VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANASANGAMA,BELAGAVI-590 018,KARNATAKA

PROJECT REPORT ON

"SignSpeak AI : A Real-Time Sign Language Translator Using LLM And Deep Learning "

Submitted in the partial fulfillment of requirements for

Major Project Phase 1(BCS685)

B.E. in Computer Science and Engineering

PROJECT ASSOCIATES
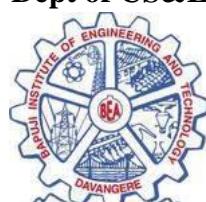
AKASH G C	**4BD22CS007**
B M DEVIKA	**4BD22CS022**
IRAMMA S G	**4BD22CS058**
SHREEKANTH K UTTARKAR	**4BD22CS151**

PROJECT GUIDE

Dr. Naveen H M *Ph.D*

Associate Professor

Dept of CS&E

BAPUJI INSTITUTE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DAVANAGERE-577004

2024-2025

ABSTRACT

Communication barriers between the deaf and hard-of-hearing community and the general population present a significant social and accessibility challenge. Sign language, while rich and expressive, is not widely understood outside these communities, limiting inclusive interaction. This project aims to bridge this gap through the development of a real-time Sign Language Translator using computer vision and deep learning techniques. The system captures static hand gestures representing the American Sign Language (ASL) alphabet via a webcam and uses a Convolutional Neural Network (CNN) model for classification. A custom dataset of over 40,000 images was used to train the model, which was preprocessed and split into training, validation, and testing sets. The trained model achieved high accuracy and was integrated into a live webcam-based application that translates recognized signs into corresponding text. The system is designed to be scalable, user-friendly, and adaptable to various environments, making it a practical tool for improving accessibility and fostering inclusive communication.

CHAPTER 1: INTRODUCTION

1.1 Introduction

Communication is the foundation of human interaction, and for individuals who are deaf or hard of hearing, sign language serves as an essential mode of expression and understand- ing. However, one of the persistent challenges faced by the hearing-impaired community is the lack of widespread knowledge of sign language among the general population. This communication barrier not only hinders social inclusion but also limits opportunities in ed- ucation, employment, healthcare, and public services. Despite advancements in technology, there remains a significant gap in the availability of intelligent systems that can interpret sign language in real-time and convert it into spoken or written language that is universally understood.

The SignSpeak AI project addresses this challenge by introducing an intelligent, real- time sign language translator powered by deep learning and computer vision technologies. The goal of the system is to empower the hearing and speech-impaired by providing them with a practical, non-intrusive solution that can facilitate seamless communication with non-signers. Unlike conventional sign language solutions that focus only on static gestures or require expensive equipment such as gloves or motion sensors, SignSpeak AI uses just a standard camera to capture gestures and interpret them through AI-powered models. This makes it both affordable and accessible to a wide range of users.

At the core of SignSpeak AI is a deep learning model that recognizes hand gestures, facial expressions, and body posture to interpret sign language in context. The system leverages Convolutional Neural Networks (CNNs) for visual gesture recognition and Long Short-Term Memory (LSTM) or Transformer-based models for sentence formation and con- textual translation. This two-stage approach ensures not only accurate gesture detection but also coherent natural language generation that reflects the intended meaning of the signer. The real-time processing capability allows for smooth conversations, eliminating the need for a human interpreter in many everyday situations.

In conclusion, SignSpeak AI is not just a translator but a bridge between two worlds—an innovative technological intervention aimed at giving voice to the silent language of signs.

CHAPTER 2: LITERATURE SURVEY

2.1 Review of Research Papers

[1] Word-level Deep Sign Language Recognition from Video

This paper introduces WLASL, the largest public word-level ASL video dataset comprising over 21,000 videos covering 2,000 distinct signs performed by 119 signers. The authors evaluate both visual appearance-based and pose-based deep learning models for sign recognition, achieving up to 62.63

[2] A Simple Multi-Modality Transfer Learning Baseline for Sign Language Translation

This paper presents a progressive transfer learning approach to improve sign language translation (SLT) by decoupling it into two tasks: Sign-to-Gloss and Gloss-to- Text. The authors pretrain models on large general-domain datasets and fine-tune them with domain-specific gloss and text corpora. They introduce a Visual-Language Mapper (V- L Mapper) to link visual and textual features, enabling end-to-end training. Their simple but effective framework outperforms previous SLT baselines on benchmarks like PHOENIX- 2014T and CSL-Daily, setting a strong foundation for future SLT research.

[3] SLTUNet: A Simple Unified Model for Sign Language Translation

SLTUNet is a unified neural model designed to jointly handle multiple sign language tasks including Sign-to-Gloss, Gloss-to-Text, and Sign-to-Text translation. It leverages task-specific encoders with a shared decoder to encourage cross-modality and cross-task knowledge trans- fer. The model achieves state-of-the-art performance on CSL-Daily and performs com- petitively on the larger and more challenging DGS Corpus. By integrating optimization strategies and multilingual training data, SLTUNet demonstrates how unified modeling can significantly boost performance in sign language translation tasks.

[4] Signs as Tokens: A Retrieval-Enhanced Multilingual Sign Language Generator

This work presents SOKE, a novel multilingual sign language generation model that translates spoken text into sign language using autoregressive generation. The model tok- enizes sign motions across body parts using a VQ-VAE-based decoupled tokenizer and inte- grates these into a pretrained multilingual language model. A multi-head decoding strategy

and retrieval-augmented mechanism using external sign dictionaries improve the naturalness and precision of generated signs. SOKE supports American, Chinese, and German Sign Languages, achieving state-of-the-art results across several multilingual SLG benchmarks.

[5] Development of a Sign Language Recognition System Using Machine Learning

 This study develops a real-time static ASL alphabet recognition system using a CNN trained on 44,654 images captured via webcam. The CNN architecture features three convolutional layers and a SoftMax output layer, achieving a test accuracy of 94.68

[6] Real-Time Sign Language Detection

 This project focuses on building a real-time ASL alphabet recognition system using computer vision and a CNN model. A dataset was created, preprocessed, and used to train a model that segments gestures, extracts features, and maps them to corresponding text. Though limited in scope, the system achieved about 95

[7] Towards Continuous Sign Language Recognition with Deep Learning

This paper addresses continuous sign language recognition by combining heuristic-based video segmentation with stacked LSTMs for sign classification. The method distinguishes between meaningful signs and translational motions (epenthesis), achieving over 80

2.2 Literature Survey Summary

2.3 Existing System

Sensor-Based Gloves: Use wearable sensors to detect finger motion but are costly, uncomfortable, and not practical for daily use. Google Teachable Machine: Allows basic sign training using a webcam but lacks dynamic gesture support and sentence translation. Sig- nAll: Provides accurate ASL translation using multiple cameras but is expensive and limited to specific setups. Mobile Apps: Many mobile apps recognize static signs or alphabets but often struggle with accuracy and cannot interpret continuous gestures. Lack of Explainabil- ity: Existing AI tools rarely explain how they interpret gestures, making it hard to trust or debug results.

2.4 Problem Statement

Sign language is an essential communication tool for the deaf and hard-of-hearing commu- nity, yet there is often a significant communication gap between sign language users and those unfamiliar with it. This lack of understanding hinders social interactions, education, and employment opportunities. The goal of this project is to develop a Sign Language Translator that uses computer vision and machine learning to automatically translate sign language gestures into text or speech, enabling real-time communication between sign lan- guage users and non-users, thus promoting inclusivity and accessibility in various settings such as education, healthcare, and public services.

2.5 Proposed System

The Sign Language Translator is designed to bridge the communication gap between sign language users and non-sign language speakers by translating sign language gestures into text or speech in real time. The system utilizes a combination of hardware and software to accurately recognize and process sign language gestures. The hardware setup includes a high-resolution camera or depth sensor, which captures the gestures performed by the user. The captured data is then processed by the software, which employs computer vision and machine learning techniques to identify and classify the gestures. The system architecture is modular, consisting of several key components. First, the gesture capture module records the gestures through a camera, providing continuous data input. The preprocessing module then cleans and normalizes the data to focus on the most relevant features, such as hand position, movement, and orientation. Following this, the gesture recognition module utilizes a trained machine learning model to identify the specific gestures. Once the gesture is recognized, the translation module maps the identified gesture to a corresponding word or phrase in text or speech. The output is then displayed to the user on the screen or read aloud through the system's text-to-speech engine. A simple user interface is employed to allow users to interact with the system and receive translated outputs with ease.

2.5 Objectives

- To develop a Sign Language Translator that accurately converts sign language gestures into text or speech.

- To implement a real-time translation system using computer vision and machine learn- ing techniques.

- To ensure the system supports various sign language types and is adaptable to different gestures.

- To create a user-friendly interface for both sign language users and non-users to facil- itate communication.

- To improve accessibility and inclusivity for the deaf and hard-of-hearing community in diverse settings.

CHAPTER 3 : SYSTEM REQUIREMENTS AND SPECIFICATION

Functional Requirements The functional requirements for the Sign Language Translator de- fine the essential features that the system must support to ensure accurate, real-time trans- lation of sign language gestures. The primary functional requirements include: Gesture Capture: The system must capture sign language gestures through a high-resolution camera or depth sensor, with the ability to process hand movements and positions in real time. Gesture Recognition: The system must accurately recognize a wide range of sign language gestures,using machine learning algorithms trained on diverse datasets to handle variations in hand shapes, movements, and facial expressions. Translation to Text/Speech: Once the gesture is recognized, the system must translate it into corresponding text or speech output, displaying the text on a screen or converting it into audible speech using a text-to-speech en- gine. Real-Time Processing: The system must ensure minimal latency between gesture input and translation output to facilitate smooth, real-time communication without noticeable de- lays. User Interface: The system must provide an intuitive, user-friendly interface, allowing users to interact with the system easily and view translated text or hear the speech output

These functional requirements aim to create a system that provides an accessible and seam- less translation experience, fostering effective communication between sign language users and non-sign language speakers.

Non-functional and Performance Requirements Non-functional requirements specify the system's operational attributes that are crucial for its performance, reliability, and user experience. The non-functional requirements for the Sign Language Translator include: Per- formance: The system must provide real-time translation with minimal latency, ensuring that gesture recognition and translation occur without delays that could hinder communi- cation. Scalability: The system should be designed to handle a growing number of users and be adaptable to include additional sign languages or new gestures in the future without significant redesign. Reliability: The system must operate consistently without crashes or errors, ensuring that users can rely on it for effective communication in various environments. Usability: The user interface must be intuitive and easy to navigate, even for individuals who may not be tech-savvy, ensuring that it provides an accessible experience for all users. Security: The system must ensure the privacy of user data, especially if personal information or gestures are being captured, and comply with relevant data protection regulation.

1   **Software requirements**

Operating System: Windows 10/11, Linux Ubuntu 20.04+, or macOS.

Programming Languages: Python 3.x, C++.

Machine Learning Framework: TensorFlow 2.x or PyTorch.

Computer Vision Libraries: OpenCV 4.x, MediaPipe.

Text-to-Speech (TTS) Engine: Google Text-to-Speech or Microsoft Speech SDK

2   **Hardware requirements**

Camera/Depth Sensor: High-resolution camera (1080p or higher) or depth sensor (e.g., Microsoft Kinect, Intel RealSense) for gesture capture.

Proces- sor (CPU/GPU): Intel i7 or equivalent (CPU); NVIDIA GTX 1080 or higher (GPU) for real-time processing.

Storage(optional): Minimum 500 GB SSD for data and model storage.

Display Device(optional): Full HD (1920x1080) display or higher for clear text output.
Speakers(optional): High-quality stereo speakers with clear sound output.

CHAPTER 4: METHODOLOGY

The development of the Sign Language Translator involved a structured approach combining computer vision, image processing, and deep learning techniques. The system is designed to recognize static American Sign Language (ASL) gestures and translate them into text in real-time. The following steps outline the methodology adopted for this project: 1. Data Collection A custom dataset of static ASL alphabet signs was created using a webcam. Each letter (excluding dynamic signs like 'J' and 'Z') was captured under controlled lighting conditions. The HandDetector module was used to isolate and crop the hand from the background, minimizing noise and distractions. 2. Image Preprocessing All captured images were resized to a standard dimension (e.g., 224x224 pixels) to maintain consistency. Normalization was applied to scale pixel values between 0 and 1. The dataset was also one-hot encoded for classification purposes. These steps ensured better performance and stability during model training. 3. Model Architecture A Convolutional Neural Network (CNN) was designed and implemented using the Keras framework. The model consists of: Three convolutional layers with ReLU activation and max pooling A flattening layer followed by two dense layers A Softmax output layer to classify the gestures into 24 different alphabet classes The model was compiled using the Adam optimizer and categorical cross-entropy loss function. 4. Model Training and Validation The dataset was split into training and validation sets (typically 705. Model Testing and Evaluation After training, the model was tested on a separate test dataset. Key evaluation metrics such as accuracy, precision, recall, and F1-score were calculated using scikit-learn to assess the system's performance. The model achieved high accuracy in recognizing most static ASL signs. 6. System Deployment The trained model was integrated into a live application that uses a webcam to capture real-time gestures. The system predicts and displays the corresponding alphabet on the screen. For best results, the system requires a well-lit environment and a clean background for hand detection.

Methodology - Sign Language Translator

Data Acquisition

Capture ASL gesture images using webcam

Preprocess Images

Model Design

Build CNN with 3 Conv layers + Dense layers

Model Training

Model Evaluation

Evaluate with test dataset

System Deployment

Real-time prediction using webcam

Display output text
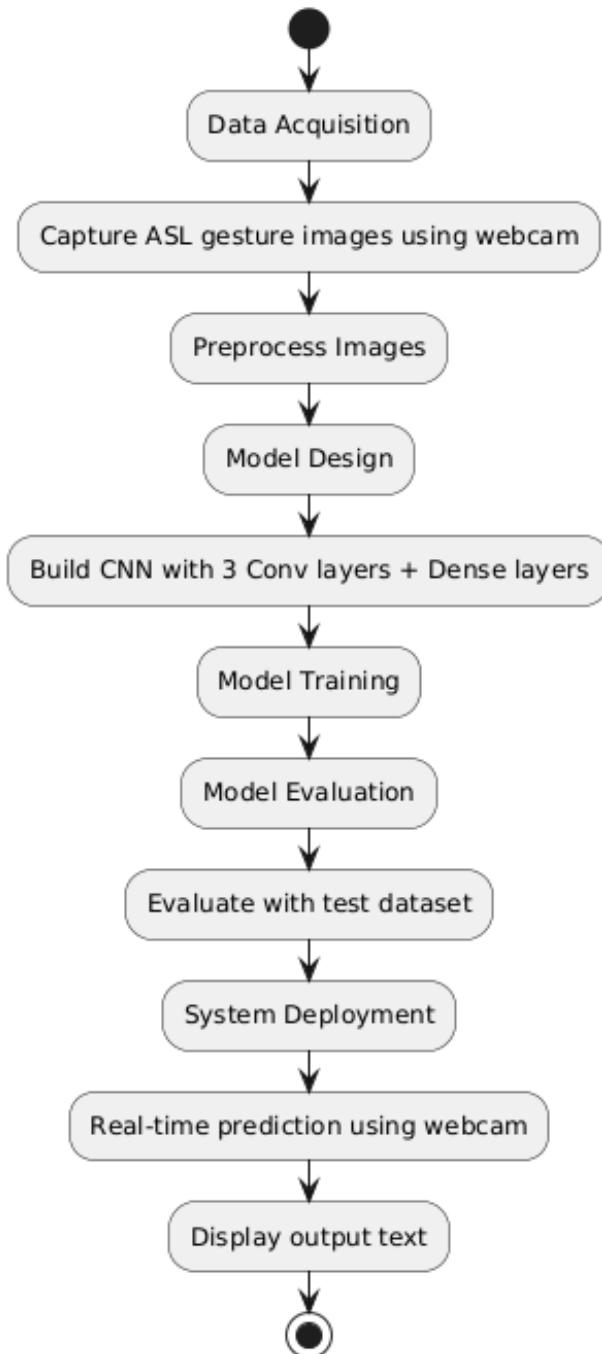
Figure 1: Block diagram

CONCLUSION

The Sign Language Translator developed in this project successfully demonstrates the use of deep learning and computer vision in bridging communication gaps for the deaf and hard-of-hearing community. By recognizing and translating static ASL gestures into text with high accuracy in real time, the system provides a practical and accessible solution for inclusive interaction. The project highlights the potential of machine learning in accessibility-focused applications and establishes a strong foundation for future enhancements such as dynamic sign recognition, multilingual support, and speech integration. This work contributes meaningfully to ongoing efforts in promoting equality, accessibility, and technological empowerment in assistive communication systems.

REFERENCES

- Hope Orovwode, Ibukun Deborah Oduntan, John Abubakar, Development of a Sign Language Recognition System Using Machine Learning, IEEE Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD), 2023. DOI: 10.1109/ICABCD59051.2023.10220456

- Maheshwari Chitampalli, Dnyaneshwari Takalkar, Gaytri Pillai, Pradnya Gaykar, Sanya Khubchandani, Real Time Sign Language Detection, International Research Journal of Modernization in Engineering, Technology and Science (IRJMETS), Volume 05, Issue 04, April 2023. DOI: 10.56726/IRJMETS36648

- TensorFlow. (n.d.). TensorFlow: An end-to-end open-source machine learning plat- form. Retrieved from https://www.tensorflow.org

- OpenCV. (n.d.). Open Source Computer Vision Library. Retrieved from https://opencv.org

- Keras. (n.d.). Deep Learning for humans. Retrieved from https://keras.io