**SignifyAI**

**PROJECT SYNOPSIS**

Of Major Project

**BACHELOR OF TECHNOLOGY**

#### Artificial Intelligence & Machine Learning

|  |  |  |
| --- | --- | --- |
| **Sr No** | **Name** | **PRN Number** |
| 1. | Mr. Digvijay Suryawanshi | 21ST114501041 |
| 2. | Mr. Saahir Syed | 21ST114501042 |
| 3. | Mr. Nikhil Vaishnav | 21ST114501043 |

SUBMITTED BY



SEPTEMBER 2024

# 

# SANJAY GHODAWAT UNIVERSITY, ATIGRE, KOLHAPUR

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Content** | **Page NO.** |
| 1 | Introduction | 2 |
| 2 | Objectives and Scope | 2-3 |
| 3 | Methodology | 3-5 |
| 4 | Resources and Limitations | 6-7 |
| 5 | Applications and Future Scope | 7-8 |
| 6 | Conclusion | 9 |
| 7 | Timeline | 10 |
| 8 | References | 11 |

**SignifyAI**

1. **Introduction**

Problem Statement:

Communication is a fundamental aspect of human interaction and for individuals with hearing or speech impairments, sign language is a primary means of expressing themselves. However, there exists a communication gap between the hearing and non-hearing communities, as many people do not understand or are unfamiliar with sign language. This communication barrier can lead to social isolation and difficulty in accessing essential services.

To address this challenge, the goal of this project is to develop an efficient Sign Language Detection System using OpenCV and Deep Learning techniques that can recognize hand gestures from videos and images and translate them into corresponding textual representation in real time.

1. **Objectives and Scope**

Objectives:

* Real-Time Detection and Tracking: Develop a system using OpenCV to detect hand movements and positions in real-time video or images.
* **Gesture Recognition using Deep Learning:** Train a deep learning model to accurately classify sign language gestures representing letters or words.
* Gesture to Text Conversion: Convert recognized gestures into text, enabling clear communication for those unfamiliar with sign language.
* User-Friendly Interface: Create a simple interface that supports live video input via webcam and allows users to upload pre-recorded images or videos.

Scope:

* Data Collection and Preparation: A dataset of sign language gestures will be collected, pre-processed, and annotated for training.
* Model Training and Optimization: Different neural network architectures will be tested and optimized for high accuracy.
* Integration with Real-Time Applications: The system will be integrated into live video feeds for real-time gesture recognition and text conversion.
* Extensibility: Future updates will support additional sign languages, vocabulary, and improvements in recognition algorithms.
* Application in Various Domains: The system can be applied in education, customer service, and healthcare to enhance communication with hearing-impaired individuals.

1. **Methodology**

The Sign Language Detection System consists of multiple stages, beginning with the input of hand gesture data (video or images), processing the data, recognizing gestures using deep learning, and converting them into text.

1. Input Capture

* The system captures real-time video or static images of hand gestures. The input can come from various sources like live webcam feeds or pre-uploaded video/image files.

2. Preprocessing

* The captured images or frames from the video undergo preprocessing to isolate the hand gesture from the background using techniques such as thresholding, resizing, color space conversion, and noise reduction. This is essential for accurate gesture detection.

3. Hand Gesture Detection

* The hand region is detected from the preprocessed input using computer vision techniques, such as contour detection and edge detection, implemented using OpenCV. The detected hand gesture is then segmented for further recognition.

4. Gesture Recognition

* After detection, the system applies a deep learning model (e.g., CNN) to classify the gesture into corresponding letters or words in sign language. The model is trained on a labeled dataset of sign language gestures.

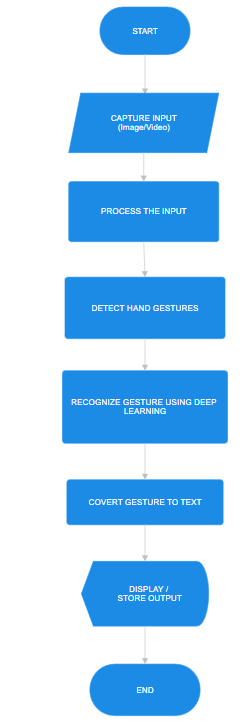
5. Text Conversion

* The recognized gestures are converted into text, which can be displayed in real-time. This allows the user to read the translated gesture in the form of a word, phrase, or sentence.

6. Output

* The system generates text as output from the detected sign language gestures. The output is displayed in a user-friendly interface or stored for further use.

Flowchart:



1. **Resources and Limitations**

Hardware Resources

1. High-Performance Computer:

* Minimum of Intel Core i5/Ryzen 5 processor and 16 GB RAM.
* NVIDIA GPU (e.g., GTX 1650 or higher) for deep learning model training and inference, especially for real-time gesture recognition.

2. Webcam:

* A high-quality webcam (720p or 1080p) to capture hand gestures for real-time detection.

3. External Storage (optional):

* For storing large datasets and models, SSD or cloud-based storage can be used.

1. Power Supply:

* Reliable power, especially for extended model training sessions when using GPU resources.

Software Resources

1. Operating System:

* Windows, macOS, or Ubuntu with Python 3.x compatibility.

1. Deep Learning Frameworks:

* TensorFlow or Keras for building and training gesture recognition models.

1. Computer Vision Library:

* OpenCV for video processing, image preprocessing, and gesture detection.

1. GPU Acceleration Libraries:

* CUDA Toolkit (for NVIDIA GPUs) to accelerate the training and real-time inference of deep learning models.

1. Development Tools:

* PyCharm, VS Code, or Jupyter Notebook for code development.

Limitations

1. Limited Vocabulary: The system may only support a specific sign language (e.g., American Sign Language) and recognize basic gestures like alphabets or words. Expanding it to complex gestures or multiple languages requires more data and training.
2. Accuracy Issues: Performance may degrade in poor lighting, noisy backgrounds, or with varying skin tones. The system might struggle in real-world environments.
3. Hardware Dependency: Real-time gesture recognition relies heavily on powerful GPUs. Users without such hardware may face slow performance.
4. Lack of Contextual Understanding: The system doesn't account for facial expressions, body movements, or the context in which gestures are made, which are essential in real-world sign language.
5. **Application and Future scope**

Applications

1. Communication Aid for Hearing-Impaired:

* The system can bridge the communication gap between hearing-impaired individuals and non-sign language users by translating sign language into text, making communication more accessible in public spaces, schools, and workplaces.

1. Education and Learning Tool:

* This system can be used as an educational tool for teaching sign language to beginners or students, providing immediate feedback on hand gestures and their meanings.

1. Real-Time Interpretation:

* It can be employed in public service areas like hospitals, government offices, and customer service centers to provide real-time sign language interpretation, improving accessibility for the hearing-impaired.

1. Telemedicine:

* In healthcare, the system can assist hearing-impaired patients in communicating with doctors during telemedicine consultations, improving accessibility and healthcare outcomes.

Future Scope

1. Gesture Contextualization:

* Future versions could incorporate facial expressions, body movements, and contextual understanding, which are essential for fully interpreting sign language. This would make the system more accurate and reliable.

1. Voice Output Integration:

* Adding a text-to-speech feature can enhance communication by allowing the translated text to be spoken aloud, making it more helpful for non-sign language users.

1. **Conclusion**

The Sign Language Detection System represents a significant innovation in bridging communication gaps between hearing-impaired individuals and the broader community. By leveraging advanced technologies such as OpenCV for computer vision and deep learning frameworks like TensorFlow, the system effectively translates hand gestures into text in real-time. This capability not only enhances communication accessibility but also promotes inclusivity in various social, educational, and professional settings.

1. **Timeline**

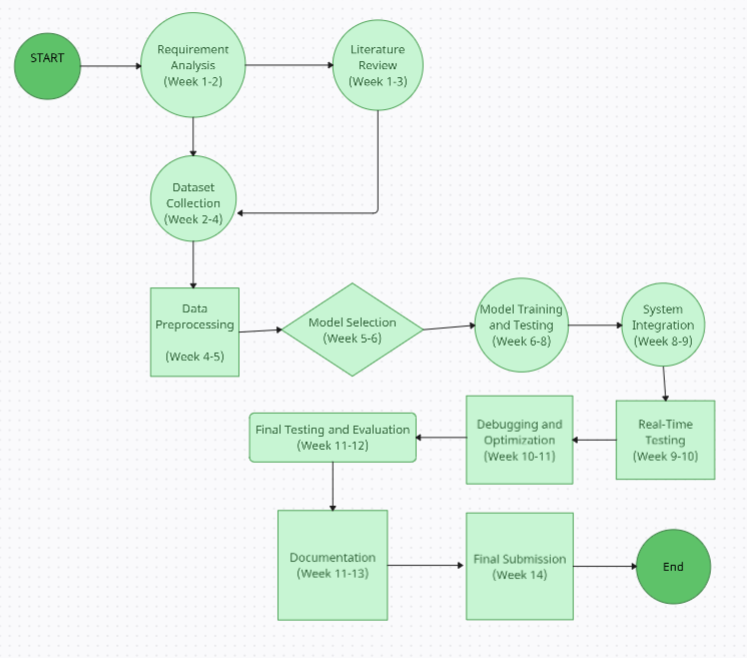


Fig. PERT Chart

1. **References**

* Koller, O., Ney, H., & Bowden, R. (2015). "Deep learning of mouth shapes for sign language."
* IEEE International Conference on Computer Vision (ICCV).

This paper demonstrates how deep learning can be applied to sign language recognition, particularly focusing on the use of Convolutional Neural Networks (CNNs).

* <https://www.researchgate.net/publication/262187093_Sign_language_recognition_State_of_the_art>
* <https://www.researchgate.net/publication/373385057_Development_of_a_Sign_Language_Recognition_System_Using_Machine_Learning>
* OpenCV Documentation

OpenCV provides extensive documentation and tutorials on working with image and video processing, which are essential for gesture detection.

Link: <https://docs.opencv.org>

* Real-time Hand Gesture Recognition Using OpenCV and Python (Blog Post)

This guide walks through hand gesture detection using OpenCV and machine learning, a key step in sign language detection.

Link: <https://learnopencv.com/real-time-hand-gesture-recognition/>

**Mrs. Varsha Kasote Dr. Mahesh P. Gaikwad**

**Guide Head of Department**