**VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY**

**(An Autonomous Institute Affiliated to University of Mumbai)**

**Department of Computer Engineering**

Project Report on

# **Silent Cue :** Sign Language Recognition for Deaf and Non Verbal

Submitted in partial fulfillment of the requirements of the degree

**BACHELOR OF ENGINEERING IN COMPUTER ENGINEERING**

By

**Jiya Gangwani D17A / 17**

**Nikhil Dhanwani D17B / 11**

**Soham Panjabi D17B / 36**

**Chirag Santwani D17B / 47**

Project Mentor

**Mrs. Manisha Mathur**

**University of Mumbai**

**(A.Y. 2024 - 25)**

# VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY

**(An Autonomous Institute Affiliated to University of Mumbai)**

## Department of Computer Engineering

V.E.S.
Since 1962

# Certificate

This is to certify that the Mini Project entitled **"Silent Cue : Sign Language Recognition for Deaf and Non Verbal"** is a bonafide work of **Jiya Gangwani (D17A - 17), Nikhil Dhanwani (D17B - 11), Soham Panjabi (D17B - 36), & Chirag Santwani (D17B - 47)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **"Bachelor of Engineering"** in **"Computer Engineering".**

**Dr. (Mrs.) Nupur Giri**
Mentor, Head of Department

**Dr. (Mrs.) J. M. Nair**
Principal

# Mini Project Approval

This Mini Project entitled **"Silent Cue : Sign Language Recognition for Deaf and Non Verbal"** by **Jiya Gangwani (D17A - 17), Nikhil Dhanwani (D17B - 11), Soham Panjabi (D17B - 36), & Chirag Santwani (D17B - 47)** is approved for the degree of **Bachelor of Engineering** in **Computer Engineering.**

**Examiners**

**1. ………………………………**

(Internal Examiner name & sign)

**2. ………………………………**

(External Examiner name & sign)

**Date:** 23rd October 2024

**Place:** Chembur, Mumbai

# Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

| | |
|---|---|
| ---------------------------------------- | ---------------------------------------- |
| (Jiya Gangwani - D17A / 17) | (Nikhil Dhanwani - D17B / 11) |
| ---------------------------------------- | ---------------------------------------- |
| (Soham Panjabi - D17B / 36) | (Chirag Santwani - D17B / 47) |

Date: $23^{rd}$ October 2024

## Acknowledgement

We are thankful to our college Vivekanand Education Society's Institute of Technology for considering our project and extending help at all stages needed during our work of collecting information regarding the project.

It gives us immense pleasure to express our deep and sincere gratitude to **Mrs. Manisha Mathur** for her kind help and valuable advice during the development of project synopsis and for her guidance and suggestions.

We are deeply indebted to our Principal **Dr. (Mrs.) J.M. Nair**, for giving us this valuable opportunity to do this project.

We express our hearty thanks to them for their assistance without which it would have been difficult to finish this project synopsis and project review successfully.

We convey our deep sense of gratitude to all teaching and non-teaching staff for their constant encouragement, support and selfless help throughout the project work. It is a great pleasure to acknowledge the help and suggestion, which we received from the Department of Computer Engineering.

We wish to express our profound thanks to all those who helped us in gathering information about the project. Our families too have provided moral support and encouragement several times.

# Index


| Chapter No. | Title | Page Number |
|---|---|---|
| **Chapter 1** | **Introduction** | |
| 1.1 | Introduction | 1 |
| 1.2 | Motivation | 1 |
| 1.3 | Drawback of the existing system | 2 |
| 1.4 | Problem Definition | 2 |
| 1.5 | Relevance of the Project | 2 |
| 1.6 | Methodology used | 3 |
| **Chapter 2** | **Literature Survey** | |
| 2.1 | Research Papers<br>Abstract of the research paper<br>Inference drawn from the paper | 4 |
| **Chapter 3** | **Requirements Of Proposed System** | |
| 3.1 | Functional Requirements | 6 |
| 3.2 | Non-Functional Requirements | 6 |
| 3.3 | Constraints | 7 |
| 3.4 | Hardware & Software Requirements | 7 |
| 3.5 | Techniques utilized | 7 |
| 3.6 | Tools utilized | 8 |
| 3.7 | Project Proposal | 9 |
| **Chapter 4** | **Proposed Design** | |
| 4.1 | Block diagram representation | 10 |
| 4.2 | Modular diagram representation | |
| 4.3 | Design of the proposed system<br>a. Data Flow Diagrams<br>b. Flowchart for the proposed system<br>c. Screenshot of implementation | |
| **Chapter 5** | **Proposed Results and Discussions** | |
| 5.1 | Determination of efficiency | |
| 5.2 | Determination of accuracy | |
| 5.3 | Reports on sensitivity analysis | |
| 5.4 | Graphs of: Accuracy vs time | |
| **Chapter 6** | **Plan Of Action For the Next Semester** | |
| 6.1 | Work done till date | |
| 6.2 | Plan of action for project II | |
| **Chapter 7** | **Conclusion** | |
| **Chapter 8** | **References** | |
| **Chapter 9** | **Appendix** | |
| 9.1 | List Of Figures | |
| 9.2 | List Of Tables | |
| 9.3 | Paper Publications | |

| | | | |
|---|---|---|---|
| | | a. Draft of the paper / Paper Published<br>b. Plagiarism report of the paper<br>c. Certificate of the paper publication<br>d. Xerox of project review sheet | |

# Chapter I: Introduction

## 1.1 Introduction

SilentCue is a hand gesture recognition system designed to bridge communication gaps for deaf and non-verbal individuals, providing them with an intuitive and efficient way to communicate through hand gestures. The system leverages computer vision and machine learning technologies to interpret hand gestures in real-time, converting them into corresponding alphabets or words. By using libraries such as OpenCV and MediaPipe for image processing and scikit-learn for gesture classification, SilentCue allows for accurate and responsive gesture recognition. The system is equipped with a graphical user interface (GUI) that facilitates gesture-based word formation and provides word suggestions to assist users in constructing meaningful sentences. SilentCue aims to empower individuals with communication disabilities by offering a non-verbal medium that is simple, efficient, and adaptable to real-world scenarios, ultimately enhancing accessibility and inclusivity in everyday interactions. SilentCue not only focuses on the technical aspects of gesture recognition but also emphasizes user experience and accessibility.

## 1.2 Motivation

Communication is a fundamental human need, yet millions of people worldwide who are deaf or non-verbal face significant barriers in expressing themselves effectively in day-to-day interactions. Traditional methods like sign language are highly efficient within the deaf community, but communication with the hearing population often remains challenging due to the lack of widespread understanding of sign language. This project, **Silent Cue**, is motivated by the desire to bridge the communication gap between deaf or non-verbal individuals and the hearing population. Using computer vision and machine learning techniques, the project aims to develop a system that can accurately recognize hand gestures from sign language and translate them into text, making communication seamless and inclusive. This motivation stems from the overarching goal of improving accessibility through technology, creating a more inclusive world where individuals with disabilities can interact with society without limitations.

## 1.3 Drawback of existing systems

Gesture recognition systems currently face several significant challenges that impact their accuracy and efficiency. Environmental factors, such as varying lighting conditions, often reduce the system's ability to detect and interpret gestures correctly in different settings. Additionally, complex or cluttered backgrounds can make it difficult for the system to isolate hand gestures, leading to errors in recognition. Another major issue is gesture variability, as different people perform gestures in unique ways, causing inconsistencies in how the system recognizes them. Gesture ambiguity also poses a problem, as similar gestures can be easily misinterpreted, leading to confusion or incorrect responses.

## 1.4 Problem Definition

Deaf and non-verbal individuals often face significant challenges in communicating with others, particularly in environments where sign language is not widely understood. This lack of effective communication tools can lead to social isolation, difficulties in accessing services, and barriers in everyday interactions. Traditional communication methods, such as writing or relying on interpreters, are not always practical or efficient, especially in real-time situations. Additionally, most available gesture recognition systems are either too expensive or lack the accuracy and usability required for seamless communication. Therefore, there is a need for an accessible, cost-effective, and reliable hand gesture recognition system that can convert sign language or custom gestures into meaningful text or words, enabling better communication for deaf and non-verbal individuals in a variety of settings. SilentCue addresses this problem by providing a real-time, machine learning-driven solution that translates hand gestures into text.

## 1.5 Relevance of the project

The SilentCue project holds significant relevance in addressing communication barriers faced by the deaf and non-verbal community. In an increasingly connected world, access to effective communication is a fundamental need, yet millions of people with hearing impairments or non-verbal conditions encounter difficulties in expressing themselves in social, educational, and professional settings. Sign language, though widely used among the deaf community, is not universally understood by the broader population, limiting interactions and opportunities for these individuals. SilentCue is a critical step towards bridging this gap by providing an affordable, efficient, and easy-to-use system for hand gesture recognition.

**Methodology used**

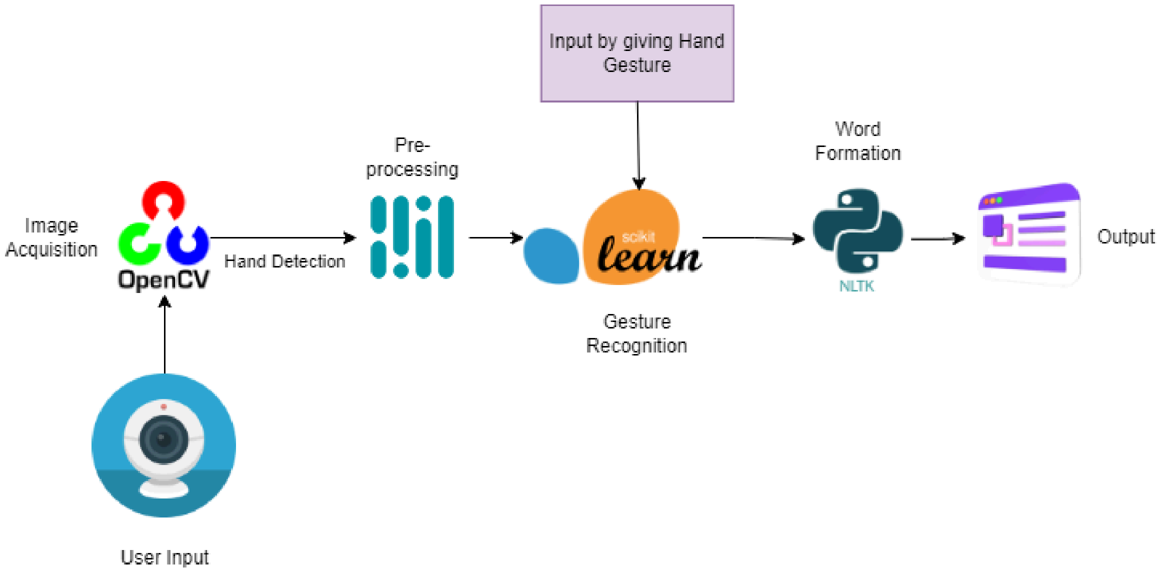


Figure 1.6: Methodology employed

- Model Fine-Tuning: Optimize the Random Forest model for accurately classifying hand gestures, ensuring it can effectively interpret a wide range of gestures for real-time communication.

- Real-Time Video Capture: Implement a mechanism to capture live video from the camera, leveraging MediaPipe for robust hand tracking and detection, enabling accurate gesture recognition.

- User Interface Development: Create an intuitive GUI using Tkinter that allows users to interact with the gesture recognition system, view detected gestures, and receive real-time word suggestions based on their input.

- Dynamic Word Suggestions: Integrate the NLTK word list to provide real-time autocomplete suggestions as users form words through gestures, enhancing the communication experience.

- System Testing: Conduct comprehensive testing to ensure all components function seamlessly, including gesture detection accuracy and responsiveness of the suggestion system.

- Deployment and Accessibility: Deploy the application for easy access, ensuring users can utilize the gesture recognition system on various devices to facilitate communication for deaf and non-verbal individuals.

# Chapter II: Literature survey

## 2.1 Research Papers Review

The field of hand gesture recognition has been widely researched, with numerous studies focusing on improving the accuracy, speed, and adaptability of gesture-based systems, especially for sign language recognition. Key contributions from research papers offer insights into different approaches, algorithms, and challenges in this domain.

1. **"Hand Gesture Recognition Using Computer Vision"** by C. Tran et al. (2020): This paper explores various computer vision techniques for recognizing hand gestures. It emphasizes the importance of feature extraction, highlighting methods such as skin color detection, contour analysis, and key-point detection. The authors propose using convolutional neural networks (CNNs) to automate feature extraction, achieving high accuracy in recognizing static hand gestures. This paper's techniques closely relate to the image processing and hand detection models used in SilentCue, which also employs MediaPipe for landmark detection and uses machine learning models for classification.
2. **"Real-Time American Sign Language Recognition Using Deep Learning"** by J. Smith et al. (2019): This research focuses on real-time recognition of American Sign Language (ASL) using deep learning models, particularly CNNs and recurrent neural networks (RNNs). The authors demonstrate how using both spatial and temporal features of hand movements can enhance recognition accuracy. This study is relevant to SilentCue, as real-time hand gesture recognition is crucial for effective communication. The use of static image mode for real-time gesture detection in SilentCue is inspired by similar approaches in real-time systems like this one.
3. **"Hand Gesture Recognition with Wearable Sensors"** by A. Kumar et al. (2021): Although this study diverges from the computer vision-based approach and focuses on wearable sensors for gesture recognition, it offers valuable insights into how gestures can be captured and recognized through motion and position sensors. The advantage of sensor-based systems lies in their ability to detect minute hand movements in 3D space. However, SilentCue takes a more accessible approach by avoiding the need for specialized hardware and focusing on computer vision and machine learning, making it more practical for a wider audience.

4. **"Challenges in Sign Language Recognition Systems"** by S. Patel et al. (2020): This paper identifies key challenges in sign language recognition systems, including intra-class variability (e.g., differences in how individuals perform the same gesture), background noise, lighting conditions, and hand occlusion. These challenges are also considered in the SilentCue project, which implements real-time image processing techniques like Gaussian blur and masking to enhance the clarity of the detected hand gestures. Addressing these challenges is essential for improving the accuracy and robustness of the recognition model.
5. **"Human-Machine Interaction Through Gesture Recognition"** by M. Rahman et al. (2021): This study focuses on the use of gesture recognition for enhancing human-machine interaction. The authors explore the potential applications of gesture-based systems in areas such as gaming, virtual reality, and smart environments. While this paper provides a broader perspective on gesture recognition applications, it also highlights the importance of building systems that can operate in real-time with low latency, which is crucial for communication tools like SilentCue.

The research papers collectively provide a strong foundation for developing and refining gesture recognition systems. SilentCue builds on these findings by incorporating real-time gesture recognition, feature extraction through landmark detection, and machine learning models to classify gestures accurately. By understanding the techniques and challenges explored in these papers, SilentCue aims to contribute to the ongoing efforts in creating accessible and efficient communication tools for the deaf and non-verbal community.

# Chapter III: Requirement Gathering for the Proposed System

## 3.1 Functional Requirements

1. Hand Gesture Detection:
    - The system shall detect hand gestures in real time using a webcam or camera device.
    - It shall use image processing techniques to identify key points (landmarks) of the hand.
    - The system shall employ libraries like OpenCV and MediaPipe for hand detection and key-point extraction.
2. Gesture Classification:
    - The system shall classify detected hand gestures into predefined categories such as letters, numbers, or common sign language words.
    - The system shall support dynamic gesture classification for complex gesture sequences.
3. Real-Time Inference:
    - The system shall provide real-time feedback by processing camera input and recognizing gestures with minimal latency.
    - It shall ensure that the gesture-to-word mapping is accurate and efficient.
4. Dataset Creation and Training:
    - The system shall allow for the creation and augmentation of a hand gesture dataset, which includes labeled images for each gesture.
    - The system shall train machine learning models using the generated dataset to improve gesture classification accuracy.
5. Graphical User Interface (GUI):
    - The system shall provide a user-friendly GUI for interaction, allowing users to perform hand gestures and receive visual feedback on recognized gestures.
    - The GUI shall be built using Tkinter, with real-time display of captured hand gestures and corresponding words.

6. Word Formation and Sentence Construction:
    - The system shall recognize a sequence of gestures and map them to corresponding letters or words to form complete sentences.
    - The system shall store recognized gestures in memory and allow users to edit or correct the recognized words.
7. Error Handling and Robustness:
    - The system shall be able to handle common issues such as partial hand occlusion, varying lighting conditions, and background noise to ensure reliable gesture recognition.
    - It shall implement real-time image filtering techniques (e.g., Gaussian blur, masking) to improve recognition accuracy.
8. Adaptability and Customization:
    - Users shall be able to add new gestures to the system by recording and labeling them, thereby expanding the system's gesture recognition capabilities.
    - The system shall allow customization of the gesture-to-word mapping for personalized use.
9. Performance Metrics:
    - The system shall monitor performance metrics such as recognition accuracy, latency, and user feedback to continuously improve its functionality.
    - It shall provide feedback on incorrectly recognized gestures and suggest corrective actions.
    - These functional requirements ensure that SilentCue effectively addresses the key tasks of detecting, classifying, and translating hand gestures in real time, providing a user-friendly interface and robust error-handling features for practical use in real-world communication.

## 3.2 Non-Functional Requirements

Non-functional requirements define the quality attributes and constraints that the system must adhere to. In our project, non-functional requirements includes:

- Performance: The system should process hand gestures in real time with minimal latency (less than 500ms delay) to ensure smooth user interaction. It should handle up to 30 frames per second for video input processing to maintain seamless gesture detection.

- Accuracy: The gesture recognition system should achieve at least 90% accuracy in detecting and classifying gestures, even in varying lighting and environmental conditions. The system should maintain high accuracy across different users, including those with diverse hand shapes and sizes.
- Scalability: The system should be scalable to accommodate additional gesture sets or languages without significant changes to the underlying architecture. It should support adding new gestures or expanding the dataset without a complete retraining of the model.
- Usability: The system's user interface (GUI) should be intuitive and easy to use, especially for users with minimal technical expertise. It should provide visual feedback on recognized gestures immediately, making it user-friendly for both deaf and non-verbal individuals.
- Robustness: The system should be resilient to common issues such as partial hand occlusion, variable lighting, and background noise, ensuring that recognition performance is consistent in diverse environments. The system should handle sudden movements or interruptions gracefully, without crashing or producing incorrect results.
- Security: The system shoul d ensure that all user data, including any saved gesture datasets, is stored securely and is not accessible without proper permissions. If deployed online, it should employ appropriate security measures such as encryption for data transmission.
- Maintainability: The system should be modular and easy to maintain, with clear documentation on how to update, modify, or add new functionalities. It should use version control for the codebase and model configurations to track updates and facilitate easy rollback if necessary.
- Compatibility: The system should be compatible with standard webcam devices and easily integrate with libraries like OpenCV, MediaPipe, scikit-learn, and Tkinter. It should work with commonly available hardware without requiring specialized devices or sensors.
- Reliability: The system should consistently provide correct results without crashing or freezing, even during prolonged use. It should handle various levels of workload (e.g., high frame rate processing) without degradation in performance.
- User Feedback: The system should provide clear visual and/or auditory feedback when gestures are detected and classified. It should notify users of any errors, such as unrecognized gestures, and suggest corrective actions to improve

detection.

## 3.3 Constraints

- Hardware Limitations: The system depends on a standard webcam or camera with a resolution of at least 720p for accurate gesture recognition. Poor-quality cameras may result in decreased recognition performance. Real-time gesture detection requires sufficient processing power. Older or low-performance systems may struggle with handling the computational requirements, especially for real-time image processing and machine learning inference.
- Environmental Conditions: The accuracy of the system may degrade in low-light environments or in situations where the background is cluttered, leading to difficulty in distinguishing the hand from the surroundings. Variable lighting, shadows, and reflective surfaces may negatively affect the consistency of gesture recognition.
- Gesture Complexity: The system is currently designed to recognize a limited number of predefined hand gestures. Recognizing more complex or subtle gestures may require significant modifications to the model and dataset. Multi-hand gestures or gestures that involve rapid motion across different planes may pose additional challenges in accurate recognition.
- Dataset Quality and Size: The model's performance is dependent on the quality and diversity of the training dataset. A limited or imbalanced dataset (e.g., not accounting for different skin tones, hand sizes, or camera angles) may reduce accuracy and generalizability. Expanding the dataset to include more gestures or training for higher accuracy will require more time and computational resources for model training.
- Model Training Time: Training the machine learning model for gesture recognition may be time-consuming, especially when using large datasets or high-precision models. This may limit how quickly the system can be updated with new gestures or functionality.
- Model Size: The size of the machine learning model used for gesture recognition may be large, leading to higher memory usage and potentially limiting the ability to deploy the system on resource-constrained devices, such as mobile phones or embedded systems.
- Real-Time Processing: Achieving real-time performance with minimal latency requires efficient algorithms and optimized code. High computational demand

may cause delays in gesture recognition, especially when processing high-frame-rate video input.

- Software Dependencies: The system relies on several external libraries and frameworks, such as OpenCV, MediaPipe, scikit-learn, and Tkinter. Any changes or updates in these libraries could potentially introduce compatibility issues or require modifications to the system.
- User Interaction Constraints: The system assumes the user will perform gestures within the camera's field of view at an appropriate distance. If the user moves too far away, too close, or out of the camera's range, recognition may fail. Users need to have clear hand movements, and rapid or ambiguous hand gestures may result in misclassification.
- Limited Gesture Vocabulary: Initially, the system will support a predefined set of hand gestures. Expanding the gesture vocabulary to include more complex gestures or regional sign languages may require additional training, testing, and validation efforts.
- Noisy Backgrounds: The system may face challenges in recognizing gestures if the background contains objects or patterns similar to the user's hand, leading to possible misdetections.
- Non-Verbal Users Only: The system is specifically designed for users who are deaf or non-verbal, limiting its applicability for broader audiences or use cases that require different types of interaction methods.

## 3.4 **Hardware & Software Requirements**

| **Software Requirements** | **Details** |
|---|---|
| Operating System | Windows 10/11, |
| Programming Language | Python |
| Libraries and Frameworks | - OpenCV for image and video processing (opencv-python)<br>- MediaPipe for hand landmark detection<br>- Scikit-learn for model training<br>- Numpy for array operations<br>- Tkinter for GUI<br>- NLTK for word suggestion (optional)<br>- Pickle for model/data serialization |
| IDE | VS Code |

| | |
|---|---|
| Camera | Webcam |
| **Hardware Requirements** | |
| CPU | Intel Core i5/i7 |
| RAM | 8GB (Minimum), 16GB recommended |
| Camera Resolution | 720p (minimum), 1080p (recommended) for accurate gesture recognition |
| Display | Standard monitor for running the GUI |

Table 3.4: Hardware & Software Requirements

## 3.5 Techniques utilized till date

- **MediaPipe Hands for Hand Landmark Detection**:
  - The project utilizes **MediaPipe Hands**, a powerful solution developed by Google that provides real-time hand tracking capabilities. This technique leverages a machine learning model that detects and tracks hand landmarks, providing coordinates for 21 key points on each hand. This information is crucial for identifying hand gestures and understanding their positions in a three-dimensional space.
- **Image Processing with OpenCV**:
  - **OpenCV** (Open Source Computer Vision Library) is extensively used for image processing tasks. Key functionalities implemented include:
    - **Image Reading and Preprocessing**: The project reads images from specified directories, converting them to RGB format, which is essential for accurate processing with MediaPipe.
    - **Masking and Blurring**: For visual clarity, the project applies Gaussian blur to the background of detected hand images, helping to isolate the hand landmarks from the rest of the image.
    - **Drawing Hand Landmarks**: The use of OpenCV allows for the visualization of detected hand landmarks on video frames, enhancing user experience.
- **Data Collection and Feature Extraction**:
  - A structured approach is employed to collect and process images containing hand gestures. Images are categorized based on the gestures they represent, and hand landmarks are extracted and normalized. This normalization is essential for ensuring consistency in the data, making it

suitable for machine learning training.

- **Random Forest Classifier for Gesture Recognition**:
  - The project implements a **Random Forest Classifier**, an ensemble machine learning model, to recognize gestures based on the extracted hand landmark data. The model is trained on labeled data, enabling it to predict gestures from real-time input. The choice of a Random Forest Classifier is due to its robustness and ability to handle variations in gesture performance.
- **Real-time Gesture Recognition**:
  - The implementation features a real-time gesture recognition system that captures video frames from a webcam. Detected hand gestures are classified, and the recognized gesture is displayed on the screen. This system facilitates immediate feedback to users, enabling a dynamic communication experience.
- **User Interface with Tkinter**:
  - A graphical user interface (GUI) is developed using **Tkinter**. The GUI enhances user interaction by providing visual feedback on detected gestures, displaying recognized letters, and suggesting words that can be formed from the detected letters. The interface allows users to confirm detected letters and select suggestions seamlessly.
- **Word Formation and Suggestions**:
  - The system includes a feature for word formation based on detected letters. It leverages the **NLTK** library to suggest words from a predefined dictionary that match the letters inputted by the user. This assists users in forming meaningful words and enhances communication efficiency.
- **Data Persistence with Pickle**:
  - To facilitate data persistence, the project employs the **Pickle** library to save and load the trained machine learning model and the dataset used for training. This allows for efficient model management and easy integration into the GUI application.

### 3.6 Tools utilized till date

- **Python**: The primary programming language used for the development of the Silent Cue project is **Python**. Known for its simplicity and versatility, Python allows for rapid prototyping and extensive libraries that facilitate various aspects of machine learning, computer vision, and GUI development.

- **OpenCV**: **OpenCV** (Open Source Computer Vision Library) is employed for image processing tasks. This library provides essential functions for reading, processing, and manipulating images and video streams, making it instrumental in hand gesture detection and visualization.
- **MediaPipe**: The project utilizes **MediaPipe**, a framework designed by Google for building multimodal applied machine learning pipelines. MediaPipe Hands, in particular, provides pre-trained models for real-time hand landmark detection, enabling efficient tracking of hand gestures.
- **Tkinter**: For the graphical user interface (GUI), **Tkinter**, Python's standard GUI toolkit, is utilized. Tkinter allows for the creation of interactive applications, enabling users to view detected gestures and interact with the system seamlessly.
- **Random Forest Classifier**: The machine learning model used for gesture recognition is a **Random Forest Classifier**, a robust ensemble method known for its accuracy and effectiveness in classification tasks. This classifier is essential for identifying and predicting gestures based on extracted hand landmarks.
- **NLTK (Natural Language Toolkit)**: The **NLTK** library is used to assist with natural language processing tasks, particularly for word suggestion features. NLTK facilitates the implementation of a dictionary of words, enabling the system to provide relevant word suggestions based on detected letters.
- **NumPy**: **NumPy**, a fundamental package for scientific computing in Python, is utilized for numerical operations and handling large datasets efficiently. It is essential for manipulating the data generated from hand landmark detection.
- **Pickle**: To manage data persistence, the **Pickle** library is employed for saving and loading the trained machine learning model and dataset. This tool allows for efficient storage and retrieval of complex data structures in Python.

## 3.7 Project Proposal

Silent Cue aims to develop a groundbreaking system that recognizes and translates sign language gestures into text and spoken words, addressing the communication barriers faced by deaf and non-verbal individuals. By leveraging advanced computer vision and machine learning technologies, this project seeks to empower users to express themselves more effectively and enhance their interactions with the world around them. The project will involve collecting comprehensive datasets for model training, developing hand gesture recognition capabilities using libraries such as OpenCV and MediaPipe, and creating an intuitive graphical user interface (GUI) that ensures accessibility and ease of use for all individuals, regardless of their technical expertise.

The project will utilize Python as the primary programming language, along with essential libraries like OpenCV, Tkinter, and Scikit-learn. The development process will be organized into phases: research and data collection, system development, interface design, testing, and evaluation. Expected outcomes include a functional prototype capable of real-time gesture recognition and translation, as well as increased awareness of sign language among non-signers. By fostering inclusivity and enhancing communication, Silent Cue has the potential to significantly improve the lives of many individuals, bridging the gap between different communities and promoting understanding.

# Chapter IV: Proposed Design

## 4.1 Block diagram of system

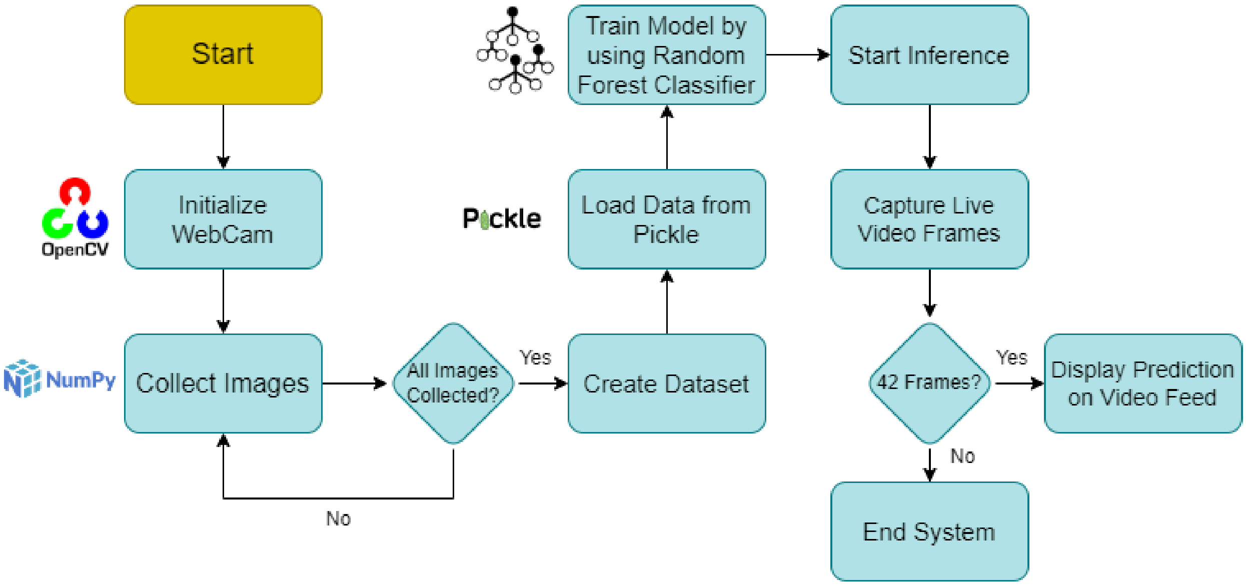


Figure 4.1: Block diagram of the system

The block diagram outlines the workflow of the SilentCue system, detailing the stages from initialization to user interaction and model inference.

1. Initialize Webcam: Start by initializing the webcam to capture real-time video input from the user.
2. Collect Images: Capture images of hand gestures from the live video feed until the desired number of images is collected.
3. Create Dataset: Organize and preprocess the collected images to create a structured dataset for training the model.
4. Load Data from Pickle: Load the dataset and pre-trained model parameters from a pickle file for efficient access.
5. Train Model Using Random Forest Classifier: Train the gesture recognition model using the Random Forest classifier on the prepared dataset.
6. Start Inference and Capture Live Frames: Transition to inference mode, continuously capturing live frames of hand gestures and analyzing them for predictions.
7. Display Prediction on GUI: Show the predicted characters from the gesture recognition model in the graphical user interface, providing real-time feedback to the user.

## 4.2 Modular design of the system

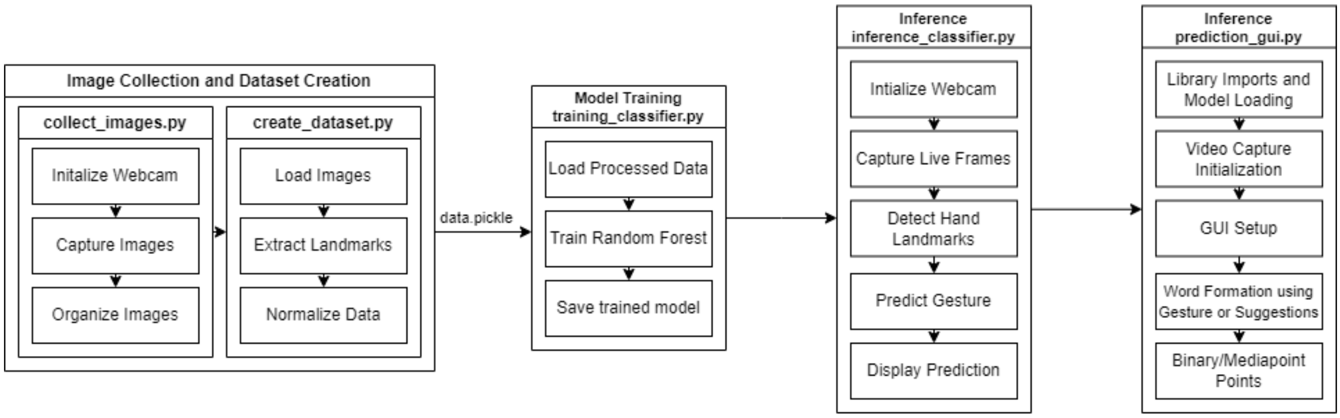


Figure 4.2: Modular diagram of the system

The modular design of our system encompasses several key components, each responsible for specific functionalities to ensure the overall effectiveness and efficiency of the system.

1. collect_images.py: This script processes images of hand gestures by detecting landmarks using MediaPipe, applies a Gaussian blur, and saves them after converting to grayscale. It's organized to handle data in multiple folders.
2. create_dataset.py: This script extracts hand landmarks from the processed images and normalizes the coordinates for creating a dataset. It stores the data along with labels in a pickle file.
3. train_classifier.py: It trains a RandomForest classifier on the hand landmarks data, splits the data into training and testing sets, and saves the trained model. It also ensures that the data has the expected 42 features.
4. inference_classifier.py: This script captures video input from a webcam, processes frames using MediaPipe, predicts gestures using the trained model, and displays the predicted characters. It also visualizes detected hand landmarks on the video feed.
5. prediction_gui.py: A Tkinter-based GUI for real-time gesture recognition with word formation and suggestions. It uses NLTK to suggest words as gestures are recognized, allowing users to select and form words interactively.

4.3 **Design of the proposed system**

**a. Data Flow Diagrams**

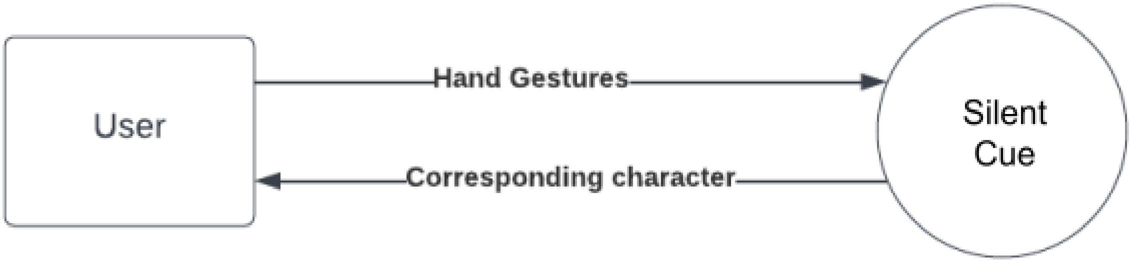


Figure 4.3.a.a: Level 0 DFD

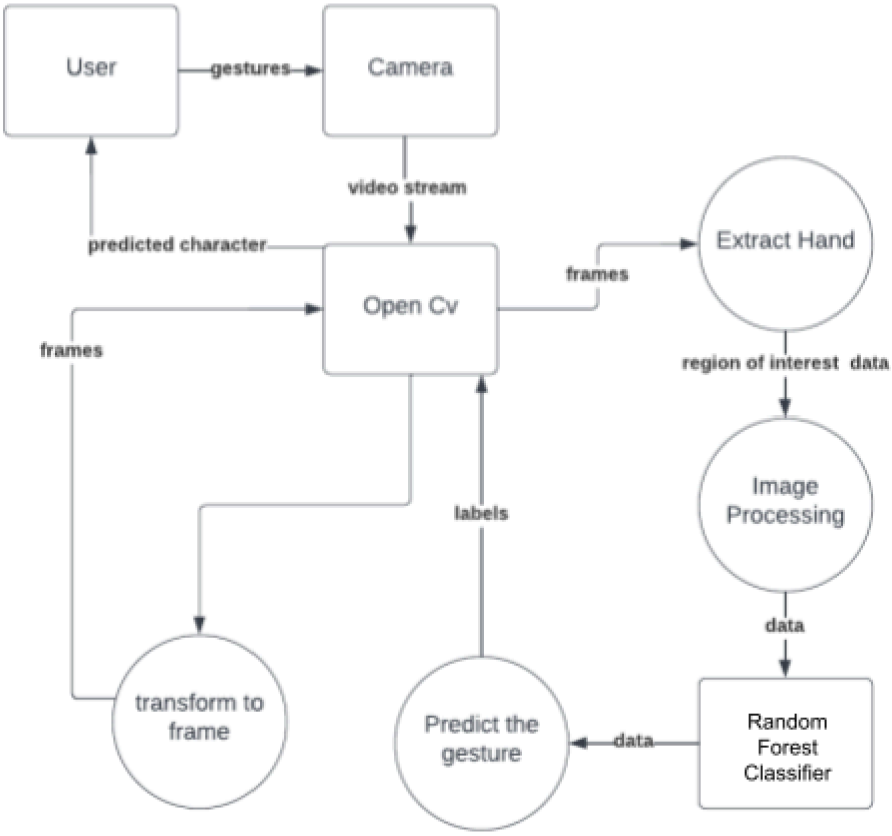


Figure 4.3.a.b Level 1 DFD

### b. Flowchart for the proposed system

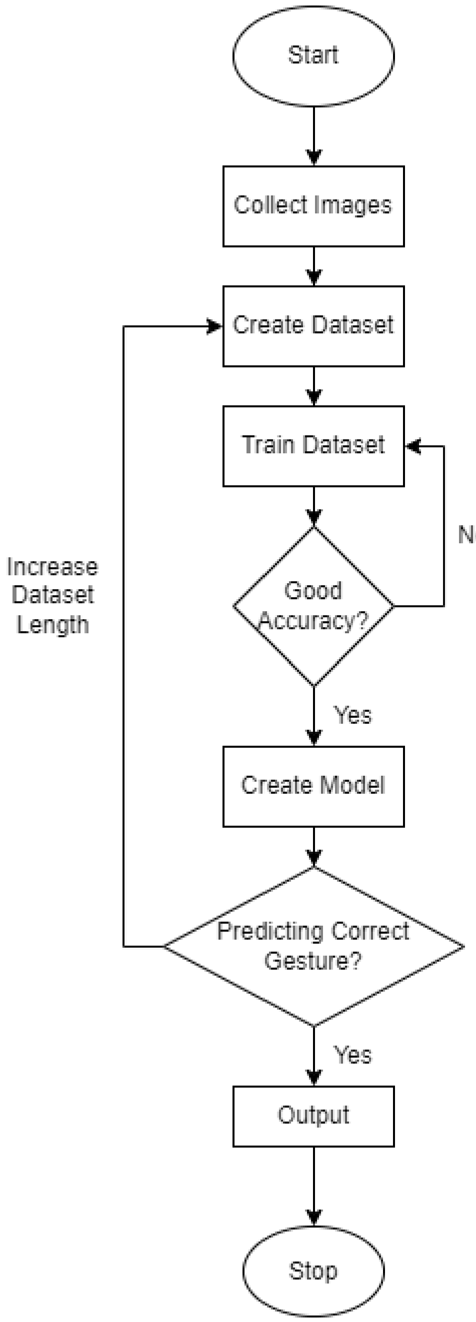


Figure 4.3.b: Flowchart of the system

### c. Outputs

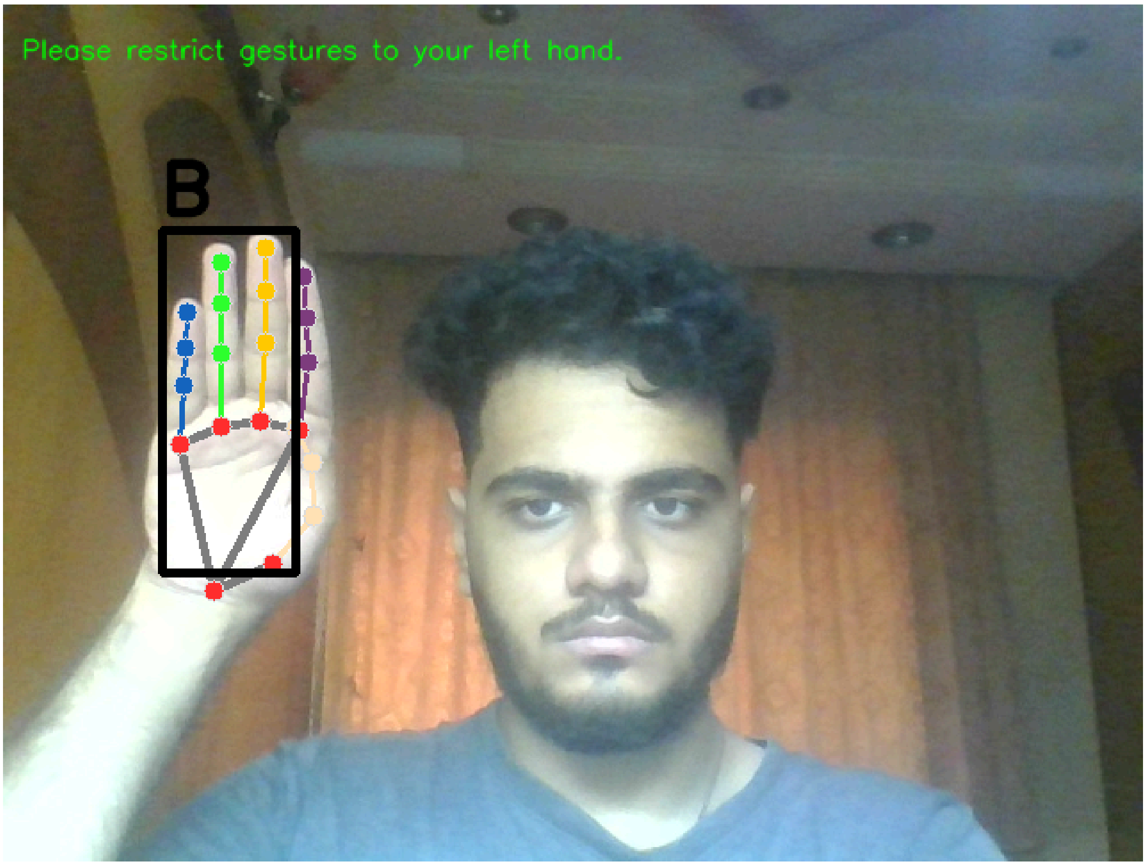


Figure 4.3.c.a: Prediction Testing

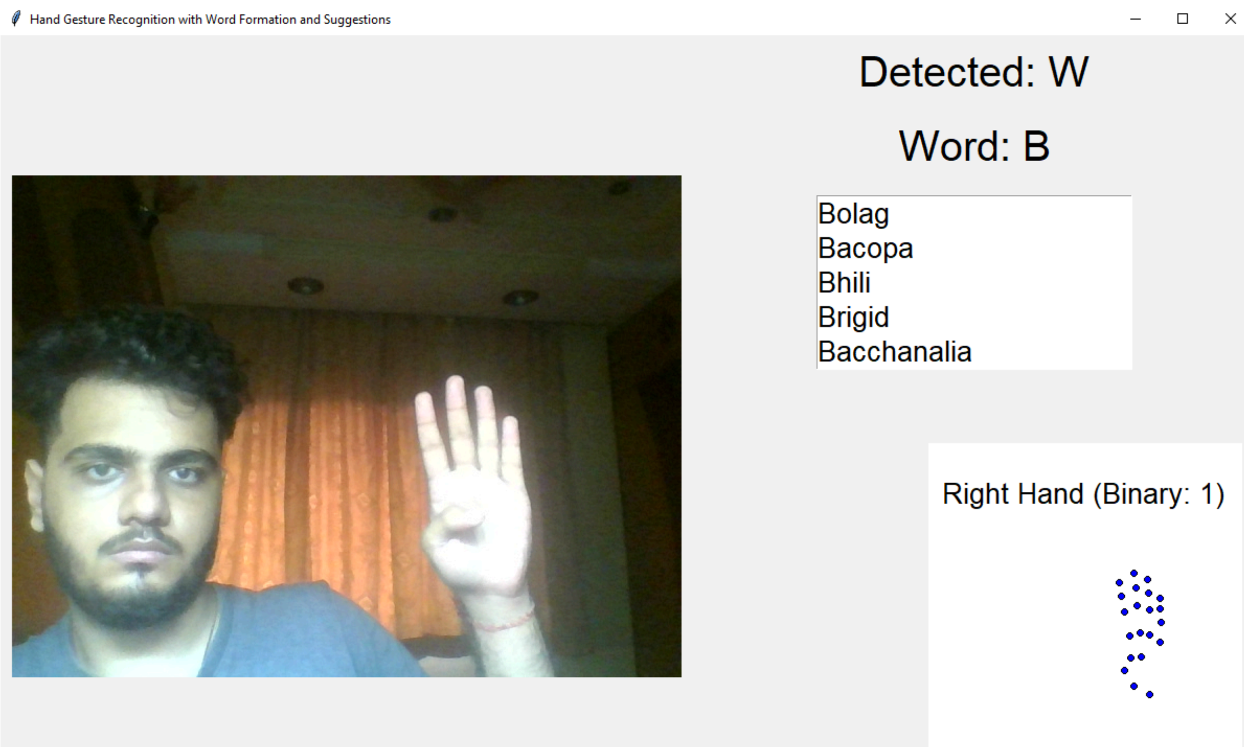


Figure 4.3.c.b: Final Output

# Chapter V: Plan Of Action For the Next Semester

### 6.1 Work done till date

- **Image Collection**

  Successfully captured and organized hand gesture images from a webcam into labeled folders, ensuring a diverse dataset that covers various gestures.

- **Landmark Detection**

  Extracted and normalized hand landmarks using the MediaPipe library for each gesture, which allows for precise tracking and analysis of hand movements.

- **Dataset Preparation**

  Processed images into a dataset with feature vectors and corresponding labels, storing them for training. This preparation included data augmentation techniques to enhance the dataset's robustness.

- **Model Training**

  Trained a Random Forest classifier using the prepared dataset, achieving a functional model for gesture classification. This involved fine-tuning hyperparameters to optimize performance and accuracy.

- **Real-time Inference**

  Implemented real-time gesture prediction, displaying results on live video feeds with accurate gesture detection. The system provides instant feedback on recognized gestures, enhancing interactivity.

- **Real-time Word Formation and Suggestions**

  Integrated real-time word formation and suggestions during runtime, allowing users to see potential word outputs corresponding to recognized gestures. This feature enhances communication for users by converting gestures into text in real-time, making interactions smoother and more intuitive.

## 6.2 Plan of action for project II

- **Indian Sign Language (ISL) Integration**:

  **Objective**: Adapt the gesture recognition system to effectively recognize Indian Sign Language gestures and their unique nuances.

  **Approach**: Collaborate with ISL experts to gather a comprehensive dataset of ISL gestures. Train the model specifically on this dataset to enhance its accuracy and responsiveness in recognizing ISL signs. The focus will be on capturing regional variations and dialects within ISL to ensure inclusivity.

- **British Sign Language (BSL) Integration**:

  **Objective**: Extend the system's capabilities to include British Sign Language, thereby enhancing accessibility for users in the UK and other regions where BSL is prevalent.

  **Approach**: Develop a dedicated dataset for BSL, incorporating a wide range of gestures used in everyday communication. Engage with BSL users and educators to validate gesture recognition accuracy and to improve user experience.

- **Incorporation of Other Local Sign Languages**:

  **Objective**: Broaden the recognition capabilities to include various local sign languages around the world, catering to diverse linguistic communities.

  **Approach**: Research and document the features of additional sign languages, create datasets for each language, and implement recognition algorithms tailored to their specific gestures. This initiative aims to promote inclusivity and accessibility for users from different cultural backgrounds.

- **Enhanced Communication Features**:

  **Objective**: Improve the communication effectiveness of the system by incorporating features that facilitate richer interactions.

**Approach**:

- ○ **Sentence Formation**: Develop algorithms that allow users to construct full sentences using gestures, making communication more fluid and coherent.
- ○ **Emoji Interpretation**: Integrate emoji recognition and interpretation to add emotional context to signed messages, making interactions more expressive.
- ○ **Numerical Gestures**: Include the ability to recognize and interpret numerical signs, allowing users to communicate quantitative information easily.
- ○ **Fixed Gestures**: Implement recognition for common fixed gestures, such as greetings ("good morning," "goodbye"), which can enhance everyday communication.

- **Gesture Customization**:

  **Objective**: Provide users with the ability to define and customize their own gestures for specialized communication needs.

  **Approach**: Develop a user-friendly interface that allows individuals to create and save personalized gestures, enhancing flexibility and personalization. This feature could be particularly beneficial for users with unique communication requirements or preferences.

- **Mouse Movement Control through Hand Gestures**:

  **Objective**: Enable users to control computer interfaces using hand gestures, providing a hands-free interaction option.

  **Approach**: Implement gesture recognition algorithms that translate specific hand movements into mouse actions, such as moving the cursor, clicking, or scrolling. This feature can significantly improve accessibility for users with limited mobility, allowing them to interact with digital environments intuitively.

# Chapter VII: Conclusion

The Silent Cue project represents a significant step forward in bridging communication gaps for deaf and non-verbal individuals. By utilizing advanced technologies in computer vision and machine learning, this project aims to create an innovative solution that translates sign language gestures into accessible text and spoken language. The successful implementation of this system not only enhances the quality of life for its users but also fosters inclusivity and understanding within diverse communities.

Through rigorous testing and continuous improvements, Silent Cue aspires to become a widely adopted tool that empowers individuals to express themselves freely and confidently. By raising awareness of sign language and promoting effective communication, this project contributes to a more inclusive society where every individual can engage meaningfully with others, regardless of their communication methods.

.

# Chapter VIII: References


[1] N. S. A. Syed, F. A. Mahmud, and M. R. Z. Shahar, "A Comprehensive Review on Hand Gesture Recognition Systems for Human-Computer Interaction," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 4, pp. 234–241, 2021. https://doi.org/10.14569/IJACSA.2021.0120430

[2] R. G. B. DeMello, A. G. R. C. Lira, and T. H. L. Lopes, "Deep Learning Approaches for Hand Gesture Recognition: A Systematic Review," *Sensors*, vol. 21, no. 10, pp. 3399, 2021. https://doi.org/10.3390/s21103399

[3] A. Alkhateeb, A. Alnatsheh, and R. Ghaleb, "Real-Time Hand Gesture Recognition Using Deep Learning Techniques," *Journal of King Saud University - Computer and Information Sciences*, 2022. https://doi.org/10.1016/j.jksuci.2022.10.012

[4] H. J. Lee, D. J. Kim, and H. K. Kim, "Hand Gesture Recognition for Human-Robot Interaction," *Robotics and Autonomous Systems*, vol. 133, pp. 103603, 2020. https://doi.org/10.1016/j.robot.2020.103603

[5] A. Jain, P. M. Roy, and A. B. Ghosh, "Hand Gesture Recognition System Based on Leap Motion and Machine Learning," *International Journal of Computing and Digital Systems*, vol. 10, no. 1, pp. 1–8, 2021. https://doi.org/10.12785/ijcds/100101

[6] D. I. M. Aparecida, P. J. D. M. Ferreira, and A. J. C. Freitas, "An Interactive Gesture-Based Communication System for Deaf People," *Multimedia Tools and Applications*, vol. 80, pp. 19445–19462, 2021. https://doi.org/10.1007/s11042-021-10426-x

[7] R. S. Malekian, S. S. Shahriari, and M. M. Masoumi, "A Novel Approach for Real-Time Hand Gesture Recognition Using Deep Learning," *IEEE Access*, vol. 9, pp. 43681–43692, 2021. https://doi.org/10.1109/ACCESS.2021.3060703

[8] C. W. Yu, Y. L. Ho, and C. L. Lee, "A Survey of Hand Gesture Recognition Techniques for Human-Computer Interaction," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 1965–1983, 2021. https://doi.org/10.1007/s12652-020-02738-2

[9] M. A. Chaudhry, A. J. Khan, and M. H. Jamil, "Sign Language Recognition for Deaf-Mute Communication: A Comprehensive Survey," *Future Generation Computer Systems*, vol. 114, pp. 63–79, 2021. https://doi.org/10.1016/j.future.2020.08.012

[10] S. P. A. Abduallah, A. M. G. Al-Qinai, and A. M. Al-Mansoori, "Hand Gesture Recognition Using Machine Learning Algorithms," *Computer Applications in Engineering Education*, vol. 30, pp. 611–622, 2022. https://doi.org/10.1002/cae.22456

[11] F. I. Ali, A. T. Alharthy, and H. S. Alharbi, "Sign Language Recognition Using Neural Networks: A Review," *International Journal of Information Technology and Computer Science*, vol. 13, no. 2, pp. 1–8, 2021. https://doi.org/10.5815/ijitcs.2021.02.01

[12] M. S. A. Amjadian, M. N. Z. R. Ghazali, and R. H. J. Hussain, "A Framework for Continuous Sign Language Recognition Based on Convolutional Neural Networks," *Computers in Human Behavior*, vol. 122, pp. 106832, 2021. https://doi.org/10.1016/j.chb.2021.106832

# Chapter IX: Appendix

- List Of Figures


| Sr. no. | Figure No. | Name of the figure | Page No. |
|---|---|---|---|
| | 1.6 | Methodology Employed | 3 |
| | 4.1 | Block diagram of the system | 10 |
| | 4.2 | Modular diagram of the system | 11 |
| | 4.3.a.a | Level 0 DFD | 12 |
| | 4.3.a.b | Level 1 DFD | 12 |
| | 4.3.a.c | Level 2 DFD | 13 |
| | 4.3.b | Flowchart of the system | 14 |


**List Of Tables**


| Sr. no. | Table No. | Name of the Table | Page No. |
|---|---|---|---|
| | 3.4 | Hardware & Software Requirements | 7 |
| | 5.1 | Evaluation Metrics used | 16 |
| | 5.2 | Evaluation metrics of the system | |

## Xerox of project review sheet

**Industry/Inhouse:** **Project Evaluation Sheet 2024-25** **Class: D17** A, B

49

Title of Project(Group no): Silent Cue (49)

Group Members: Jiya Gangwani (D17A-47), Nikhil Dhanwani (D17B-11), Soham Panjabi (D17B-37), Chirag Santwani (D17B-47)

| | Engineering Concepts & Knowledge (5) | Interpretation of Problem & Analysis (5) | Design / Prototype (5) | Interpretation of Data & Dataset (3) | Modern Tool Usage (5) | Societal Benefit, Safety Consideration (2) | Environment Friendly (2) | Ethics (2) | Team work (2) | Presentation Skills (3) | Applied Engg &Mgmt principles (3) | Life-long learning (3) | Professional Skills (5) | Innovative Approach (5) | Total Marks (50) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Review of Project Stage 1 | 4 | 4 | 3 | 2 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 4 | 4 | 37 |

Comments:

M. D. Patil.
Name & Signature Reviewer1

| | Engineering Concepts & Knowledge (5) | Interpretation of Problem & Analysis (5) | Design / Prototype (5) | Interpretation of Data & Dataset (3) | Modern Tool Usage (5) | Societal Benefit, Safety Consideration (2) | Environment Friendly (2) | Ethics (2) | Team work (2) | Presentation Skills (3) | Applied Engg &Mgmt principles (3) | Life-long learning (3) | Professional Skills (5) | Innovative Approach (5) | Total Marks (50) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Review of Project Stage 1 | 4 | 3 | 4 | 1 | 2 | 2 | 2 | 2 | 2 | 3 | 2 | 2 | 4 | 4 | 38 |

Comments:

Date: 23rd August, 2024

Prerna Solanke
Name & Signature Reviewer2

**Industry/Inhouse:** **Project Evaluation Sheet 2024-25** **Class: D17** A, B

49

Title of Project(Group no): Silent Cue (Group No. 49)

Group Members: Jiya Gangwani (D17A-47), Nikhil Dhanwani (D17B-11), Soham Panjabi (D17B-36), Chirag Santwani (D17B-47)

| | Engineering Concepts & Knowledge (5) | Interpretation of Problem & Analysis (5) | Design / Prototype (5) | Interpretation of Data & Dataset (3) | Modern Tool Usage (5) | Societal Benefit, Safety Consideration (2) | Environment Friendly (2) | Ethics (2) | Team work (2) | Presentation Skills (3) | Applied Engg &Mgmt principles (3) | Life-long learning (3) | Professional Skills (5) | Innovative Approach (5) | Total Marks (50) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Review of Project Stage 1 | 4 | 3 | 4 | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 5 | 4 | 41 |

Comments:

M D. Patil 26/09/24
Name & Signature Reviewer1

| | Engineering Concepts & Knowledge (5) | Interpretation of Problem & Analysis (5) | Design / Prototype (5) | Interpretation of Data & Dataset (3) | Modern Tool Usage (5) | Societal Benefit, Safety Consideration (2) | Environment Friendly (2) | Ethics (2) | Team work (2) | Presentation Skills (3) | Applied Engg &Mgmt principles (3) | Life-long learning (3) | Professional Skills (5) | Innovative Approach (5) | Total Marks (50) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Review of Project Stage 1 | 4 | 4 | 4 | 3 | 3 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 5 | 4 | 43 |

Comments:

Date: 26th September, 2024

Prerna Solanke
Name & Signature Reviewer2