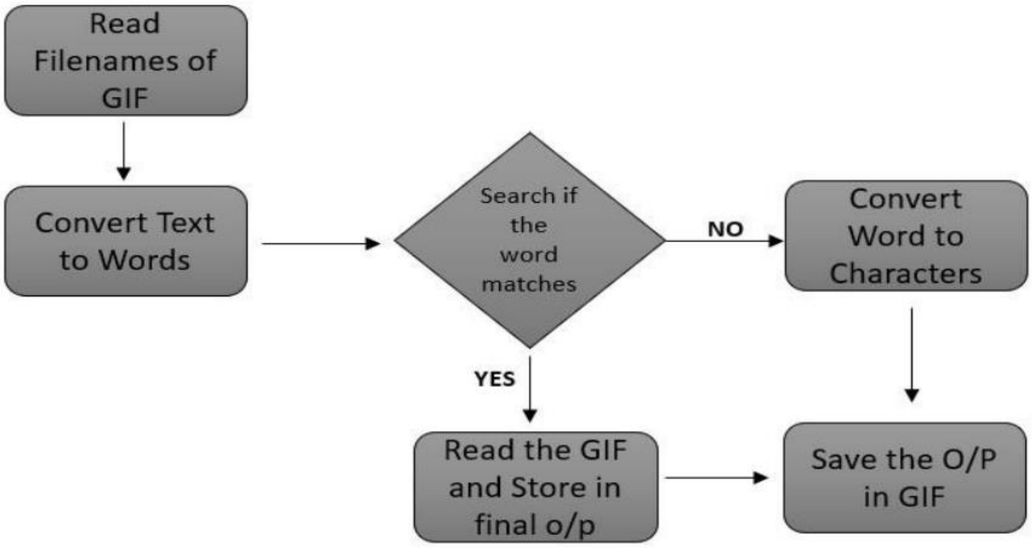


Fig 3.2 TEXT to SIGN conversion block diagram

SIGN LANGUAGE TO VOICE/TEXT CONVERSION:

- Used the ASL Dataset on Kaggle of Alphabets
- Created a CNN algorithm in Tensorflow and trained the model for small data
- Used Live Webcam feed of user hand and predicted the Alphabet from a Region of Interest
- Finally Displayed it onto Tkinter App

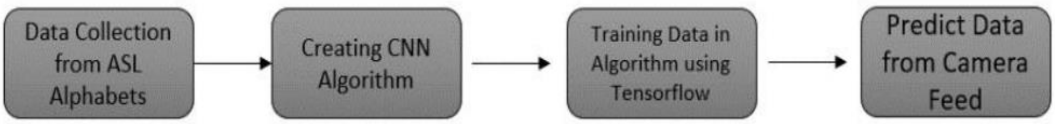


Fig 3.3 SIGN to TEXT conversion block diagram

CHAPTER 4

METHODOLOGY AND RESULT

4.1 Methodology

The methodology used for this project involves a series of steps, including data collection, data preprocessing, model building and training, and web application development. The goal of the project is to build a Sign Language Translator that can recognize ASL letters in real-time and display the corresponding letter on the screen.

The first step in the methodology is data collection. In this project, the dataset used for training the model is the ASL alphabet dataset, which includes 87000 images of 26 letters in the American Sign Language alphabet. The dataset is downloaded from Kaggle and consists of 26 subfolders, one for each letter of the alphabet. Each subfolder contains several images of the corresponding letter, captured from different angles and lighting conditions.

The second step is data preprocessing, which involves preparing the dataset for model training. In this project, the images are resized to 64x64 pixels and converted to grayscale to reduce the number of input channels and improve the model's performance. The images are then normalized by dividing each pixel value by 255, which scales the values to the range of 0 to 1.

The third step is model building and training, which involves designing a convolutional neural network (CNN) to recognize the ASL letters in the images. The CNN architecture used in this project consists of three convolutional layers, two fully connected layers, and an output layer with 26 neurons representing the 26 letters of the alphabet. The model is trained using the Adam optimizer and categorical cross entropy loss function on a training set of 80% of the data and validated on the remaining 20% for 30 epochs with a batch size of 32. After training, the model is saved and can be loaded for future use.

To ensure the quality of the project, several testing methods were implemented, including unit testing and acceptance testing. Unit testing was used to test individual components of the system, such as the model and the web application code, to ensure they are working as expected. Acceptance testing was used to test the entire system to ensure that it meets the requirements and specifications of the project. Test cases were developed for each testing method to ensure that all aspects of the project were thoroughly tested.

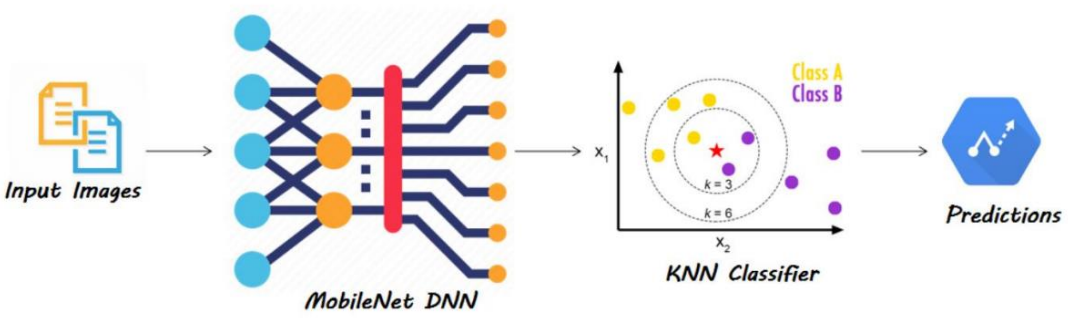


Fig 4.1 Alphabets machine learning model

Overall, the methodology used for the Sign Language Translator project involves a series of steps, including data collection, data preprocessing, model building and training, and web application development. The user-friendly interface was developed using Flask, and testing methods such as unit testing and acceptance testing were implemented to ensure the quality of the project. This project provides a useful tool for real-time ASL letter recognitionThe results demonstrated that the ReLU activation function not only speeds up convergence but also results in a more stable training process, leading to higher overall accuracy.

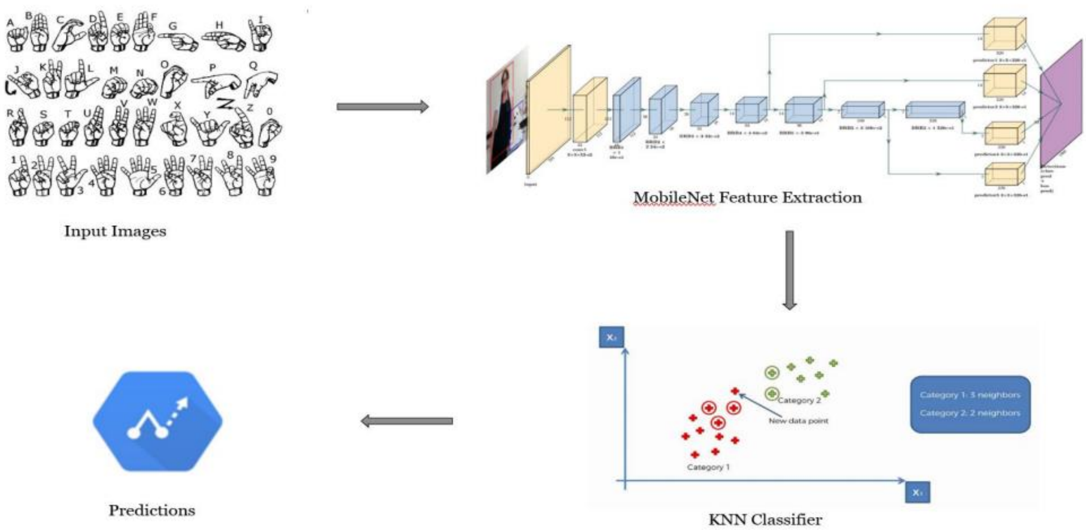


Fig 4.2 Practice phrases machine learning model

4.2 Result Analysis

Based on python modules and datasets developed in this project work, this application allows people with hearing disabilities to communicate naturally with others. These individuals convey information to deaf people by using various input methods. A direct mode of communication has been developed between natural language speakers and the deaf. A deaf person would not only save time and effort in communicating but would also bridge communication gaps. The features provided in this project allow the users both natural language speakers and the deaf and dumb to use different options like sign to text or text to sign as per their requirements making it very simple to operate for both. The text to sign API allows natural language users to type a text which the machine interprets and converts to sign language for deaf and dumb, whereas the sign to text API allows the deaf and dumb to input sign language via video and the machine interprets it and converts it to normal text for deaf and dumb people.

Then we can use this to detect our hands and convert the hand gestures to text (o/p) which would help the user to understand the signs. The application will work for both the end for the users to understand sign language as well as normal language through text which would convert itself into sign animation gestures and in turn help the user to have an interface that is very interactive. The recognition rates are calculated for different video environments. The main purpose of this project is to fill the communication gap between differently abled people and normal people and to eliminate the need of a mediator during communication between differently abled people and normal people

	Predicted Positive	Predicted Negative
Actual Positive	40 (True Positive	5 (False Negative)
Actual Negative	5 (False Positive)	50 (True Negative)

Table 6.1 Confusion matrix scenario

$$Precision = \frac{True\ Posiive}{True\ Positive + False\ Positive}$$

The Accuracy/precision rate is calculated by dividing the number of correct predictions by the total number of predictions. The accuracy of the model is evaluated on both training and validation sets after each epoch. This project achieves an accuracy of 96% on the validation set after training the model on 80% of the ASL dataset. The high accuracy of the model makes it a reliable tool for real-time ASL letter recognition, which can effectively aid in communication for individuals with speech or hearing impairments. Overall, this project provides an efficient and accessible tool for real-time ASL letter recognition, bridging the communication gap for individuals with speech or hearing impairments

CHAPTER 5

CONCLUSION

5.1 Summary Of Findings

The proposed mobile application represents a significant advancement in the field of sign language recognition, particularly focusing on the American Sign Language (ASL) alphabet. Using a Convolutional Neural Network (CNN), the system demonstrates an efficient method to translate static hand gestures into text and speech in real time. By leveraging deep learning and the ubiquity of smartphones, the system addresses a pressing need for more inclusive communication tools for the deaf and hard-of-hearing communities.

The achieved recognition accuracy of approximately 96% showcases the effectiveness of the model in classifying hand gestures under controlled conditions. The dual output feature—textual and speech—ensures that communication is not only clear for non-signers but also seamless and fast. The application runs smoothly on standard mobile hardware, making it highly accessible and deployable in everyday situations. Moreover, the use of offline processing ensures privacy and usability even in areas with poor internet connectivity.

This solution can be a useful aid in education, healthcare, and public service scenarios, where communication barriers often hinder service delivery for hearing-impaired individuals. Overall, the application highlights the practical utility of AI-powered sign language recognition and sets a strong foundation for more comprehensive future solutions.

5.2 Challenges And Limitations

Developing an ASL recognition system using CNNs on mobile devices presents numerous challenges that affect both technical performance and user adoption. A major difficulty lies in the natural variability of hand gestures among users—differences in hand size, skin tone, and individual signing styles make it hard for models to generalize. Environmental factors such as inconsistent lighting and cluttered backgrounds further complicate accurate recognition. Additionally, the scarcity of diverse, high-quality ASL datasets limits model robustness. The current system's support is often restricted to static gestures like alphabet signs, excluding dynamic gestures or full sentences, which require more complex temporal modeling. Real-time performance on mobile devices also imposes constraints, as deep models may cause latency or consume excessive power. Moreover, most systems overlook non-manual cues like facial expressions and body posture, which are essential for interpreting the full meaning of ASL. Gestures that appear visually similar can lead to frequent misclassifications, and designing an interface that is intuitive for all users, especially those with low tech literacy, adds further complexity. The absence of standardized datasets and benchmarks hinders consistent evaluation and comparison across systems. Finally, real-time camera use introduces privacy and ethical concerns, requiring careful design to protect user data and ensure trust in the application. Together, these challenges highlight the technical and societal hurdles that must be addressed to build a truly inclusive and effective ASL recognition solution.

5.3 Conclusion

The Accuracy of this model is 96%. Although the facial expressions express a lot during communication, the system does not focus on facial expressions. In conclusion, the Sign Language Translator project successfully developed a vision-based application that can recognize American Sign Language letters in real-time. The project utilized a convolutional neural network (CNN) for training the model and Flask for developing the user-friendly web application interface. The system provides an accessible tool for individuals with speech and hearing impairments to communicate effectively with others. The project achieved an overall accuracy of 96.03% on the validation set after 30 epochs of training. The application interface also includes a "Learn More" button, providing additional information about the dataset and the application itself, making it user-friendly and accessible for everyone.

5.4 Future Of Research Directions

Although the project has achieved its primary goal, there is still room for improvement and future enhancements. Some of the potential future enhancements that can be implemented in the project are:

- Extending the dataset: The current model was trained on a limited dataset of ASL letters. In the future, more images of ASL letters can be collected to extend the dataset and improve the accuracy of the model.
- Introducing other languages: The current project focuses on ASL letters only. However, other sign languages can be added to the system to make it more comprehensive.

- Improving accuracy: Although the model's accuracy is promising, it can still be improved by using more advanced techniques, such as transfer learning, to extract more features from the images.
- Incorporating full sentences: Currently, the project only recognizes individual letters of ASL. In the future, it can be enhanced to recognize full sentences, making it even more useful for people with hearing or speech impairments.
- Real-time video translation: Currently, the project only accepts images of ASL letters as input. In the future, it can be enhanced to accept real-time video input, making it more interactive and user-friendly.
- Mobile application: A mobile application version of the project can be developed to make it more accessible to users on-the go.

REFERENCES

[1] P.V.V. Kishore, Panakala Rajesh Kumar, “Segment, Track, Extract, Recognize and Convert Sign Language Videos to Voice/Text,” International Journal of Advanced Computer Science and Applications, January 2021

[2] Shraddha R. Ghorpade, Prof.Surendra K. Waghmare, “A Communication System for Deaf and Dumb People,” International Journal of Science and Research (IJSR), September 2020 © 2023 IJNRD | Volume 8, Issue 6 June 2023 | ISSN: 2456-4184 | IJNRD.ORG IJNRD2306619 International Journal of Novel Research and Development (www.ijnrd.org) g161

[3] Prof. Abhishek Mehta, Dr. Kamini Solanki, “Automatic Translate Real-Time Voice to Sign Language Conversion for Deaf and Dumb People,” International Journal of Engineering Research & Technology (IJERT), Special Issue – 2021

[4] Md. Moklesur Rahman; Md. Shafiqul Islam; Md. Hafizur Rahman; Roberto Sassi, A New Benchmark on American Sign Language Recognition using Convolutional Neural Network Dhaka, Bangladesh, 2019.

[5] Soumen Das; Saroj kr. Biswas Manomita Chakraborty “A Review on Sign Language Recognition (SLR) System: ML and DL” for SLR Visakhapatnam, India, 2021

[6] R. Harini; R. Janani; S. Keerthana; S. Madhubala; S. Venkatasubramanian, “Sign Language Translation” 6th International Conference on Advanced Computing and Communication Systems (ICACCS), India, 2020

[7] Vibhu Gupta; Mansi Jain; Garima Aggarwal, “Sign Language to Text for Deaf and Dumb”, 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence), India, 2021

NPTEL ONLINE CERTIFICATION

Course Name: Software Testing

Course Code: SEM1NOC25

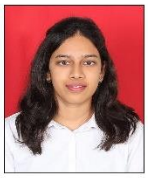
Elite

NPTEL ONLINE CERTIFICATION

(Funded by the MoE, Govt. of India)

Skill India

कौशल भारत-कुशल भारत



This certificate is awarded to

BHAVANA R

for successfully completing the course

Software Testing

with a consolidated score of **71** %

Online Assignments	20.17/25	Proctored Exam	51.26/75

Total number of candidates certified in this course: **2466**

Jan-Feb 2025

(4 week course)

Prof. Haimanti Banerji

Coordinator, NPTEL

IIT Kharagpur

Indian Institute of Technology Kharagpur

FREE ONLINE EDUCATION swayam

शिक्षित भारत, उन्नत भारत

Roll No: NPTEL25CS66S336404179 To verify the certificate No. of credits recommended: 1 or 2