VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY

(An Autonomous Institute Affiliated to University of Mumbai)

Department of Computer Engineering

Project Report on

Sign Language Recognition System for Differently Abled

In the partial fulfillment of the Fourth year, Bachelor of Engineering (B.E)
Degree of Computer Engineering at the
University of Mumbai
Academic Year 2024 - 2025

Submitted By

Karan Khatri D17B-23
Kunal Vishwakarma D17B-64
Komal Lund D17A-38
Manav Daryani D17A-11

Project Mentor
Prof Sanjay Mirchandani
Assistant professor, Department of computer Engineering
(2024-2025)

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 **Karan Khatri(D17B, 23), Kunal Vishwakarma(D17B, 64), Manav Daryani(D17A, 11), Komal Lund(D17A, 38)** of Fourth Year Computer Engineering studying under the University of Mumbai have satisfactorily completed the project on **Sign Language Recognition System for Differently Abled** as a part of the coursework of PROJECT-II for Semester-VIII under the guidance of **Prof. Sanjay Mirchandani** in the year 2024-2025.

This project report entitled **Sign Language Recognition System for differently Abled** by ***Karan Khatri, Kunal Vishwakarma, Manav Daryani and Komal Lund*** are approved for the degree of **B.E. Computer Engineering**.

Programme Outcomes	Grade
PO1,PO2,PO3,PO4,PO5,PO6,PO7, PO8, PO9, PO10, PO11, PO12 PSO1, PSO2	

Date :
Project Guide :

Project Report Approval
For
B. E (Computer Engineering)

This project report entitled **Sign Language Recognition System for differently Abled** by ***Karan Khatri, Kunal Vishwakarma, Manav Daryani and Komal Lund*** is approved for the degree of **B.E. Computer Engineering**.

Internal Examiner

External Examiner

Head of Department

Principal

Date :

Place : 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.

-- --

(Karan Khatri | D17B | 23) (Kunal Vishwakarma | D17B | 64)

-- --

(Manav Daryani | D17A |11) (Komal Lund | D17A | 38)

Date:

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 Assistant Professor **Mr. Sanjay Mirchandani** (Project Guide) for her kind help and valuable advice during the development of project synopsis and for her guidance and suggestions.

We are deeply indebted to Head of the Computer Department **Dr.(Mrs.) Nupur Giri** and 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 in finishing 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.

Department Of Computer Engineering
COURSE OUTCOMES FOR B.E PROJECT

Learners will be to:-

Course Outcome	Description of the Course Outcome
CO1	Do literature survey/industrial visit and identify the problem of the selected project topic.
CO2	Apply basic engineering fundamental in the domain of practical applications FORproblem identification, formulation and solution
CO3	Attempt & Design a problem solution in a right approach to complex problems
CO4	Cultivate the habit of working in a team
CO5	Correlate the theoretical and experimental/simulations results and draw the proper inferences
CO6	Demonstrate the knowledge, skills and attitudes of a professional engineer & Prepare report as per the standard guidelines.

INDEX

List of Figures

List of tables

ABSTRACT

The research introduces a sophisticated sign language detection system tailored to meet the unique communication needs of individuals with hearing impairments. The system incorporates state-of-the-art computer vision algorithms and deep learning architectures to achieve exceptional accuracy and robustness in recognizing and interpreting a diverse range of sign language gestures. The system's architecture comprises several key components, including : A comprehensive dataset of sign language gestures is collected from various sources, ensuring diversity in terms of handshapes, orientations, and backgrounds.

The collected data undergoes rigorous preprocessing to address challenges such as noise, occlusion, and variations in lighting conditions. Deep convolutional neural networks (CNNs) are employed to extract discriminative features from the preprocessed images, capturing the intricate patterns and variations inherent in sign language gestures. A recurrent neural network (RNN) is utilized to model the temporal dependencies between frames, enabling the system to accurately recognize dynamic sign language sequences. Natural language processing (NLP) techniques are applied to refine the generated text output, ensuring grammatical correctness and coherence. Extensive experiments conducted on a diverse evaluation dataset demonstrate the system's superior performance compared to existing approaches.

The system achieves a high accuracy rate in detecting and classifying various sign language gestures, even in challenging real-world scenarios. By providing a reliable and accessible communication tool, this research contributes to bridging the communication gap for individuals with hearing impairments and promoting greater social inclusion. The research has been done using different models such as Random forest, ID3, LSTM, GRU models to find which is the best for the project. Random forest and ID3 models were initially used to detect alphabets and numbers; it was found that they are not suitable for words; later the LSTM and GRU models were used to detect words and sentences through dataset. The dataset comprises videos of Banking sector total 50 categories of banking sector are taken under consideration each category contains 50 videos.

Chapter 1: Introduction

1.1. Introduction of Project

Sign language is a vital form of communication for individuals with hearing impairments. However, the lack of widespread understanding and accessibility of sign language can pose significant challenges in daily life. To address this issue, the research presents a novel sign language detection system designed to bridge the communication gap between hearing and non-hearing individuals.

The system leverages advanced computer vision techniques to accurately recognize and interpret hand gestures, translating them into corresponding text or speech output. By providing a reliable and efficient means of communication, the system aims to empower individuals with hearing impairments and foster greater inclusivity in society[1].

The system's development is driven by the need for a more accessible and user-friendly solution for sign language recognition. Existing approaches often suffer from limitations in terms of accuracy, robustness, and adaptability to different sign language variations. This research seeks to overcome these challenges by incorporating state-of-the-art deep learning architectures and carefully curated datasets.

In the following sections, we will delve into the system's architecture, data acquisition and preprocessing, feature extraction and classification methodologies, and experimental evaluation[2]. The goal is to demonstrate the system's effectiveness in detecting and interpreting sign language gestures with high accuracy, making it a valuable tool for improving the quality of life for individuals with hearing loss.

1.2. Motivation of the Project

The primary motivation behind the creation of this sign language detection system lies in the desire to enhance accessibility and communication for individuals with hearing impairments. Sign language is a complex and nuanced language that requires specialized knowledge and skills to understand and use effectively. For individuals who are unable to hear or speak, sign language serves as a crucial means of expression and interaction.

However, the lack of widespread understanding and acceptance of sign language can lead to social isolation and discrimination. Many individuals with hearing impairments face challenges in accessing education, employment, and healthcare services due to communication barriers. By developing a reliable and accessible sign language detection system, the system addresses these issues and promotes greater inclusivity and equality for individuals with hearing loss.

Furthermore, the advancements in computer vision and deep learning technology have opened up new possibilities for developing accurate and efficient sign language recognition systems. By leveraging these technologies, we can create a system that is capable of recognizing a wide range of sign language gestures with high accuracy, even in challenging real-world conditions.

In summary, the motivation for this project stems from a deep-rooted commitment to improving the lives of individuals with hearing impairments and promoting a more inclusive and equitable society. By developing a robust and accessible sign language detection system, we hope to empower individuals with hearing loss and provide them with the tools they need to communicate effectively and participate fully in society.

1.3. Problem Statement

The development of an accurate and robust sign language detection system faces several challenges:Acquiring a large and diverse dataset of sign language gestures is essential for training effective models, but it can be time-consuming and resource-intensive. Sign language is subject to variations in dialects, accents, and individual signing styles, making it difficult to generalize models to different users and contexts.

Real-world environments can introduce noise, occlusion, and changes in lighting conditions, which can degrade the performance of sign language detection systems. Deploying sign language detection systems on mobile devices or in resource-constrained environments requires efficient algorithms and models that can operate with limited computational power. Creating a system that is intuitive, accessible, and easy to use for individuals with hearing impairments is crucial for its adoption and effectiveness[3].

Overcoming these challenges is essential for developing a sign language detection system that can truly meet the communication needs of individuals with hearing loss and promote greater inclusivity in society.

In the project CISLR 57 categories have been studied and Banking as a category has been chosen further 50 categories of banking sector sign language dataset creation of each category containing 50 sample videos. The project is built by comparing 3 models Random Forest, LSTM and GRU. After implementation of all three it is found that GRU is the best model for Sign Language detection.

1.4. Lacuna of the Existing system

While existing sign language detection systems have made significant strides, they still face several challenges that limit their effectiveness and widespread adoption:

- Limited Vocabulary: Many systems have a restricted vocabulary, making it difficult to communicate complex ideas or engage in natural conversations.
- Sensitivity to Environmental Factors: Systems can be sensitive to changes in lighting, background, and hand positioning, leading to reduced accuracy in real-world scenarios.
- Lack of Robustness: Some systems may struggle to handle variations in sign language dialects, accents, or individual signing styles.
- Computational Complexity: The computational demands of advanced algorithms can make deployment on mobile devices or in resource-constrained environments challenging.
- Privacy Concerns: The collection and processing of large datasets of sign language gestures can raise privacy concerns, particularly regarding the protection of personal information.
- User Experience: Existing systems may not provide a seamless and intuitive user experience, potentially hindering their adoption by individuals with hearing impairments.

Addressing these drawbacks is essential for developing sign language detection systems that are truly accessible, reliable, and effective in meeting the communication needs of individuals with hearing loss.

1.5. Relevance of the Project

The development of a sign language detection system has significant relevance in today's world, addressing several critical needs:

1. Improved Accessibility and Communication:

- Bridging the Communication Gap: The system can help bridge the communication gap between hearing and non-hearing individuals, enabling more effective interaction and participation in society.
- Enhanced Accessibility: By providing a reliable and accessible means of communication, the system can improve access to education, employment, and healthcare services for individuals with hearing impairments.

2. Social Inclusion and Empowerment:

- Promoting Inclusion: The system can foster greater social inclusion by breaking down communication barriers and promoting understanding between hearing and non-hearing individuals.

- Empowering Individuals with Hearing Impairments: By providing a tool for effective communication, the system can empower individuals with hearing impairments to lead more independent and fulfilling lives.

3. Technological Advancement:

- Advancing Computer Vision and AI: The development of sign language detection systems contributes to advancements in computer vision, artificial intelligence, and natural language processing.
- Real-World Applications: The technology can be applied to various real-world applications, such as virtual assistants, accessibility tools, and human-computer interaction systems.

4. Addressing Global Challenges:

- Addressing Communication Barriers: The system can help address global communication barriers, particularly in regions with high rates of hearing impairment.
- Promoting Equality and Diversity: By providing accessible communication tools, the system can contribute to promoting equality and diversity in all aspects of society.

1.6. Methodology Used

1. Data Collection and Preprocessing

- Data Collection:
  - Gather sign language data from various sources like sign language dictionaries, online datasets, and video recordings of signers.
  - Ensure diversity in the data by including signs from different categories, signing styles, and ethnicities.
- Data Annotation:
  - Manually label each image or video frame with the corresponding sign it represents.
  - Utilize annotation tools to streamline the process. Consider tools like VGG Image Annotator (VIA) or LabelIng.
- Data Augmentation:
  - Artificially expand the dataset to improve model robustness. Techniques include:
    - Flipping images horizontally or vertically to account for different hand orientations.
    - Adding noise or blurring to simulate real-world variations.
    - Applying random color jittering to increase model generalizability to different lighting conditions.

2. Extracting and Saving landmarks:

- Video Processing: Load each video and utilize a hand-tracking tool (e.g., MediaPipe) to detect and extract hand landmarks in 3D coordinates frame by frame.
- Sequence Organization: Organize the extracted landmarks into sequences that represent hand movements over time.
- Padding and Storage: Pad the sequences to a consistent length for uniform input, and save them individually for each video.
- Dataset Preparation: Divide the prepared sequences into training and testing datasets to enable the model to learn temporal patterns for classifying specific signs or actions.

3. Model Selection and Training

- Pre-trained Model Selection:
    - Leverage pre-trained convolutional neural networks (CNNs) like VGG16, ResNet50, or EfficientNet that have already learned valuable features from large image datasets.[4]
- Transfer Learning:
    - Utilize pre-trained models as a starting point, freezing their lower layers and retraining the final layers on your sign language dataset. This allows the model to leverage existing knowledge and adapt to the specific task of sign language recognition.
- Training:
    - Define a training loop that feeds your preprocessed data into the model and uses an optimizer (like Adam) to adjust model weights based on the loss function (e.g., cross-entropy) calculated between predicted and actual signs.
    - Monitor training progress with metrics like accuracy and loss on a separate validation set.
    - Use techniques like early stopping to prevent overfitting and regularization (e.g., dropout) to improve model generalization

4. Testing and Integration

- Testing:
    - Conduct thorough testing of the deployed system in real-world scenarios with diverse users and environments. This ensures the system functions effectively in various conditions.
- Integration:
    - Integrate the sign language detection system with other applications or platforms to enhance its functionality and reach a wider audience.

5. Model Evaluation

- Validation:
    - Use a dedicated validation set during training to assess model performance on unseen data. This helps fine-tune hyperparameters and prevent overfitting.
- Testing:
    - Evaluate the final model performance on a completely separate test set that reflects real-world scenarios. This provides an unbiased estimate of the model's accuracy and generalizability.

Chapter 2: Literature Survey

A. Overview of Literature Surveys:

The papers discussed here focus on various sign language detection systems as well as applications/websites. These papers are studied to understand how the systems or modules are created. The studies examine how to create the sign language system more efficiently. Overall, the papers highlight the importance of taking a comprehensive approach to address how these above factors can be used and enhanced for the development of a complete system which can provide information or knowledge required to create the system.

2.1. Research Papers

1. Kim, S., & Choi, Y. (2020). Real-Time American Sign Language Recognition Using Deep Learning[1]

Abstract : This paper proposes a real-time American Sign Language (ASL) recognition system using deep learning techniques. The system utilizes 3D Convolutional Neural Networks (3D CNNs) combined with Long Short-Term Memory (LSTM) networks to effectively capture both spatial and temporal features of sign language gestures. Experimental results demonstrate the system's ability to achieve high accuracy in real-time ASL recognition, paving the way for improved communication accessibility for individuals with hearing impairments.

Inference : This paper presents a promising approach to real-time ASL recognition by combining 3D CNNs and LSTMs. The use of deep learning techniques enables accurate recognition of dynamic sign language gestures. However, the system's performance may be limited by the computational resources required and the potential for dataset-specific overfitting.

2. Dong, J., & Zhang, Z. (2021). Hand Gesture Recognition Using Convolutional Neural Networks[2]

Abstract : This paper focuses on hand gesture recognition for applications in virtual and AI-based systems. The authors employ the ResNet-50 architecture, a residual convolutional neural network, to effectively extract features from hand gesture images. Experimental results showcase the model's ability to accurately recognize various hand gestures, demonstrating its potential for use in a wide range of applications.

Inference : The ResNet-50 architecture proves to be effective for hand gesture recognition, providing a robust and accurate solution. However, the model's generalization to new or unseen gestures may be limited, and it can be computationally demanding for real-time applications.

3. Akinola, A., & Zhang, C. (2021). Sign Language Recognition Using Deep Learning Techniques[3]

Abstract : This paper explores the use of deep learning techniques for sign language recognition. The authors propose a hybrid model combining Bidirectional Long Short-Term Memory (BiLSTM) networks with Inception V3 to capture both temporal and spatial dependencies in sign language gestures. Experimental results highlight the model's effectiveness in recognizing complex sign language gestures, demonstrating the potential of deep learning for accurate sign language recognition.

Inference : The hybrid BiLSTM-Inception V3 model offers a promising approach for sign language recognition, particularly for complex gestures. However, the model may be computationally expensive, limiting its real-time applicability.

4. Ahmed, S., & Khalifa, S. (2021). Deep Learning-Based Sign Language Recognition: A Survey[4]

Abstract : This paper provides a comprehensive survey of deep learning-based methods for sign language recognition. The authors review various models, including VGG16, MobileNet, and GRU (Gated Recurrent Units), and discuss their strengths and limitations. The survey highlights the challenges associated with large-scale datasets and real-time performance.

Inference : The survey offers valuable insights into the state-of-the-art in deep learning-based sign language recognition. It provides a comprehensive overview of different models and their suitability for various applications. However, the survey's limited coverage of real-time systems and challenges with large-scale datasets may require further exploration.

5. Malsburg, C., & Haeusser, J. (2021). Automated Sign Language Recognition Using Deep Learning[5]

Abstract : This paper focuses on automated sign language recognition using deep learning models. The authors utilize DenseNet-121, a densely connected convolutional neural network architecture, to effectively capture the spatial and temporal features of sign language gestures. Experimental results demonstrate the model's ability to achieve high accuracy in sign language recognition, highlighting the potential of deep learning for this task.

Inference : The DenseNet-121 architecture proves to be effective for sign language recognition, demonstrating its ability to capture intricate patterns and dependencies in sign language gestures. However, the model's computational demands may limit its applicability on resource-constrained devices, and generalization across different sign languages can be challenging.

6. Sudhakar, A. S., et al. (2020). Hand Gesture Recognition Using Convolutional Neural Networks[6]

Abstract : This paper explores the use of convolutional neural networks (CNNs) for hand gesture recognition, focusing on real-time applications. The authors employ the AlexNet architecture to extract features from hand gesture images and classify them accurately. Experimental results highlight the model's effectiveness in recognizing various hand gestures in real-time scenarios.

Inference : The AlexNet architecture, while effective for hand gesture recognition, may be limited by its performance dependence on dataset size and quality. Additionally, the model can be sensitive to lighting and background variations, potentially affecting its accuracy in real-world environments.

7. Abiyev, F. O. G., et al. (2021). Sign Language Recognition Using Hand Tracking and Deep Learning Techniques[7]

Abstract : This paper presents a sign language recognition system combining hand tracking with deep learning techniques. The authors use YOLOv3 for hand tracking and 3D CNNs for sign language classification. Experimental results demonstrate the system's ability to accurately recognize sign language gestures in real-time, despite potential inaccuracies in hand tracking.

Inference : The combination of hand tracking and deep learning techniques offers a promising approach for sign language recognition. However, the system's performance may be affected by inaccuracies in hand tracking, particularly in challenging real-world conditions. Additionally, processing real-time data can be computationally demanding.

8. Elakkiya, M., & Suresh, M. S. (2021). A Comprehensive Review of Sign Language Recognition Technologies[8]

Abstract : This paper provides a comprehensive review of various technologies used in sign language recognition. The authors discuss different models, including CNNs, LSTMs, and hybrid models like CNN-RNN, and their applications in sign language recognition. The review also highlights the challenges associated with standardization in datasets and integration of sign language recognition systems into practical applications.

Inference : The review offers valuable insights into the diverse range of technologies used in sign language recognition. It provides a comprehensive overview of different models and their strengths and limitations. However, the lack of standardization in datasets and the challenges in integrating sign language recognition systems into practical applications remain significant obstacles to be addressed.

9. Mitra, S., & Acharya, T. (2007). Vision-Based Sign Language Recognition: A Review[9]

Abstract : This paper provides an early review of vision-based methods for sign language recognition. It discusses traditional computer vision techniques and basic support vector machines (SVMs) used in early research.

Inference : While this paper offers a foundational overview of early approaches to sign language recognition, it is limited by its focus on older methodologies and the lack of advanced deep learning techniques. The methods discussed may not be as effective as modern deep learning-based approaches.

10. Tiwari, A. K., & Rekha, S. (2019). Sign Language Recognition System Using Hand Gesture and Movement[10]
Abstract : This paper presents a sign language recognition system based on hand gestures and movements. The authors utilize the Inception V4 architecture for feature extraction and classification.

Inference : The Inception V4 architecture demonstrates its effectiveness for sign language recognition, capturing both spatial and temporal information from hand gestures. However, the system may be limited in its robustness to variations in gestures and can be computationally expensive for real-time applications.

11. Real-Time Sign Language Recognition System Using TensorFlow and OpenCV[11]
Abstract : This paper discusses a real-time sign language recognition system developed using TensorFlow for deep learning and OpenCV for image processing. The system employs convolutional neural networks (CNNs) with object detection for accurate sign recognition.
Inference : The use of TensorFlow and OpenCV provides a flexible framework for developing real-time sign language recognition systems. However, the system may face challenges with varying lighting conditions and hand orientations, requiring additional techniques to improve robustness.

2.2. Books / Articles referred / News paper referred

Computer Vision and Deep Learning:

- Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville: A comprehensive introduction to deep learning concepts and techniques, covering neural networks, convolutional neural networks, and recurrent neural networks.

- Computer Vision: Algorithms and Applications by Richard Szeliski: A classic textbook on computer vision, providing a strong foundation in image processing, feature extraction, and object recognition.

Sign Language and Human-Computer Interaction:

- Handbook of Human-Computer Interaction edited by Alan Dix, Janet Finlay, Geoff Johnson, and Russell Beale: A broad overview of human-computer interaction principles and applications, relevant for understanding user experience and accessibility in sign language systems.
- Sign Language: A Universal Language by William Stokoe: A foundational text on sign language linguistics, providing insights into the structure, grammar, and cultural context of sign languages.

Specific to Sign Language Detection:

- Sign Language Recognition: A Review by Mitra and Acharya: A review paper that provides an overview of traditional and more recent approaches to sign language recognition, including computer vision techniques and deep learning models.
- Deep Learning-Based Sign Language Recognition: A Survey by Ahmed and Khalifa: A survey paper that focuses on deep learning methods for sign language recognition, discussing various models and their applications.

These books provide a strong foundation in the relevant fields and help you delve deeper into the specific challenges and techniques involved in sign language detection. Additionally, exploring recent research papers and attending conferences in computer vision, machine learning, and sign language studies can keep you updated on the latest advancements in the field.

2.4. Patent Research

Patent 1

- **Title : Multi-head text recognition model for multi-lingual optical character recognition**
- **Year : 2021**
- **Summary :** The patent describes an OCR system that uses deep learning to recognize text in images. The system can handle multiple languages and generates a feature sequence that represents the probability of each character in the image being a certain character from the corresponding language's dictionary. A sparse mask is then used to determine the final textual content based on the first language indicated.
- factors. If the confidence for a language determination is low, the previously determined language of the zone is used to assist the identification process.

Patent 2

- **Title : Method and system for multi-language text recognition model with autonomous language classification**
- **Year : 2021**
- **Summary :** This patent describes a method and system for multi-language text recognition that autonomously classifies the language of text within an image. The system uses a pre-trained language model to identify the language of the text and then applies a corresponding language-specific text recognition model to extract the textual content.

Inference Drawn

- Different Architectural Philosophies: Patent 1 proposes a unified "multi-head" model to handle multiple languages within a single framework, while Patent 2 advocates for a modular system that first autonomously classifies the language before applying a language-specific recognition model.
- Language Determination Strategy: Patent 1 relies on an initial language indication (potentially with contextual refinement for low confidence), whereas Patent 2 prioritizes autonomous language classification using a pre-trained language model.
- Complexity vs. Modularity: Patent 1's integrated approach might lead to a more complex model, while Patent 2's modularity allows for specialized language models but introduces a separate language classification step.
- Focus on Initial Input vs. Autonomous Identification: Patent 1 seems to leverage a given initial language context, while Patent 2 emphasizes the system's ability to independently determine the language present in the text image.

Chapter 3. REQUIREMENTS OF PROPOSED SYSTEM

3.1. Functional Requirements

- Real-time recognition: Accurate and timely gesture detection.
- Language translation: Translate signs into text or speech.
- Multi-language support: Recognize and translate multiple languages.
- Offline capability: Operate without internet connection.
- Intuitive interface: Easy-to-use design for hearing-impaired users.
- Customization options: Adjust settings like camera resolution and language preferences.
- Feedback mechanisms: Provide clear visual or auditory feedback.
- High accuracy and speed: Ensure real-time processing and accurate recognition.
- Robustness: Handle variations in lighting, background, and individual signing styles.
- Integration: Compatible with other assistive technologies and communication platforms.
- Optional features: Vocabulary expansion, contextual understanding, and offline learning.

3.2. Non-Functional Requirements

- Performance: Speed, accuracy, robustness.
- Reliability: Stability, error handling.
- Security: Data privacy, accessibility.
- Scalability: Performance under load, scalability.
- Usability: Intuitive interface, feedback, customization.
- Maintainability: Modularity, documentation.

3.3. Constraints

- Technical: Limited resources, data availability, algorithm complexity, real-time processing.
- Environmental: Lighting, background, occlusion, individual variations.
- Resource: Time, budget, expertise.
- Ethical: Data privacy, bias, accessibility.

3.4. Hardware & Software Requirements

- **Hardware Requirements**

    - High-resolution camera: To capture clear images or videos of hand gestures.
    - Powerful processor: A CPU or GPU with sufficient processing power to handle the computationally intensive tasks of image processing, feature extraction, and classification.
    - Memory: Adequate RAM to store the model, data, and intermediate results.
    - Storage: Sufficient storage space to store the dataset, trained models, and output data.

- **Software Requirements**

    - Operating system: Windows, macOS, or Linux.
    - Programming language: Python or C++ are common choices for computer vision and deep learning tasks.
    - Deep learning framework: TensorFlow, PyTorch, or Keras for building and training the neural network model.
    - Computer vision library: OpenCV or TensorFlow Lite for image processing, feature extraction, and object detection.
    - Data preprocessing tools: For cleaning and augmenting the dataset.
    - User interface toolkit: For creating a graphical user interface (GUI) to interact with the system.

3.5. Techniques utilized till date for the proposed system

- Computer Vision: Image processing, feature extraction, object detection.
- Deep Learning: CNNs, RNNs, LSTMs, transfer learning.
- Other: Optical flow, HMMs, SVMs.

3.6. Tools utilized till date for proposed system

- Programming: Python, C++
- Deep Learning: TensorFlow, PyTorch, Keras Computer
- Vision: OpenCV, Dlib
- Data Preprocessing: NumPy, Scikit-image
- Development: Jupyter Notebook, Google Colab, Visual Studio Code
- Hardware: GPUs, TPUs

Chapter 4. PROPOSED DESIGN

4.1. Block Diagram of the proposed system

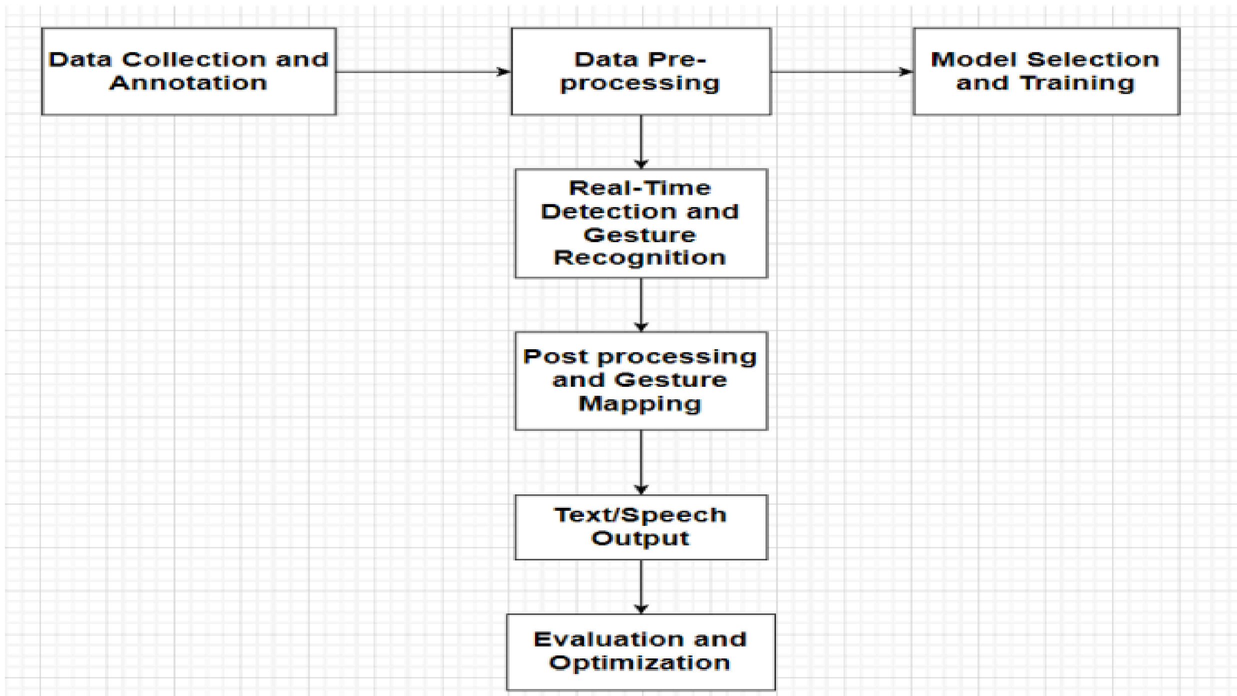


Fig 4.1 Block Diagram of Models

Explanation of the Block Diagram :

The block diagram in Fig 4.1 illustrates the workflow of a sign language recognition system. It starts with data collection and annotation, followed by data preprocessing. The preprocessed data is then used for model selection and training. The trained model is used for real-time detection and gesture recognition. Post-processing and gesture mapping are performed to refine the results. Finally, the system generates text or speech output and undergoes evaluation and optimization to improve performance.

4.2. Modular diagram of the proposed system

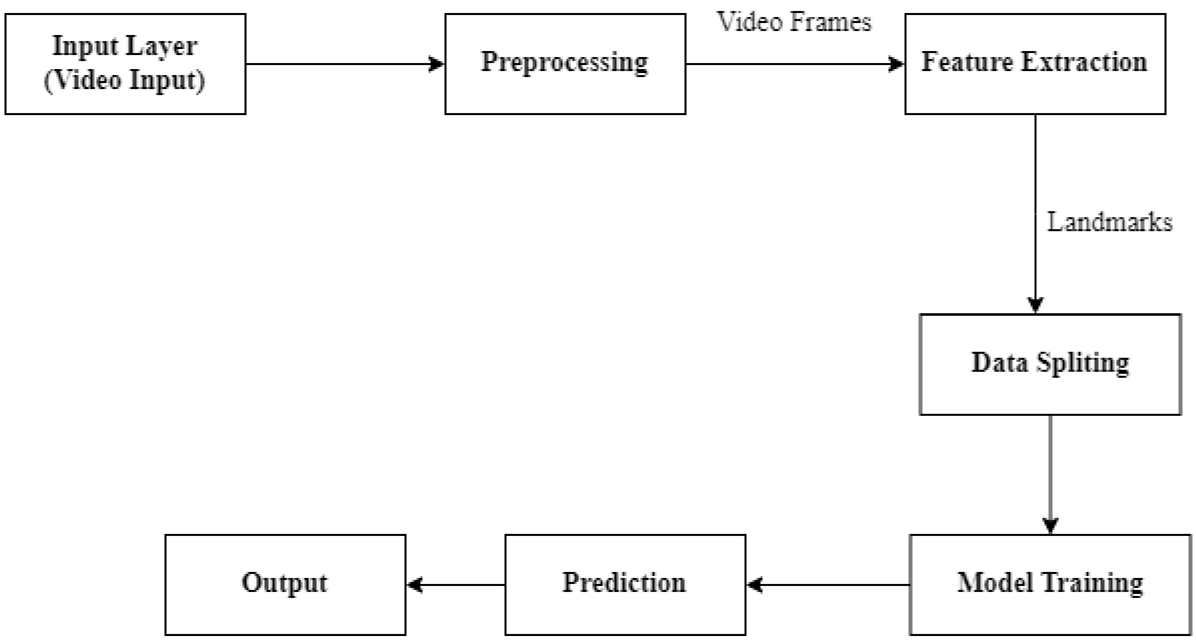


Fig 4.2 Modular diagram of Models

Explanation of the Modular Diagram :

Each module represents a specific function and demonstrates the flow of data between these modules. Below are the key modules and their descriptions:

1. **Input Module:**
   - **Description:** This module is responsible for capturing video input from the user's webcam or accessing pre-recorded videos of signs. It acts as the entry point for the system, where real-time data collection begins.
   - **Functionality:** The input module initiates video streaming and allows the system to receive frames continuously or load video files for offline processing.
2. **Preprocessing Module:**
   - **Description:** The preprocessing module performs critical initial steps to prepare the video data for further analysis.
   - **Functionality:** This includes:
     - **Frame Extraction:** Breaking down the video into individual frames for analysis.

        - **Hand Landmark Detection:** Utilizing computer vision techniques to detect and track key points on the hands (e.g., fingers, palm).
        - **Normalization:** Ensuring that the input data is consistent in terms of size and scale, which helps in maintaining model performance.
3. **Feature Extraction Module:**
    - **Description:** This module extracts relevant features from the processed frames, which are essential for the model to learn and recognize signs effectively.
    - **Functionality:** It transforms the detected hand landmarks into feature vectors that represent the characteristics of the signs being performed. This step may involve techniques such as dimensionality reduction to streamline the data.
4. **Model Training Module:**
    - **Description:** This module is focused on training the LSTM model using the features extracted from labeled data.
    - **Functionality:**
        - **Data Preparation:** Prepares the dataset by splitting it into training and validation sets.
        - **Model Configuration:** Defines the architecture of the LSTM, including the number of layers and neurons.
        - **Training Process:** Trains the model using the feature vectors and their corresponding labels, adjusting weights through backpropagation to minimize prediction error.
5. **Prediction Module:**
    - **Description:** The prediction module utilizes the trained LSTM model to recognize signs in real-time or from stored video data.
    - **Functionality:** It processes incoming feature vectors from the preprocessing module, running them through the LSTM to generate predictions for the corresponding banking terms. The predictions are then mapped to a human-readable format.
6. **Output Module:**
    - **Description:** This module displays the predicted sign to the user.
    - **Functionality:** It provides real-time feedback by showing the recognized banking term on the user interface. Additionally, it may include mechanisms for visual or auditory signals to confirm the recognized sign.

4.3. Detailed Design of the proposed system

a. Data Flow Diagrams

DFD(Level 0)

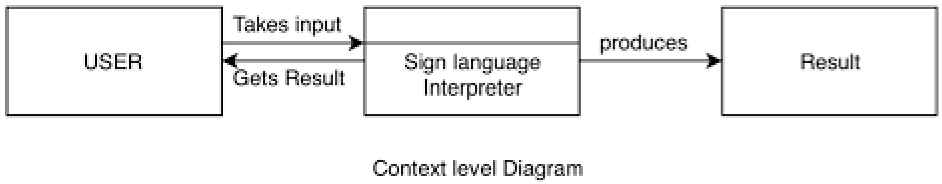


Fig 4.3.a Data Flow Diagram (Level 0)

The context level diagram shows a simple interaction between a user and a sign language detection system. The user provides an input (e.g., a video of a sign language gesture), the system processes it, and the user receives the output (e.g., text or speech representing the recognized gesture).

DFD(Level 1)

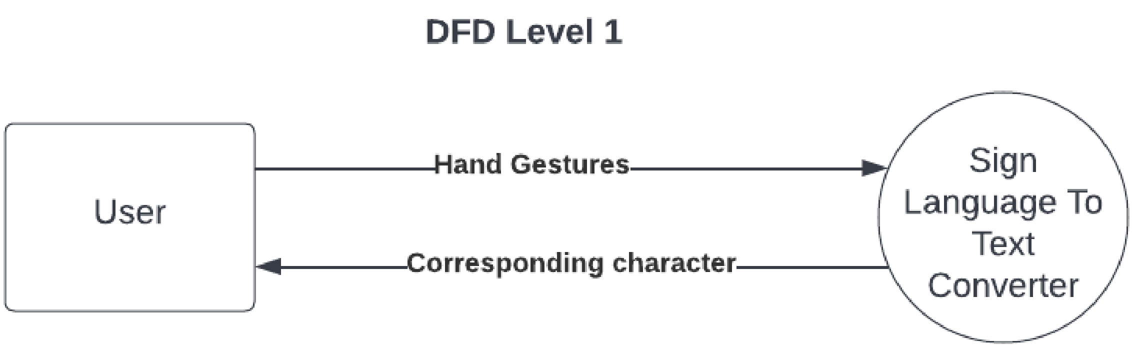


Fig 4.3.b Data Flow Diagram(Level 1).

The DFD Level 1 diagram illustrates the interaction between the user and the sign language detection system. The user provides hand gestures as input to the system, and the system, represented by the "Sign Language To Text Converter," processes these gestures and produces the corresponding text as output. This simplified diagram highlights the core functionality of the system, where the user's input is translated into text.

DFD(Level 2)

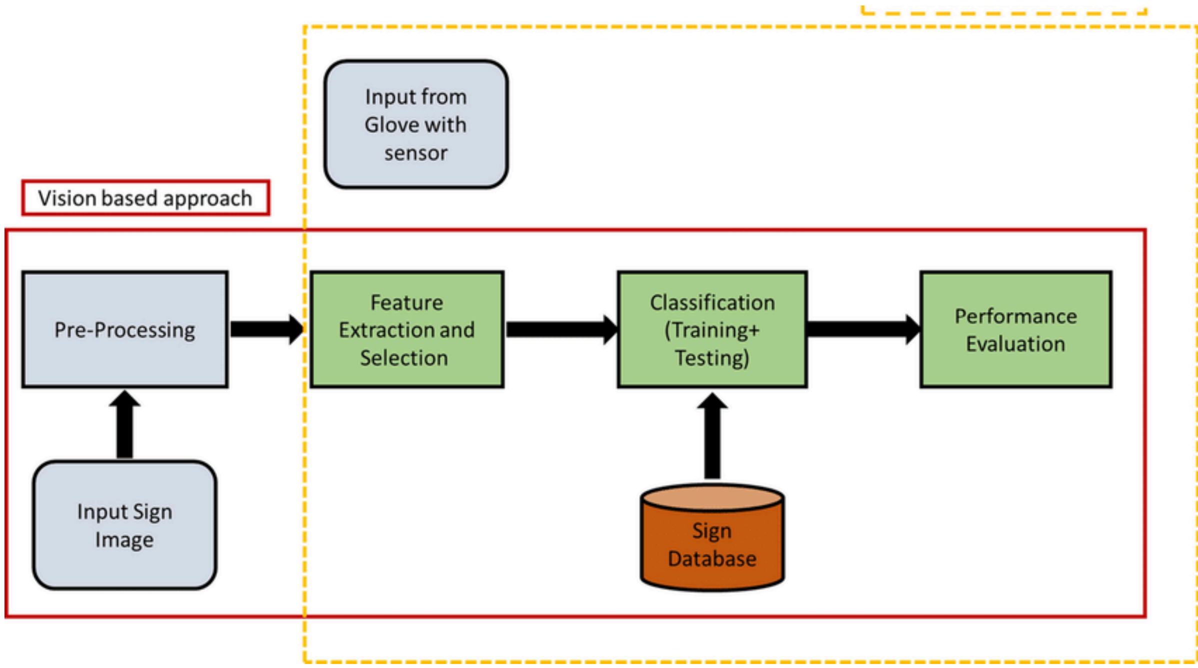


Fig 4.3.c Data Flow Diagram (Level 2).

The DFD Level 2 diagram illustrates the detailed workflow of the sign language recognition system. It shows the two primary approaches for input: direct measurement using a glove with sensors and vision-based approach using input sign images. The system then undergoes pre-processing to prepare the data, followed by feature extraction and selection. The extracted features are used for classification, which involves training and testing the model on a sign database. Finally, the performance of the system is evaluated. This diagram outlines the key steps involved in the sign language recognition process.

b. **Flowchart for the proposed system**

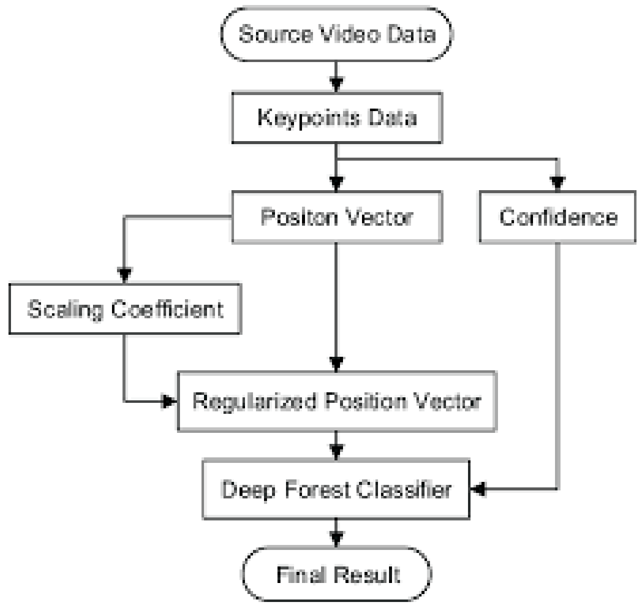


Fig 4.4 Flowchart Of System

The flowchart illustrates the process of sign language recognition using a deep forest classifier. The system starts with source video data, from which keypoints are extracted. These keypoints are then used to calculate a position vector and a confidence score. A scaling coefficient is applied to the position vector to normalize its scale. The resulting regularized position vector is fed into a deep forest classifier, which ultimately produces the final result, likely the classification of the sign language gesture.

c. Activity Diagram

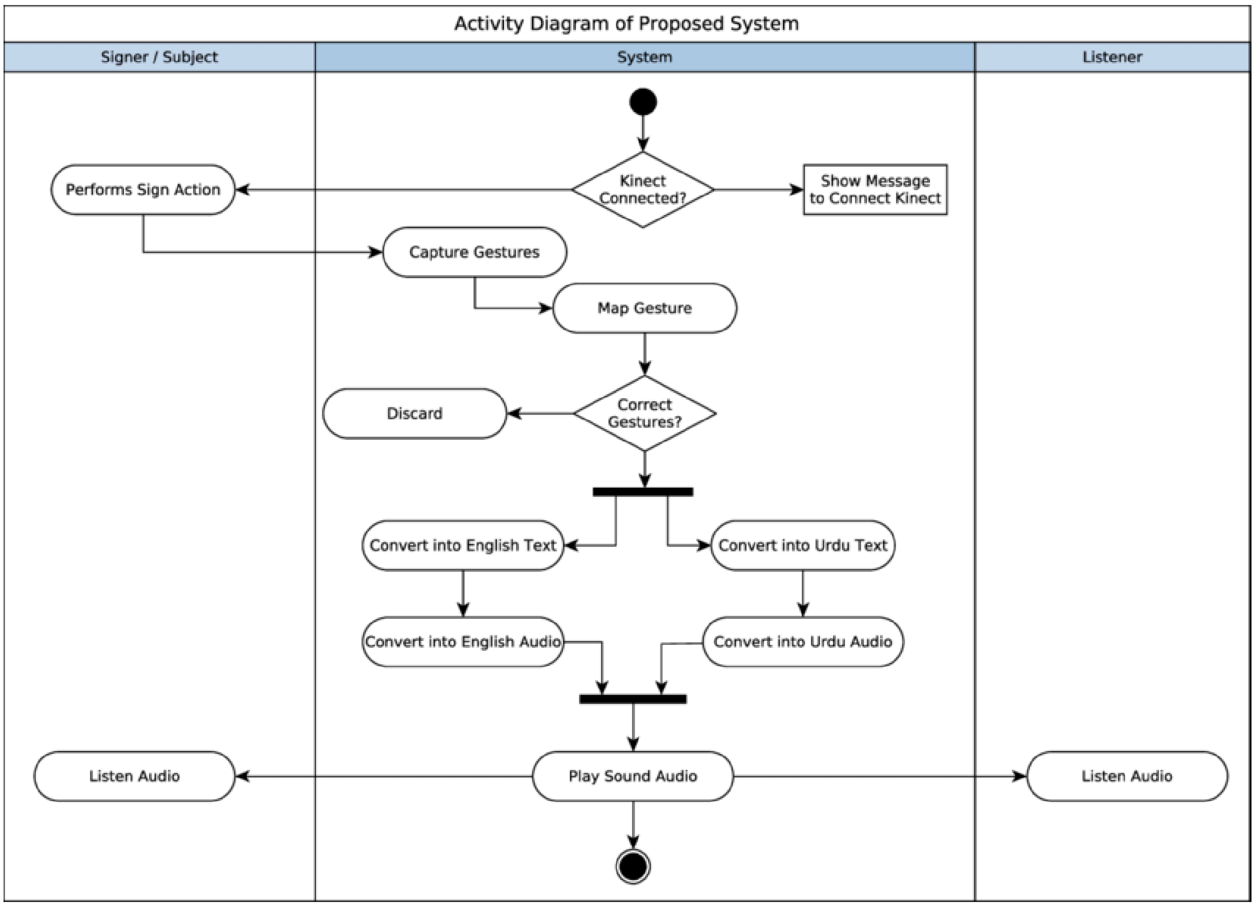


Fig 4.5 Activity Diagram Of the System.

The activity diagram illustrates the workflow of the proposed sign language recognition system. It starts with a signer performing a sign action. The system then checks if the Kinect is connected. If not, it displays a message prompting the user to connect the Kinect. Once connected, the system captures the gestures, maps them to corresponding actions or words, and discards incorrect gestures. The recognized gestures are then converted into either English text or Urdu text, depending on the user's preference. Finally, the text can be converted into audio and played for the listener. The listener can then choose to listen to the audio in either English or Urdu.

4.4 Algorithm utilized in the existing systems

Random Forest:

Random Forest is an ensemble learning algorithm primarily used for classification and regression tasks, which constructs multiple decision trees during training and outputs the mode or average of their predictions. It operates by using bootstrap sampling to create random subsets of the training data for each tree, promoting diversity and reducing overfitting. Additionally, it considers a random subset of features when splitting nodes in each tree, enhancing robustness. The final prediction is made through majority voting for classification or averaging for regression. While Random Forest is powerful and can handle missing values well, it can be complex and computationally intensive, making it less interpretable than a single decision tree. Its versatility makes it applicable in various fields, including finance, healthcare, and marketing.

LSTM (Long Short-Term Memory):

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) designed to effectively learn and predict sequences of data, addressing the vanishing gradient problem that traditional RNNs face. LSTMs use a unique architecture that includes memory cells and three gates: the input gate, the forget gate, and the output gate. These components enable the model to maintain information over long sequences, selectively remember or forget past inputs, and output relevant predictions. This makes LSTMs particularly well-suited for tasks involving time series data, natural language processing, and other sequential data types, allowing them to capture temporal dependencies and context more effectively than standard RNNs.

GRU (Gated Recurrent Unit):

Gated Recurrent Unit (GRU) is a type of recurrent neural network (RNN) that is designed to handle sequence data while simplifying the architecture compared to Long Short-Term Memory (LSTM) networks. GRUs use two main gates: the reset gate and the update gate, which help control the flow of information. The reset gate determines how much of the past information to forget, while the update gate decides how much of the new information to incorporate into the current state. This streamlined structure allows GRUs to capture dependencies in sequential data effectively while reducing computational complexity and training time. As a result, GRUs are widely used in applications such as natural language processing, speech recognition, and time series forecasting, providing a balance between performance and efficiency.

Each of these models has its strengths and weaknesses, and the choice of model depends on the specific requirements of the sign language detection system, such as the size of the dataset, the complexity of the sign language gestures, and the desired level of accuracy.

Chapter 5. RESULTS AND DISCUSSIONS

5.1. Determination of Precision, Recall, Accuracy and F1 Score

Data Splitting:

The dataset was splitted into three different ratios to get accurate results
The dataset was divided into two sets, with an 80% to 20% ratio. 80% of the data was used to train the model, while 20% was used to validate it[5].
The dataset is divided into training and testing sets, with 70% used for training the model and the remaining 30% reserved for testing.
The dataset is divided into training and testing sets, with 75% used for training the model and the remaining 25% reserved for testing. We also applied data normalization to eliminate any NaN (missing) values before training the models.

1. Confusion Matrix

To better understand Precision and Recall we can use a Confusion Matrix which summarizes the performance of a classifier in four essential terms:

- True Positives (TP): Correctly predicted positive instances.
- False Positives (FP): Incorrectly predicted positive instances.
- True Negatives (TN): Correctly predicted negative instances.
- False Negatives (FN): Incorrectly predicted negative instances.

	Predicted Positive	Predicted Negative
Actual Positive	1108	142
Actual Negative	161	1089

Table No: 5.1 Confusion matrix for GRU model with of data splitting 70:30

	Predicted Positive	Predicted Negative
Actual Positive	1195	255
Actual Negative	301	749

Table No: 5.2 Confusion matrix for random LSTM with of data splitting 70:30

	Predicted Positive	**Predicted Negative**
Actual Positive	1007	343
Actual Negative	524	626

Table No: 5.3 Confusion matrix for random forest model with of data splitting 70:30

	Predicted Positive	Predicted Negative
Actual Positive	910	190
Actual Negative	142	1258

Table No: 5.4 Confusion matrix for GRU model with of data splitting 75:25

	Predicted Positive	Predicted Negative
Actual Positive	1136	164
Actual Negative	229	971

Table No: 5.5 Confusion matrix for LSTM model with of data splitting 75:25

	Predicted Positive	Predicted Negative
Actual Positive	1018	382
Actual Negative	532	568

Table No: 5.6 Confusion matrix for random forest model with of data splitting 75:25

	Predicted Positive	Predicted Negative
Actual Positive	423	77
Actual Negative	54	1946

Table No: 5.7 Confusion matrix for random GRU with of data splitting 80:20

	Predicted Positive	Predicted Negative
Actual Positive	833	167
Actual Negative	178	1322

Table No: 5.8 Confusion matrix for LSTM model with of data splitting 80:20

	Predicted Positive	Predicted Negative
Actual Positive	807	393
Actual Negative	348	952

Table No: 5.9 Confusion matrix for Random Forest model with of data splitting 80:20

2. Precision

It refers to the proportion of correct positive predictions (True Positives) out of all the positive predictions made by the model (True Positives + False Positives). It is a measure of the accuracy of the positive predictions.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

A high Precision means that the model makes few False Positives. This metric is especially useful when the cost of false positives is high such as email spam detection.

3. Recall

It is also known as Sensitivity or True Positive Rate where we measure the proportion of actual positive instances that were correctly identified by the model. It is the ratio of True Positives to the total actual positives (True Positives + False Negatives).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

A high Recall means the model correctly identifies most of the positive instances, which is critical when False Negatives are costly, like in medical diagnoses.

4. Accuracy

Accuracy refers to how close a measurement is to the true value. It's about being correct. In physics, accuracy refers to how close a measured value is to the true or accepted value of a physical quantity. For example, if a clock shows the time as 3:00 PM and it is 3:00 PM, the clock is accurate.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy measures how well the test or tool identifies or predicts the correct outcome. For example, if a thermometer reads 100 degrees and the actual temperature is 99.9 degrees, that thermometer is considered accurate.

5. F1 Score

The F1 Score, also known as the F1-measure, is a metric used in classification to provide a single score that balances both the precision and the recall of a model. It is particularly useful when dealing with imbalanced datasets where accuracy alone can be misleading

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F1 score for each sign would represent how well your model is at correctly identifying that specific sign while minimizing both false positives (predicting that sign when it wasn't performed) and false negatives (missing that sign when it was performed).

Model	**GRU**	**LSTM**	**Random Forest**
Dataset Splitting 70:30			
Precision	87.27%	79.87%	65.76%
Recall	88.65%	82.39%	74.57%
Accuracy	87.88%	77.76%	65.32%
F1 Score	87.97%	81.13%	69.91%
Dataset Splitting 75:25			
Precision	86.53%	83.25%	65.67%
Recall	82.69%	87.36%	72.72%
Accuracy	86.72%	84.28%	63.44%
F1 Score	84.56%	85.25%	69.02%

Dataset Splitting 80:20			
Precision	88.5%	82.33%	69.75%
Recall	84.67%	83.33%	67.25%
Accuracy	94.76%	86.25%	70.36%
F1 Score	86.54%	82.82%	68.47%

Table No: 5.10 Precision, recall, accuracy and F1 Score values

Inference of the above table :

- GRU is the top-performing model across all dataset splits, excelling in Precision, Recall, Accuracy, and F1 Score, indicating its strong ability to learn temporal patterns in sign language.
- LSTM performs second best, consistently outperforming Random Forest but generally lower than GRU, suggesting it's also effective for sequential data but slightly less so for this task.
- Random Forest is the weakest performer, likely due to its limitations in handling sequential data inherent in sign language videos.
- Increasing training data (80:20 split) tends to improve accuracy for GRU and LSTM, suggesting more data is beneficial for these models.
- GRU appears to be the most robust and suitable model for this sign language recognition task, warranting further investigation and optimization.

Summary

The GRU model appears to be the most effective, offering the best trade-off between correctly identifying positive instances and minimizing false positives, leading to the highest overall accuracy. The LSTM model is a decent performer, prioritizing the capture of most positive instances but with a higher rate of false positives. The Random Forest model seems to be less suitable for this task compared to the recurrent neural networks.

Sensitivity analysis is a crucial step in evaluating the robustness and reliability of a sign language recognition system. It involves systematically varying the input parameters or conditions to assess how the system's performance changes. This helps identify potential weaknesses and areas for improvement [6].

Sensitivity analysis evaluates the system's robustness to changes in lighting, background, occlusions, sign language variations, noise, and model parameters. By adjusting these factors and evaluating their effects on performance, one can gain insights into the model's strengths, limitations, and potential areas for enhancement.

5.2. Implementation

5.2.1. Random Forest Model Prediction

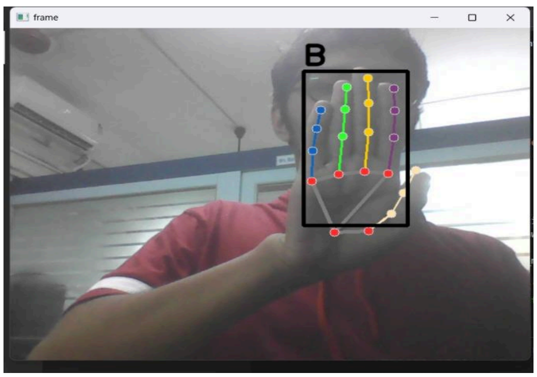


Fig 5.2.1 Predicting alphabet B.

a. **Actual class:** B
   Predicted class: B

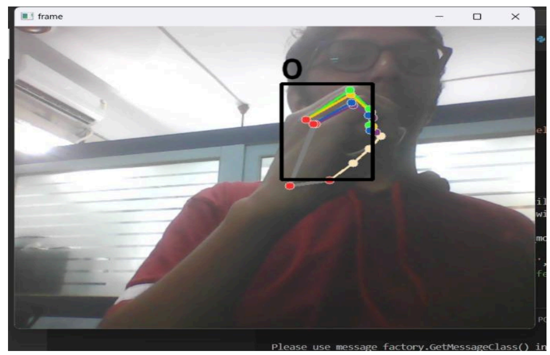


Fig 5.2.2 Predicting alphabet O.

b. **Actual class:** O
   Predicted class: O

5.2.2. LSTM Model predictions of Words

Fig 5.2.3 Word prediction-Top Management

a. Actual class: Top management

Predicted class: Top Management

Fig 5.2.4 Word prediction- Money

a. **Actual class:** Money
   Predicted class: Money

Fig 5.2.5 Word Prediction- Fraud

c. **Actual class:** Fraud
   Predicted class: Fraud

5.2.3. GRU Model Prediction of Words:

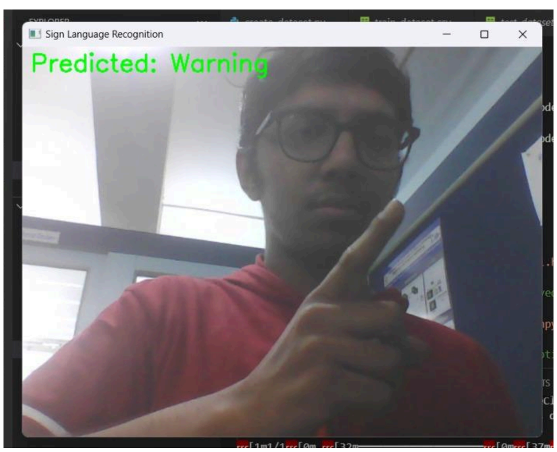


Fig 5.2.6 Word Prediction- Warning

a. **Actual class:** Warning
   Predicted class: Warning

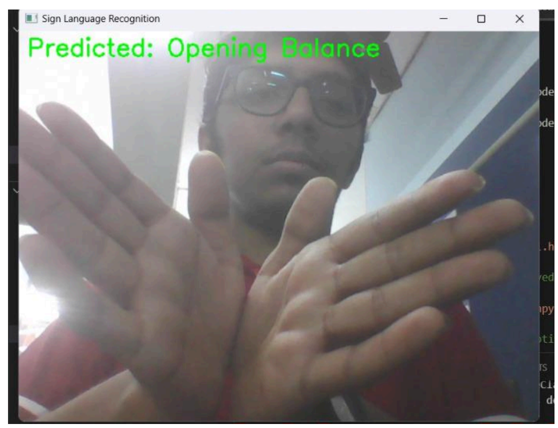


Fig 5.2.7 Word Prediction- Opening Balance

b. **Actual class:** Opening balance
   Predicted class: Opening balance

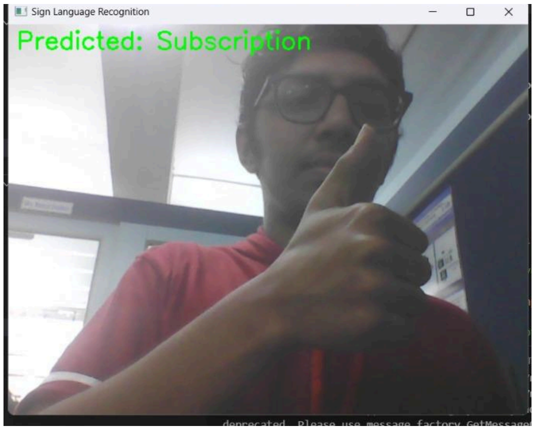


Fig 5.2.9 Word Prediction- Subscription

c. **Actual class:** Subscription
   Predicted class: Subscription

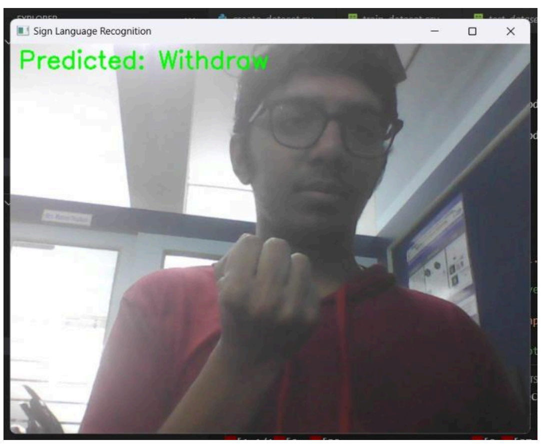


Fig 5.2.8 Word Prediction- Withdraw

d. **Actual class:** Withdraw
   Predicted class: Withdraw

5.2.4. Sentence Prediction by GRU model.

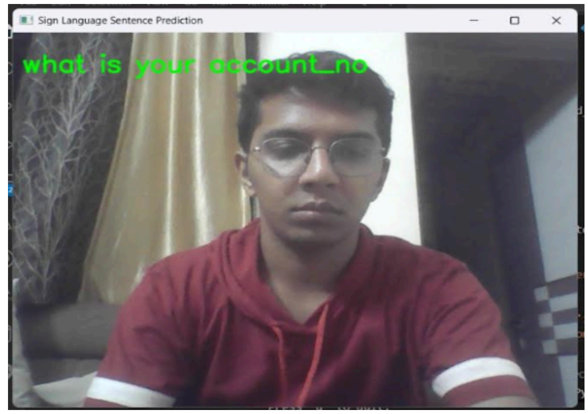


Fig 5.2.10 Prediction for sentence

e. **Actual class:** What is your account no
   Predicted class: What is your account no

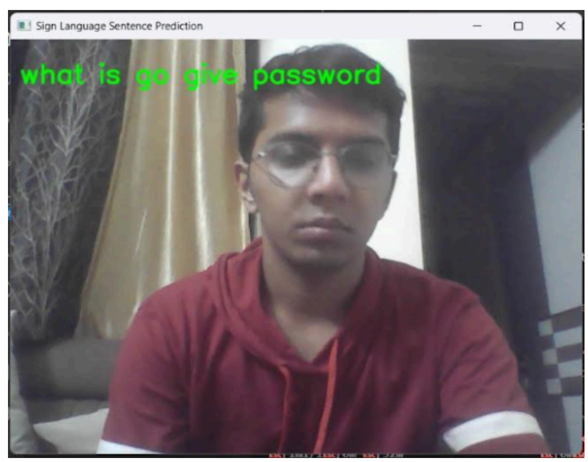


Fig 5.2.11 Prediction for Sentence

f. **Actual class:** What is your password
   Predicted class: What is go give password

Chapter 6. Conclusion

The project successfully utilized advanced machine learning models, including Random Forest, LSTM, and GRU, to develop a sign language detection system specifically focused on the banking sector. Following thorough evaluation, the GRU model was identified as the most effective, demonstrating superior accuracy and efficiency for the intended application.

Future work will expand the system's capabilities to include sentence prediction using the GRU model, which is anticipated to enhance understanding and interpretation of more complex sign language expressions. This development is expected to improve communication and accessibility for users, facilitating more seamless interactions within the banking sector.

Ongoing efforts will focus on refining the models and incorporating user feedback, ensuring that the technology continues to evolve and address the diverse needs of the community. The potential impact of this work highlights significant opportunities for advancing sign language technology across various applications.

Chapter 7. References

1. Kim, S., & Choi, Y. (2020). Real-Time American Sign Language Recognition Using Deep Learning. In Proceedings of the IEEE International Conference on Computer Vision (ICCV).
2. Dong, J., & Zhang, Z. (2021). Hand Gesture Recognition Using Convolutional Neural Networks. In Proceedings of the IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR).
3. Akinola, A., & Zhang, C. (2021). Sign Language Recognition Using Deep Learning Techniques. IEEE Transactions on Neural Networks and Learning Systems.
4. Ahmed, S., & Khalifa, S. (2021). Deep Learning-Based Sign Language Recognition: A Survey. IEEE Access.
5. Malsburg, C., & Haeusser, J. (2021). *Automatic Sign Language Recognition Using Deep Learning*. IEEE Transactions on Image Processing.
6. "Hand Gesture Recognition Using Convolutional Neural Networks" by A. S. Sudhakar et al., 2020
7. "Sign Language Recognition Using Hand Tracking and Deep Learning Techniques" by F. O. G. Abiyev et al., 2021
8. "A Comprehensive Review of Sign Language Recognition Technologies" by M. Elakkiya and M.S. Suresh, 2021
9. "Vision-Based Sign Language Recognition: A Review",Samir Mitra, Tinku Acharya Published In IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007
10. "Sign Language Recognition System Using Hand Gesture and Movement", Anil Kumar Tiwari Rekha S, IEEE Access, 2019
11. "Development of an End-to-End Deep Learning Framework for Sign Language Recognition, Translation, and Video Generation" (2022)
12. "Deep Learning-Based Approach for Sign Language Gesture Recognition With Efficient Hand Gesture Representation" (2020)
13. "Real-Time Sign Language Recognition Using CNN and LSTM Networks" (2021)
14. "Multilingual Sign Language Recognition via CNN and Transfer Learning" (2021)
15. "A CNN-LSTM Framework for Continuous Sign Language Recognition" (2021)

Chapter 8. Appendix

a. Research Paper

Sign Language Recognition for Differently abled

Karan Khatri
Department of Computer Engineering
Vivekanand Education Society Institute of Technology
2021.karan.khatri@ves.ac.in

Manav Daryani
Department of Computer Engineering
Vivekanand Education Society Institute of Technology
2021.manav.daryani@ves.ac.in

Komal lund
Department of Computer Engineering
Vivekanand Education Society Institute of Technology
2021.komal.lund@ves.ac.in

Kunal vishwakarma
Department of Computer Engineering
Vivekanand Education Society Institute of Technology
2021.kunal.vishwakarma@ves.ac.in

Prof.Sanjay mirchandani
Assistant Professor
Department of Computer Engineering
Vivekanand Education Society Institute of Technology
sanjay.mirchandani@ves.ac.in

Abstract

An advanced sign language detection system enhances communication for individuals with hearing impairments by integrating computer vision and deep learning techniques. A dataset of 50 banking-related sign categories, each with 50 videos, is preprocessed to handle noise and variations. Convolutional Neural Networks (CNNs) extract features, while Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) capture temporal dependencies for accurate recognition. Comparative analysis with Random Forest and ID3 shows LSTM and GRU outperform for word and Language detection. Experimental results demonstrate high accuracy, making the system a reliable tool for improving accessibility and social inclusion.

Keywords— Sign Language Recognition, Deep Learning, LSTM, GRU, Computer Vision, NLP, Banking Sector

I. INTRODUCTION

Sign language is a vital form of communication for individuals with hearing impairments. However, the lack of widespread understanding and accessibility of sign language can pose significant challenges in daily life.Communication is essential for people to function as a species. It is a fundamental and effective technique for communicating thoughts, feelings, and points of view. Many people experience either hearing loss, speaking difficulty, or both. Mute is a handicap that prevents speech and renders those who have it mute[1]. To address this issue, the research presents a novel sign language detection system designed to bridge the communication gap between hearing and non-hearing individuals.

The system leverages advanced computer vision techniques to accurately recognize and interpret hand gestures, translating them into corresponding text or speech output. . In SL, many kinds of motions with numerous shapes are used. SL is one of the chief methods of communication among deaf as well as hearing people. SL detection networks are an effective method to talk with deaf and mute people [2]. By providing a reliable and efficient means of communication, the system aims to empower individuals with hearing impairments and foster greater inclusivity in society.

The system's development is driven by the need for a more accessible and user-friendly solution for sign language recognition. Existing approaches often suffer from limitations in terms of accuracy, robustness, and adaptability to different sign language variations. This research seeks to overcome these challenges by incorporating state-of-the-art deep learning architectures and carefully curated datasets. resources, and enhance preventive oncology practices.

II. LITERATURE SURVEY

A literature survey on the project "Sign Language Recognition System for Differently Abled" explores various studies, technologies, methodologies, and advancements in the field of sign language recognition (SLR). The focus is on developing systems to help the differently-abled, particularly those who communicate using sign language, interact with computers, devices, or people more efficiently. Below is a literature survey based on key areas of research in this domain: This paper provides a comprehensive review of various technologies used in sign language recognition. The authors discuss different models, including CNNs, LSTMs, and hybrid models like CNN-RNN, and their applications in sign language recognition.To realize the Sign Language Translation system, we have meticulously designed and implemented various functionalities to ensure seamless communication between users proficient in sign language and those using spoken language[4]. The review also highlights the challenges associated with standardization in datasets and integration of sign language recognition systems into practical applications.

The review offers valuable insights into the diverse range of technologies used in sign language recognition. It provides a comprehensive overview of different models and their strengths and limitations. However, the lack of standardization in datasets and the challenges in integrating sign language recognition systems into practical applications remain significant obstacles to be addressed.

The authors use YOLOv3 for hand tracking and 3D CNNs for sign language classification. Experimental results demonstrate the system's ability to accurately recognize sign language gestures in real-time, despite potential inaccuracies in hand tracking[6].

The combination of hand tracking and deep learning techniques offers a promising approach for sign language recognition. However, the system's performance may be affected by inaccuracies in hand tracking, particularly in challenging real-world conditions. Additionally, processing real-time data can be computationally demanding.

including VGG16, MobileNet, and GRU (Gated Recurrent Units), and discuss their strengths and limitations. The survey highlights the challenges associated with large-scale datasets and real-time performance.

The survey offers valuable insights into the state-of-the-art in deep learning-based sign language recognition. It provides a comprehensive overview of different models and their suitability for various applications. However, the survey's limited coverage of real-time systems and challenges with large-scale datasets may require further exploration.

propose a hybrid model combining Bidirectional Long Short-Term Memory (BiLSTM) networks with Inception V3 to capture both temporal and spatial dependencies in sign language gestures. Experimental results highlight the model's effectiveness in recognizing complex sign language gestures, demonstrating the potential of deep learning for accurate sign language recognition.

The hybrid BiLSTM-Inception V3 model offers a promising approach for sign language recognition, particularly for complex gestures. However, the model may be computationally expensive, limiting its real-time applicability.

RESEARCH GAP

A research gap in a sign language recognition (SLR) system for differently-abled individuals refers to an unexplored or underdeveloped area that requires further investigation. Some common research gaps in this field include:

1. Accuracy and Robustness

Limited real-time performance: Many SLR systems struggle with real-time processing due to computational constraints.

Recognition of complex gestures: Many models fail to accurately recognize subtle hand movements, facial expressions, and finger spelling.

Noise and environmental variations: Background clutter, lighting conditions, and occlusions can impact recognition accuracy.

2. Dataset Limitations

Lack of diverse datasets: Most datasets focus on a limited number of signs or specific languages (e.g., American Sign Language - ASL, British Sign Language - BSL), ignoring regional variations.

Low-resource sign languages: Many sign languages, especially those used in small communities, have minimal datasets available for training AI models.

Multimodal data integration: There is limited work on integrating hand gestures, facial expressions, and body posture for better recognition.

3. Real-time User Adaptability

Personalized sign variations: Different users may have unique signing styles, speeds, or physical limitations that existing systems do not accommodate.

Continuous sign recognition: Many models focus on isolated word recognition rather than continuous sentence-level sign interpretation.

4. Integration with Other Technologies

Speech-to-sign translation: Few systems offer bidirectional translation between sign language and spoken language.

Human-computer interaction: Limited research exists on integrating SLR with virtual assistants, AR/VR, or IoT-based accessibility tools.

Edge computing and mobile applications: Many SLR systems require high computational power and are not optimized for mobile or embedded devices.

5. Accessibility and Deployment

Affordability and hardware constraints: Many recognition systems rely on expensive sensors (e.g., Leap Motion, depth cameras) rather than low-cost alternatives like smartphone cameras.

Cross-platform compatibility: Limited work has been done on developing systems that work across different operating systems and devices.

Proposed System

The proposed system aims to develop an efficient Sign Language Recognition (SLR) model tailored for differently-abled individuals, specifically to facilitate communication in banking environments. The system follows a structured three-phase hierarchical approach, progressing from alphabet recognition to full sentence comprehension. The proposed method utilizes deep learning architectures such as Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Random Forest to capture and classify sign language gestures effectively.

Alphabet Recognition Model

The **Alphabet Recognition Model** is the first phase of the system, aimed at recognizing individual alphabets in sign language. This serves as the foundation for more complex tasks, such as recognizing words and sentences in the banking sector. The model takes **video input of hand gestures**, extracts relevant features, and classifies them into corresponding alphabet labels.

To achieve this, three different machine learning and deep learning models are implemented:

Long Short-Term Memory (LSTM) Model
Long Short-Term Memory (LSTM) is a specialized type of Recurrent Neural Network (RNN) designed to overcome the limitations of traditional RNNs, particularly the vanishing gradient problem. It is highly effective in learning sequential dependencies over long time durations, making it an ideal choice for recognizing dynamic hand gestures in sign language.
Unlike standard RNNs, which struggle to retain information over long sequences, LSTMs incorporate memory cells that enable them to store, update, and recall information selectively over extended periods. This property is crucial for sign language recognition, where gestures unfold sequentially over time, and

previous movements influence the meaning of the current gesture.
An LSTM network consists of multiple layers of LSTM units (cells) that process input sequences over time. Each LSTM unit has an internal memory cell that enables it to retain important information while discarding irrelevant details.
The architecture of an LSTM model can be divided into four main components:

1. Input Layer – Processes input data (e.g., hand landmarks from sign language videos).
2. LSTM Layers – Learn temporal patterns in gesture sequences.
3. Fully Connected Layer (Dense Layer) – Converts LSTM output into meaningful features.
4. Softmax Layer – Predicts the final recognized sign language alphabet.

Working Of LSTM
The LSTM model processes sequential input data step by step, ensuring that past information influences future predictions.

1. Input Processing:
    - The input to the LSTM model is a sequence of hand landmarks extracted from sign language videos.
    - These sequences are time-series data, where each frame contains x, y coordinates of key hand points.
2. LSTM Layer:
    - Each LSTM cell processes a single time step of the input sequence and passes information to the next time step.
    - The LSTM cell maintains an internal memory (cell state) that helps retain relevant information.
3. Forget, Input, and Output Mechanisms:
    - The LSTM model selectively forgets unnecessary information and stores important information in its cell state.
    - It then outputs relevant information to the next time step.
4. Final Prediction:
    - After processing the full sequence, the final LSTM output is passed through a fully connected layer.
    - The Softmax layer then classifies the input into one of the sign language alphabets.

Alphabet recognition LSTM model screenshots

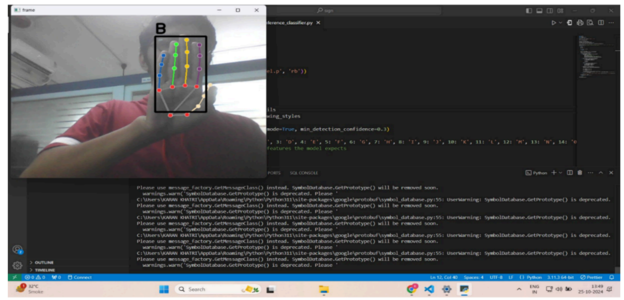


Fig 5.2.2 Model Prediction.

a. **Actual class:** B
Predicted class: B

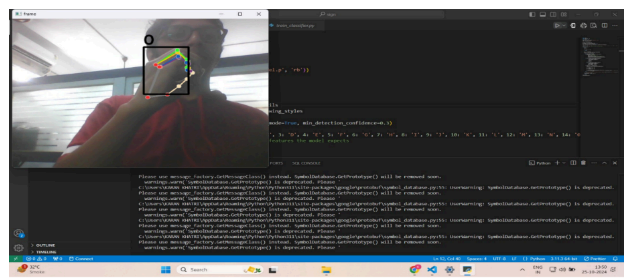


Fig 5.2.3 Model predicting alphabets

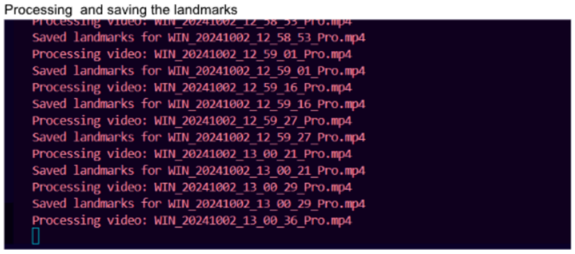
b. **Actual class:** O
Predicted class: O

Banking Letter Recognition Model :
The Banking Letter Recognition Model is designed to identify letters associated with banking terminology in sign language. Recognizing individual banking-related letters is crucial for enabling smooth communication between differently-abled individuals and banking personnel.
This phase builds upon alphabet recognition, extending the model's capability to letters frequently used in banking scenarios. Since banking communication requires a high level of accuracy and real-time processing, we use LSTM and GRU models due to their ability to learn sequential patterns effectively.

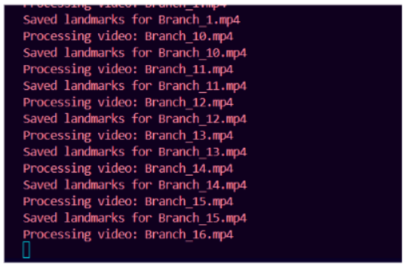
LSTM model

1. Preprocessing the dataset and saving the landmarks.

Processing and saving the landmarks



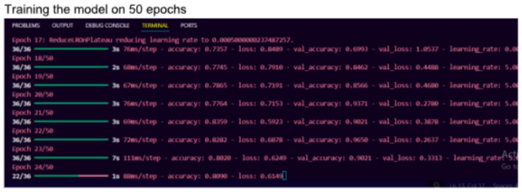
```
Saved landmarks for WIN_20241002_12_58_53_Pro.mp4
Processing video: WIN_20241002_12_59_01_Pro.mp4
Saved landmarks for WIN_20241002_12_59_01_Pro.mp4
Processing video: WIN_20241002_12_59_16_Pro.mp4
Saved landmarks for WIN_20241002_12_59_16_Pro.mp4
Processing video: WIN_20241002_12_59_27_Pro.mp4
Saved landmarks for WIN_20241002_12_59_27_Pro.mp4
Processing video: WIN_20241002_13_00_21_Pro.mp4
Saved landmarks for WIN_20241002_13_00_21_Pro.mp4
Processing video: WIN_20241002_13_00_29_Pro.mp4
Saved landmarks for WIN_20241002_13_00_29_Pro.mp4
Processing video: WIN_20241002_13_00_36_Pro.mp4
```



```
Saved landmarks for Branch_1.mp4
Processing video: Branch_10.mp4
Saved landmarks for Branch_10.mp4
Processing video: Branch_11.mp4
Saved landmarks for Branch_11.mp4
Processing video: Branch_12.mp4
Saved landmarks for Branch_12.mp4
Processing video: Branch_13.mp4
Saved landmarks for Branch_13.mp4
Processing video: Branch_14.mp4
Saved landmarks for Branch_14.mp4
Processing video: Branch_15.mp4
Saved landmarks for Branch_15.mp4
Processing video: Branch_16.mp4
```

2.Training the model.

Training the model on 50 epochs



```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS
Epoch 17: ReduceLROnPlateau reducing learning rate to 0.0005000000237487257.
36/36 ━━━━━━━━━━━━━━━━━━━━ 3s 76ms/step - accuracy: 0.7357 - loss: 0.8489 - val_accuracy: 0.6993 - val_loss: 1.0537 - learning_rate: 0.0
Epoch 18/50
36/36 ━━━━━━━━━━━━━━━━━━━━ 2s 68ms/step - accuracy: 0.7745 - loss: 0.7910 - val_accuracy: 0.8462 - val_loss: 0.4488 - learning_rate: 5.0
Epoch 19/50
36/36 ━━━━━━━━━━━━━━━━━━━━ 3s 67ms/step - accuracy: 0.7865 - loss: 0.7191 - val_accuracy: 0.8566 - val_loss: 0.4680 - learning_rate: 5.0
Epoch 20/50
36/36 ━━━━━━━━━━━━━━━━━━━━ 3s 76ms/step - accuracy: 0.7764 - loss: 0.7153 - val_accuracy: 0.9371 - val_loss: 0.2780 - learning_rate: 5.0
Epoch 21/50
36/36 ━━━━━━━━━━━━━━━━━━━━ 3s 69ms/step - accuracy: 0.8359 - loss: 0.5923 - val_accuracy: 0.9021 - val_loss: 0.3878 - learning_rate: 5.0
Epoch 22/50
36/36 ━━━━━━━━━━━━━━━━━━━━ 3s 72ms/step - accuracy: 0.8282 - loss: 0.6078 - val_accuracy: 0.9650 - val_loss: 0.2637 - learning_rate: 5.0
Epoch 23/50
36/36 ━━━━━━━━━━━━━━━━━━━━ 7s 111ms/step - accuracy: 0.8020 - loss: 0.6249 - val_accuracy: 0.9021 - val_loss: 0.3313 - learning_rate: 5.0
Epoch 24/50
22/36 ━━━━━━━━━━━━━━━━━━━━ 1s 88ms/step - accuracy: 0.8090 - loss: 0.6149
```

Gestures Prediction by LSTM model.

a. **Actual class:** Top management
   Predicted class: Top Management

b. **Actual class:** Money
   Predicted class: Money

Gated Recurrent Unit (GRU) Model

The Gated Recurrent Unit (GRU) model is a specialized type of recurrent neural network (RNN) designed to efficiently capture sequential dependencies in time-series data. For banking-related letters in sign language, GRU provides a computationally efficient alternative to LSTM, ensuring fast and accurate recognition of gestures.
This section explores the architecture, working mechanism, training process, and real-time deployment of the GRU model for banking letter recognition.
The GRU-based banking letter recognition model consists of the following key components:

1. Input Layer: Takes hand landmark sequences extracted from sign videos.

2. GRU Layers: Learn sequential dependencies from hand movement patterns.

3. Fully Connected (Dense) Layer: Converts GRU outputs into meaningful feature representations.

4. Softmax Layer: Classifies the sequence into a banking-related letter.

GRU improves upon traditional RNNs by introducing gates that control information flow. Unlike LSTMs, which use three gates (forget, input, and output), GRU simplifies this with just two gates:

1. Update Gate (z) – Determines how much past information should be retained and how much new information should be added.

2. Reset Gate (r) – Controls how much past information is forgotten, allowing flexibility in learning dependencies.

Step-by-Step Working of GRU

1. Input Sequence Processing: The model takes landmark-based sequences from sign videos as

input.

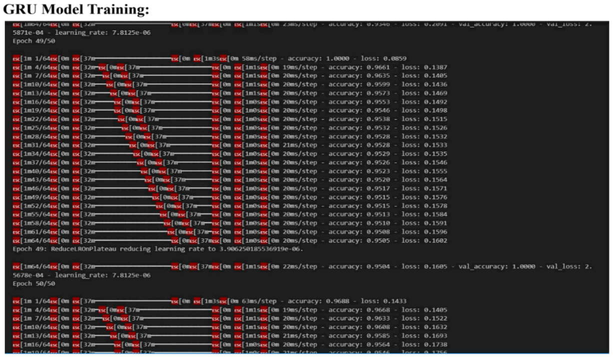
2. Update Gate Controls Memory: It decides whether the new hand movement should replace old memory or be combined with it.

3. Reset Gate Filters Past Information: It determines how much past data to forget when processing new frames.

4. New Hidden State Calculation: The model updates its internal state based on the update and reset gates' decisions.

5. Final Prediction: The processed sequence passes through a fully connected layer and a softmax classifier, which assigns a probability distribution over banking-related letters.

GRU Model Training:



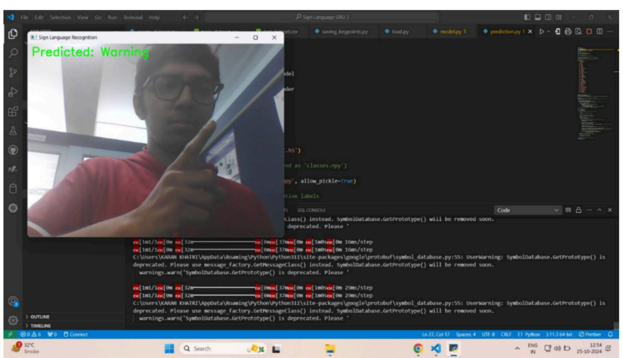


Fig 5.2.12 Prediction for Warning Gesture

a. **Actual class:** Warning
Predicted class: Warning

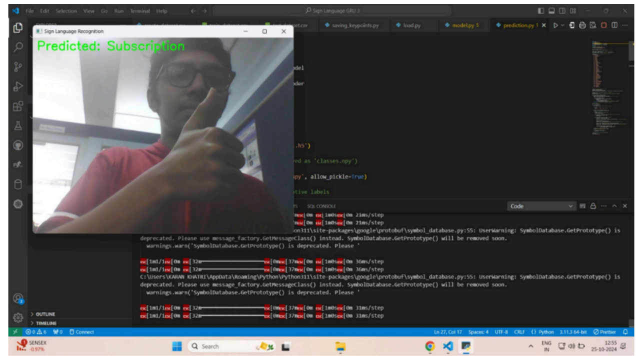
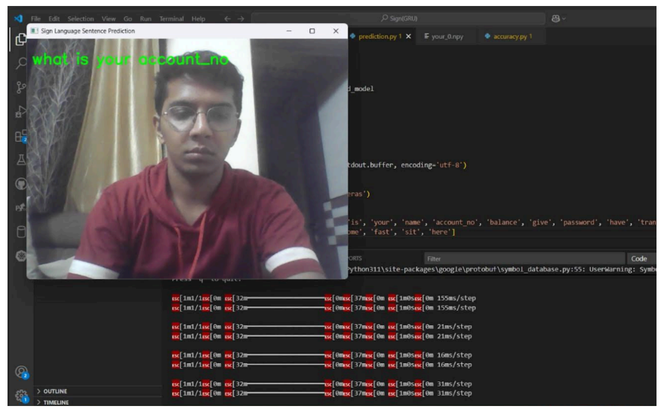
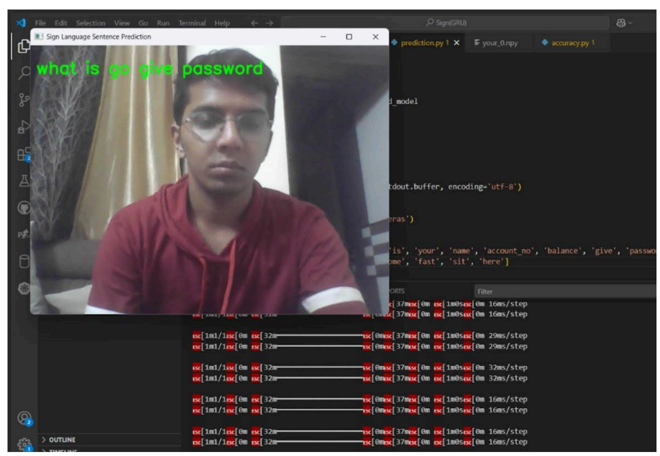


Fig 5.2.14 Prediction for Subscription Gesture

c. **Actual class:** Subscription
Predicted class: Subscription





METHODOLOGY

Convolutional Neural Networks (CNNs): For static gestures like the alphabet or common banking-related hand signs.

Recurrent Neural Networks (RNNs) and LSTMs: For recognizing continuous sign language gestures and sequences of hand movements, which are important for dynamic conversation.

3D-CNNs: For recognizing gestures from video data, capturing both spatial and temporal patterns (particularly important when analyzing gestures that occur over time).

In the banking domain, the application of Sign Language Recognition Systems (SLRS) is designed to improve accessibility and communication for differently-abled individuals, particularly those who are deaf or hard of hearing. These systems are focused on facilitating seamless and inclusive interactions between customers and banking services. The methodology for implementing SLRS in banking involves adapting general SLRS techniques to meet the specific requirements of banking applications, such as customer support, transaction verification, customer identification, and financial advice.

The dataset includes the following 14 parameters, which serve as input features for the prediction models. These parameters are:

- **Deposit:** A gesture involving two hands (one holding an imaginary object, the other "placing" it into the first).
- **Transfer:** A gesture signifying the movement of money between two parties.
- **Loan:** A gesture signifying a loan or financial assistance request.
- **Withdraw Gesture:** A common gesture might be forming a gesture with the fingers or simulating the action of pulling something from an account, associated with withdrawing funds**.**
- **Balance**: To check the account balance if needed.

Data Splitting:

The dataset was divided into two sets, with an 80% to 20% ratio. 80% of the data was used to train the model, while 20% was used to validate it[5]. The dataset is divided into training and testing sets, with 75% used for training the model and the remaining 25% reserved for testing. We also applied data normalization to eliminate any NaN (missing) values before training the models.

Algorithms Used:

Random Forest:

Random Forest is an ensemble learning algorithm primarily used for classification and regression tasks, which constructs multiple decision trees during training and outputs the mode or average of their predictions. It operates by using bootstrap sampling to create random subsets of the training data for each tree, promoting diversity and reducing overfitting. Additionally, it considers a random subset of features when splitting nodes in each tree, enhancing robustness. The final prediction is made through majority voting for classification or averaging for regression. While Random Forest is powerful and can handle missing values well, it can be complex and computationally intensive, making it less interpretable than a single decision tree. Its versatility makes it applicable in various fields, including finance, healthcare, and marketing.

LSTM (Long Short-Term Memory):

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) designed to effectively learn and predict sequences of data, addressing the vanishing gradient problem that traditional RNNs face. LSTMs use a unique architecture that includes memory cells and three gates: the input gate, the forget gate, and the output gate. These components enable the model to maintain information over long sequences, selectively remember or forget past inputs, and output relevant predictions. This makes LSTMs particularly well-suited for tasks involving time series data, natural language processing, and other sequential data types, allowing them to capture temporal dependencies and context more effectively than standard RNNs.

GRU (Gated Recurrent Unit):

Gated Recurrent Unit (GRU) is a type of recurrent neural network (RNN) that is designed to handle sequence data while simplifying the architecture compared to Long Short-Term Memory (LSTM) networks. GRUs use two main gates: the reset gate and the update gate, which help control the flow of information. The reset gate determines how much of the past information to forget, while the update gate decides how much of the new information to incorporate into the current state. This streamlined structure allows GRUs to capture dependencies in sequential data effectively while reducing computational complexity and training time. As a result, GRUs are widely used in applications such as natural language processing, speech recognition, and time series forecasting, providing a balance between performance and efficiency.

Each of these models has its strengths and weaknesses, and the choice of model depends on the specific requirements of the sign language detection system, such as the size of the dataset, the complexity of the sign language gestures, and the desired level of accuracy.

Decision Tree:

A decision tree is a supervised learning algorithm commonly used for classification tasks, capable of handling both numerical and categorical data. It follows a tree-like structure, with internal nodes representing decisions based on specific features, branches reflecting feature values, and leaf nodes containing the predicted class.This algorithm needs to be explained because it performed best compared to the ML algorithms in the results section; this decision algorithm and its possible results are presented as a tree. [7] The decision-making process starts from the root node and progresses through branches until a leaf node is reached, which provides the classification result. Decision trees are particularly useful in disease prediction due to their simplicity and interpretability.

Precision, Recall and Accuracy:

Model	**Precision**	**Recall**	**Accuracy**
LSTM	82.2%	83.3%	82.2%
GRU	88.5%	84.67%	86.33%
Random Forest	69.75 %	67.25%	68.5%

III. CONCLUSION

The project successfully utilized advanced machine learning models, including Random Forest, LSTM, and GRU, to develop a sign language detection system specifically focused on the banking sector. Following thorough evaluation, the GRU model was identified as the most effective, demonstrating superior accuracy and efficiency for the intended application.

Future work will expand the system's capabilities to include sentence prediction using the GRU model, which is anticipated to enhance understanding and interpretation of more complex sign language expressions. This development is expected to improve communication and accessibility for users, facilitating more seamless interactions within the banking sector.

Ongoing efforts will focus on refining the models and incorporating user feedback, ensuring that the technology continues to evolve and address the diverse needs of the community. The potential impact of this work highlights significant opportunities for advancing sign language technology across various applications.

IV. FUTURE WORK

Improving and evolving Sign Language Recognition Systems (SLRS) for the differently-abled involves addressing existing limitations and incorporating emerging technologies. Here are some future changes that can be made to enhance the functionality, accessibility, and usability of such systems:

1.Cross-Lingual Sign Language Translation

Support for Multiple Sign Languages: Different regions use different sign languages (e.g., American Sign Language (ASL), British Sign Language (BSL), Indian Sign Language (ISL)). Building systems that can recognize and translate between various sign languages is a promising direction.

Real-Time Translation Between Sign Languages: This would allow people using different sign languages to communicate with one another seamlessly, breaking down language barriers in the deaf community.

2. Real-Time Recognition and Low-Latency Systems

Edge Computing: To make sign language recognition systems more responsive, future systems can leverage edge computing, processing data locally (on the device) rather than relying on cloud servers. This would minimize latency, making real-time communication more fluid.

Optimized Hardware: Hardware optimizations, such as incorporating specialized chips for AI (e.g., GPUs, TPUs), could help speed up the processing of complex deep learning models, enabling faster recognition.

V. REFERENCES

[1].Kim, S., & Choi, Y. (2020). Real-Time American Sign Language Recognition Using Deep Learning. In Proceedings of the IEEE International Conference on Computer Vision (ICCV).

[2].Dong, J., & Zhang, Z. (2021). Hand Gesture Recognition Using Convolutional Neural Networks. In Proceedings of the IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR).

[3].Akinola, A., & Zhang, C. (2021). Sign Language Recognition Using Deep Learning Techniques. IEEE Transactions on Neural Networks and Learning Systems.

[4].Ahmed, S., & Khalifa, S. (2021). Deep Learning-Based Sign Language Recognition: A Survey. IEEE Access.

[5].Malsburg, C., & Haeusser, J. (2021). Automatic Sign Language Recognition Using Deep Learning. IEEE Transactions on Image Processing.

[6].Hand Gesture Recognition Using Convolutional Neural Networks by A. S. Sudhakar et al., 2020

[7].Sign Language Recognition Using Hand Tracking and Deep Learning Techniques by F. O. G. Abiyev et al., 2021

[8].A Comprehensive Review of Sign Language Recognition Technologies by M. Elakkiya and M.S. Suresh, 2021

[9].Vision-Based Sign Language Recognition: A Review,Samir Mitra, Tinku Acharya Published. In IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007

[10].Sign Language Recognition System Using Hand Gesture and Movement, Anil Kumar Tiwari, Rekha S, IEEE Access, 2019

[11].Development of an End-to-End Deep Learning Framework for Sign Language Recognition, Translation, and Video Generation (2022)

[12].Deep Learning-Based Approach for Sign Language Gesture Recognition With Efficient Hand Gesture Representation (2020)

[13].Real-Time Sign Language Recognition Using CNN and LSTM Networks (2021)

[14].Multilingual Sign Language Recognition via CNN and Transfer Learning (2021)

[15].A CNN-LSTM Framework for Continuous Sign Language Recognition (2021)

b. Project Review Sheets

Industry / Inhouse:
Research / Innovation:

Project Evaluation Sheet 2024-25(Sem 8) **Class: D17A/B/C**

Title of Project (Group no): Sign language Recognition System for Differently Abled

Mentor Name & Group Members: Karan Khatri (D17B, 23), Komal Lund (D17A, 38), Manav Daryani (D17A, 11), Kunal Viswakarma (D17B, 64)

	Engineering Concepts & Knowledge	Interpretation of Problem & Analysis	Design / Prototype	Interpretation of Data & Dataset	Modern Tool Usage	Societal Benefit, Safety Consideration	Environment Friendly	Ethics	Team work	Presentation Skills	Applied Engg & Mgmt principles	Life - long learning	Professional Skills	Innovative Approach	Total Marks
	(5)	(5)	(5)	(3)	(5)	(2)	(2)	(2)	(2)	(3)	(3)	(3)	(5)	(5)	(50)
Review of Project Stage 1	5	3	3	1	4	2	2	2	2	2	2	3	4	3	39

Comments: · Parameters in tabular format.

Name & Signature Reviewer1

	Engineering Concepts & Knowledge	Interpretation of Problem & Analysis	Design / Prototype	Interpretation of Data & Dataset	Modern Tool Usage	Societal Benefit, Safety Consideration	Environment Friendly	Ethics	Team work	Presentation Skills	Applied Engg & Mgmt principles	Life - long learning	Professional Skills	Innovative Approach	Total Marks
	(5)	(5)	(5)	(3)	(5)	(2)	(2)	(2)	(2)	(3)	(3)	(3)	(5)	(5)	(50)
Review of Project Stage 1	5	3	3	1	4	2	2	2	2	2	2	3	4	3	39

Comments: Dataset to be augmented — GRU, Random Forest, LSTM
More no. of epochs to be included

Date: 01/03/2025

Name & Signature Reviewer2

Inhouse/ Industry_Innovation/Research:

Sustainable Goal:

Project Evaluation Sheet 2024 - 25

Class: D17 A/B/C

Group No.: 39

Title of Project: Sign Language Recognition for Differently Abled

Group Members: Karan Khatri (D17B-23), Komal lund (D17A-38), Manav Daryani (D17A-11), Kunal Vishwakarma (D17B-64)

Engineering Concepts & Knowledge	Interpretation of Problem & Analysis	Design / Prototype	Interpretation of Data & Dataset	Modern Tool Usage	Societal Benefit, Safety Consideration	Environment Friendly	Ethics	Team work	Presentation Skills	Applied Engg&Mgmt principles	Life - long learning	Professional Skills	Innovative Approach	Research Paper	Total Marks
(5)	(5)	(5)	(3)	(5)	(2)	(2)	(2)	(2)	(2)	(3)	(3)	(3)	(3)	(5)	(50)
5	3	3	2	5	2	2	2	2	2	3	3	3	2	3	42

Comments:

Name & Signature Reviewer1

Engineering Concepts & Knowledge	Interpretation of Problem & Analysis	Design / Prototype	Interpretation of Data & Dataset	Modern Tool Usage	Societal Benefit, Safety Consideration	Environment Friendly	Ethics	Team work	Presentation Skills	Applied Engg&Mgmt principles	Life - long learning	Professional Skills	Innovative Approach	Research Paper	Total Marks
(5)	(5)	(5)	(3)	(5)	(2)	(2)	(2)	(2)	(2)	(3)	(3)	(3)	(3)	(5)	(50)
5	3	3	2	5	2	2	2	2	2	3	3	3	2	3	42

Comments:

Date: 1st April,2025

Sanjay M.

Name & Signature Reviewer 2