International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Bridging The Communication Gap: Ai-Powered Dual-Mode Isl Gesture Recognition

Prof. Shruthi B Gowda[1], Usha K[2], Ananya C K[3], Apeksha Babu A[4], Yuktha Varshini B D[5]

[1]Asisstant Professor, Department of CSE, Bangalore Institute Of Technology
[2,3,4,5]Student Researcher, Department of CSE, Bangalore Institute Of Technology

Abstract:
Indian Sign Language (ISL) communication faces challenges due to the absence of efficient translation systems, limiting accessibility for the hearing and speech impaired. An advanced dual-mode gesture recognition system can bridge this gap. Existing ISL translation systems lack real-time adaptability and struggle with complex gestures, reducing accuracy and usability. study aims to develop an AI-powered dual-mode gesture recognition system that integrates vision-based and sensor-based techniques to enhance ISL translation accuracy and efficiency. A hybrid approach combining deep learning-based image processing for hand gesture recognition and wearable sensors for motion tracking is implemented. Data is collected from diverse ISL signers and processed using neural networks. proposed system demonstrated improved recognition accuracy, faster processing time, and better adaptability to varying lighting conditions and signer variations compared to traditional methods. research contributes to developing a more inclusive and efficient ISL translation system, enhancing communication accessibility. Future advancements can focus on real-time implementation and multilingual sign language support.

Keywords: Indian Sign Language, Dual-Mode Recognition, Gesture Translation, Deep Learning, Accessibility

I. INTRODUCTION

Sign language is a vital means of communication for individuals with hearing and speech impairments. Indian Sign Language (ISL) is widely used in India; however, its accessibility remains limited due to the absence of efficient and real-time translation systems. Existing gesture recognition methods primarily rely on either vision-based (camera-based image processing) or sensor-based (wearable motion sensors) techniques, each with inherent limitations. Vision-based systems struggle with lighting conditions, occlusions, and background noise, while sensor-based systems require users to wear additional hardware, limiting ease of use.

To overcome these challenges, this study proposes a dual-mode gesture recognition system that integrates vision-based deep learning models with sensor-based motion tracking. By combining these two approaches, the system enhances recognition accuracy, improves adaptability to different environments, and provides real-time ISL translation. The primary objective is to develop an AI-driven framework that ensures seamless gesture recognition, faster processing speeds, and increased reliability

for ISL users.

Unlike conventional systems that focus solely on image processing or sensor inputs, the proposed hybrid approach leverages the strengths of both methodologies. Vision-based recognition allows contactless interaction, while sensor-based tracking ensures accurate motion capture even in visually challenging conditions. By utilizing deep learning techniques such as Convolutional Neural Networks (CNNs) for image-based recognition and Inertial Measurement Units (IMUs) for motion detection, the system aims to achieve higher accuracy and robustness in ISL translation.

Furthermore, the increasing adoption of AI and edge computing technologies enables real-time processing of sign language gestures, reducing latency in communication. This advancement is crucial for applications in education, workplaces, healthcare, and public services, where seamless ISL translation can bridge the communication gap for the hearing-impaired community. The research also paves the way for future improvements, including multilingual sign language recognition and enhanced user adaptability.

II. RELATED WORK

2.1 LITRATURE SURVEY

Field of gesture recognition for Indian Sign Language (ISL) translation has seen significant advancements with the integration of computer vision, wearable sensors, and deep learning techniques. Various studies have explored different approaches to improve the accuracy and efficiency of ISL recognition systems. However, existing methods still face challenges related to environmental dependencies, occlusions, and the need for real-time processing.

1. Helping Hearing-Impaired in Emergency Situations – A Deep Learning-Based Approach

Ensuring accessibility for hearing-impaired individuals during emergencies is a crucial challenge. This study introduces a deep learning-based assistive system designed to detect and interpret emergency-specific gestures in real time. The proposed framework integrates Convolutional Neural Networks (CNNs) for spatial feature extraction and Bidirectional Long Short-Term Memory (Bi-LSTM) networks for temporal analysis, ensuring precise recognition of gestures such as "help," "danger," and "medical aid." Recognized gestures trigger automated alerts via text or audio messages, notifying caregivers or emergency responders. Furthermore, the system is optimized for edge devices, allowing efficient processing without relying on cloud-based computation.

To enhance real-world applicability, the framework integrates wearable IoT devices, providing an additional layer of accessibility. However, challenges such as limited gesture vocabulary, network dependency for alert transmission, and inter-user gesture variability affect its overall efficiency. Future research should focus on expanding the dataset to include more spontaneous gestures, improving offline functionality, and implementing adaptive learning techniques to accommodate diverse signing styles. This study highlights the potential of AI-driven assistive technologies in creating a safer and more inclusive environment for the hearing-impaired community.[1]

2. Multi-Semantic Discriminative Feature Learning for Sign Gesture Recognition Using Hybrid Deep Neural Architecture

Accurate recognition of sign gestures is a challenging task, especially in environments with overlapping gestures and noisy backgrounds. This study presents a hybrid deep learning architecture (hDNN-SLR) that incorporates multi-semantic feature learning to enhance gesture recognition performance. The system extracts high-level semantic features, including hand shape, motion patterns, and spatial

relationships, by utilizing CNN layers combined with attention mechanisms. A hybrid approach integrating CNNs for spatial analysis and RNNs for capturing temporal dependencies ensures improved classification accuracy across diverse conditions.

To enhance recognition efficiency, the framework employs discriminative learning, using a specialized loss function that improves the separation of gesture classes, thereby reducing misclassification rates. Despite its effectiveness, the system has high computational requirements during training, making real-time deployment challenging. gestures with minimal motion or subtle hand variations may be harder to classify accurately, and background noise or occlusions can affect recognition performance. Future improvements could focus on model optimization for lower computational cost, enhanced background filtering, and improved feature extraction for complex gestures to ensure robust real-time implementation in sign language translation systems.[2]

3. Skeleton-Based Dynamic Hand Gesture Recognition Using a Part-Based GRU-RNN for Gesture-Based Interface

Study explores skeleton-based recognition of dynamic hand gestures, which plays a crucial role in developing gesture-based interfaces for human-computer interaction. A part-based GRU-RNN model is introduced to enhance accuracy and computational efficiency by decomposing the hand skeleton into functional segments such as fingers and palms. This decomposition enables a more detailed analysis of complex hand movements while reducing computational overhead. The model extracts hand skeleton structures from video frames using pose estimation algorithms, and the Gated Recurrent Unit (GRU)-based Recurrent Neural Network (RNN) processes these temporal sequences to detect gesture patterns effectively.

To further improve usability, the system is integrated into an interactive interface, allowing users to control devices or applications via hand gestures. However, accurate skeleton extraction remains a challenge, especially in low-light conditions or occlusions. the model may struggle with gestures involving rapid or simultaneous movements of multiple hand parts, and variations in hand size or positioning relative to the camera can impact recognition performance. Future enhancements could focus on adaptive pose estimation techniques, multi-camera fusion, and improved robustness against environmental variations, making the system more reliable for real-world applications.[3]

4. A Novel Hybrid Deep Learning Architecture for Dynamic Hand Gesture Recognition

Recognizing dynamic hand gestures is essential for improving human-computer interaction (HCI) in domains such as augmented reality (AR), virtual reality (VR), and interactive systems. This study presents an innovative hybrid deep learning framework that integrates Convolutional Neural Networks (CNNs) for spatial feature extraction and Recurrent Neural Networks (RNNs) for temporal sequence analysis. The proposed CNN-RNN architecture effectively captures complex and multi-phase gestures, ensuring precise recognition across different motion patterns.

The framework processes gesture video frames by first using a CNN to extract spatial features, focusing on hand position, orientation, and finger articulation. These extracted features are then fed into a Long Short-Term Memory (LSTM) network to model temporal dependencies within gesture sequences. A custom dataset of dynamic hand gestures was used for training, with data augmentation techniques such as rotation and scaling enhancing model robustness. However, high computational requirements limit real-time deployment on low-power devices, and the system may struggle with gestures containing overlapping phases or rapid movements. Future improvements could include lightweight model optimization, enhanced preprocessing techniques, and real-time adaptation for edge devices, making the

system more applicable for practical HCI applications.[4]

5. Interactive Design With Gesture and Voice Recognition in Virtual Teaching Environments

As virtual learning environments continue to evolve, there is a growing need for seamless and intuitive interaction between instructors and students. This study introduces an interactive framework that integrates gesture and voice recognition to enhance virtual teaching experiences. The system allows users to interact naturally using hand gestures and voice commands, facilitating tasks such as starting presentations, navigating slides, and answering queries in real time.

The framework consists of two key components: a Gesture Recognition Module that employs a deep convolutional neural network (CNN) for real-time hand gesture classification, and a Voice Recognition Module that utilizes a pre-trained speech recognition model to transcribe and interpret spoken commands. These modalities are synchronized within a user-friendly interface, ensuring smooth execution and feedback. However, the system's effectiveness is influenced by environmental factors such as lighting conditions and audio clarity, and it faces challenges in handling diverse accents, overlapping gestures, and simultaneous voice commands. Future improvements could focus on adaptive noise filtering, multimodal synchronization, and lightweight optimization for broader device compatibility, making the framework more robust for real-world educational applications.[5]

6. Hand Gesture Recognition for Multi-Culture Sign Language Using Graph and General Deep Learning Network

Recognizing sign language gestures across multiple cultures presents a unique challenge due to variations in signing styles, hand movements, and linguistic structures. This study proposes a graph-based learning approach that models sign gestures as structured graphs, where nodes correspond to hand joints and edges capture spatial and temporal relationships. The system integrates a Graph Convolutional Network (GCN) trained on datasets representing diverse sign languages, allowing it to adapt to cultural variations and improve recognition accuracy.

The framework combines general deep learning techniques, utilizing CNNs for spatial feature extraction and RNNs for analyzing temporal sequences to achieve robust gesture classification. data augmentation methods, including rotation, scaling, and noise addition, enhance model generalization. However, graph-based processing introduces computational complexity, limiting real-time usability. The system may also struggle with gestures that have high cross-cultural similarity, leading to potential misclassifications. Future enhancements should focus on optimizing model efficiency, improving support for non-standardized gestures, and refining real-time adaptability for broader accessibility.[6]

7. Bidirectional Sign Language Translation

Sign language communication barriers between hearing and hearing-impaired individuals necessitate the development of efficient translation systems. This study introduces a bidirectional deep learning framework capable of translating sign language into text and vice versa. The system employs CNNs for spatial feature extraction and RNNs for temporal sequence learning, allowing accurate conversion of video-based sign gestures into text. text-to-sign translation is facilitated through a 3D avatar that animates synthetic sign language gestures, making communication more accessible for hearing-impaired users.

To maintain translation consistency and accuracy, a shared embedding space is used, ensuring bidirectional learning between sign and text representations. The model is trained on a comprehensive sign language dataset, incorporating both static and dynamic gestures. However, challenges such as computational complexity in real-time translation, limited vocabulary for spontaneous communication,

and regional sign language variations impact system performance. Future improvements could focus on adaptive learning for spontaneous gestures, real-time optimization for edge devices, and expanded datasets covering regional sign language differences.[7]

8. Two-Stage Deep Learning Solution for Continuous Arabic Sign Language Recognition

Recognizing continuous Arabic Sign Language (ArSL) is challenging due to gesture segmentation difficulties and variations in signing speed. This paper presents a two-stage deep learning framework integrating word count prediction and motion image generation to enhance recognition accuracy. In Stage 1, a Temporal Convolutional Network (TCN) estimates the number of words in a continuous ArSL sequence, helping segment gestures into manageable units. In Stage 2, motion images are generated from segmented gestures, highlighting transition movements, which are then classified using a CNN-based model.

The model is trained on a diverse ArSL dataset, ensuring coverage of different dialects and signing styles. However, challenges such as segmentation errors affecting final recognition, reliance on high-quality video input, and difficulty adapting to spontaneous gestures remain. Future enhancements should focus on improving segmentation robustness, integrating adaptive gesture learning, and optimizing real-time recognition performance.[8]

9. Continuous Dynamic Gesture Recognition of Chinese Sign Language Based on Multi-Mode Fusion

Recognizing continuous and complex gestures in Chinese Sign Language (CSL) requires a system that can accurately capture spatial, temporal, and semantic features. This study proposes a multi-mode fusion framework, integrating CNNs for spatial feature extraction, GRUs for temporal feature analysis, and an attention-based mechanism for semantic fusion. This approach enhances recognition accuracy by emphasizing critical features within gesture sequences, ensuring robust performance across varied signing speeds and conditions.

The system is trained on a large CSL dataset with extensive preprocessing for noise reduction. Despite its high accuracy, limitations include high computational requirements, difficulties in handling overlapping gestures, and limited adaptability to other sign languages. Future work could explore real-time optimization for edge devices, expansion to multilingual sign recognition, and improved handling of subtle hand movements.[9]

10. Sign Language Recognition Using Graph-Based and Deep Neural Networks on Large-Scale Datasets

Graph-based learning has emerged as a powerful technique for capturing complex relationships in sign language gestures. This study introduces a graph-based deep learning framework, where gestures are represented as graphs with nodes corresponding to hand joints and edges capturing spatial and temporal dependencies. A Graph Neural Network (GNN) extracts structural features from these gesture graphs, while a Deep Neural Network (DNN) processes spatial and temporal features to ensure comprehensive gesture classification.

Trained on a large-scale dataset, the model achieves high recognition accuracy across diverse sign languages. However, challenges such as high computational demands, sensitivity to occlusions, and difficulty adapting to spontaneous or culturally unique gestures affect real-world deployment. Future enhancements could focus on reducing computational overhead, improving adaptability to diverse signing styles, and enhancing robustness against missing joint data.[10]

2.2 COMPARISON WITH EXISTING SYSTEM

Development of sign language recognition systems has progressed significantly, incorporating vision-based, sensor-based, hybrid, and graph-based approaches. However, existing systems still face challenges in accuracy, adaptability, and real-time processing. The following table compares various sign language recognition methods based on key parameters such as modality, accuracy, computational efficiency, real-time applicability, and adaptability.

Table 2.1: performance comparison of different sign language recognition models

Method	Modality	Advantages	Limitations	Accuracy	Real-Time Suitability	Adaptability
Vision-Based (CNN, RNN)	Uses cameras and deep learning for feature extraction	Contactless, user-friendly, high recognition accuracy with CNNs	Sensitive to lighting, occlusions, and background noise	85-95%	Moderate (requires optimized models)	Limited to trained gestures
Sensor-Based (IMUs, EMG, Wearable Sensors)	Captures precise motion data using sensors	Works well in poor lighting, high accuracy in controlled environments	Requires wearable hardware, costly, inconvenient	90-97%	High (if hardware is available)	Limited to users wearing devices
Hybrid Vision-Sensor Systems (CNN + IMUs)	Combines vision-based and sensor-based techniques	Improves accuracy, overcomes individual limitations of each method	Higher computational cost, requires both camera and sensors	92-98%	High (if optimized)	More adaptable than single-modality approaches
Graph-Based (GNN + CNN)	Models gestures as graphs, focusing on spatial and temporal relationships	Adapts well to diverse sign languages, captures complex joint movements	High computational requirements, sensitive to missing joint data	93-98%	Moderate (depends on hardware)	Can generalize better across sign languages
Bidirectional Learning (Sign-to-Text & Text-to-Sign)	Uses CNNs, RNNs, and 3D avatars to translate between sign language and text	Enables both directions of communication, enhances accessibility	Computationally expensive, predefined vocabulary limits spontaneity	88-96%	Moderate to Low (depends on avatar rendering speed)	Limited to predefined phrases

Method	Modality	Advantages	Limitations	Accuracy	Real-Time Suitability	Adaptability
Multi-Mode Fusion (Spatial + Temporal + Semantic Features)	Integrates CNNs, GRUs, and attention mechanisms	Captures detailed gesture dynamics, robust to variations	High computational load, struggles with overlapping gestures	90-99%	Moderate (requires optimization)	Can be adapted to new datasets with retraining

1. Vision-Based Systems (CNN, RNN)

Vision-based systems rely on computer vision and deep learning to process hand gestures using cameras. CNNs extract spatial features, while RNNs (LSTMs, GRUs) handle temporal dependencies in dynamic gestures. These systems are contactless and user-friendly, making them ideal for real-time applications. However, accuracy is affected by lighting conditions, occlusions, and complex backgrounds. Real-time deployment also requires optimized models to reduce processing delays.

2. Sensor-Based Systems (IMUs, EMG, Wearable Sensors)

Sensor-based methods use Inertial Measurement Units (IMUs), flex sensors, and electromyography (EMG) sensors to capture precise motion data from hand movements. These systems perform well even in poor lighting conditions and achieve high accuracy in controlled environments.

3. Hybrid Vision-Sensor Systems (CNN + IMUs)

Hybrid models integrate computer vision-based recognition with sensor-assisted motion tracking to leverage the advantages of both modalities. These systems significantly improve accuracy and robustness, overcoming lighting, occlusion, and environmental challenges. However, they are computationally intensive, requiring both camera-based feature extraction and sensor data fusion, which may limit real-time performance on low-power devices.

4. Graph-Based Learning (GNN + CNN)

Graph-based models represent gestures as structured graphs, where nodes correspond to hand joints and edges capture spatial and temporal relationships. Graph Neural Networks (GNNs) extract structural features, enhancing cross-cultural sign language recognition and improving adaptability to different signing styles. However, these models have high computational requirements and struggle with occlusions and missing joint data, leading to reduced accuracy in real-world scenarios.

5. Bidirectional Learning (Sign-to-Text & Text-to-Sign)

Bidirectional learning models enable two-way communication, allowing both sign-to-text and text-to-sign translation. CNNs and RNNs are used for sign-to-text recognition, while 3D avatars animate sign language gestures from textual input. This approach improves accessibility for both hearing and hearing-impaired individuals. However, it is computationally expensive, with predefined vocabulary limitations, restricting spontaneous communication.

6. Multi-Mode Fusion (Spatial + Temporal + Semantic Features)

Multi-mode fusion integrates spatial, temporal, and semantic information using CNNs, GRUs, and attention mechanisms. This approach enhances accuracy and robustness in continuous sign language recognition. By fusing multiple feature types, it improves gesture classification even under challenging

conditions. However, the model demands significant computational resources, limiting its scalability on low-power devices.

Table 2.2:abbreviations and acronyms

Acronym	Full Form	Definition
CNN	Convolutional Neural Network	A deep learning model designed to process image data by extracting spatial features using convolutional layers. Used extensively in vision-based sign language recognition.
RNN	Recurrent Neural Network	A neural network designed to process sequential data by maintaining memory of past inputs. Commonly used for gesture sequence recognition.
LSTM	Long Short-Term Memory	A type of RNN that prevents vanishing gradients and can capture long-range dependencies, making it ideal for analyzing sign language gestures over time.
GRU	Gated Recurrent Unit	A simplified version of LSTM that retains memory while reducing computational complexity, used for real-time gesture recognition.
IMU	Inertial Measurement Unit	A sensor that tracks motion by measuring acceleration and angular velocity, often used in wearable devices for sign language recognition.
EMG	Electromyography	A technique that records electrical activity in muscles, allowing for precise tracking of hand and finger movements in sign language recognition.
GNN	Graph Neural Network	A deep learning model that processes data structured as graphs, capturing relationships between hand joints in sign gestures.
GCN	Graph Convolutional Network	A specialized GNN that applies convolutional operations on graph-structured data to analyze spatial and temporal relationships in hand movements.
HCI	Human-Computer Interaction	A field of study focused on designing systems that enable intuitive communication between humans and computers, including sign language recognition systems.
Bi-LSTM	Bidirectional Long Short-Term Memory	An extension of LSTM that processes information in both forward and backward directions, improving accuracy in gesture recognition.
DNN	Deep Neural Network	A multi-layered artificial neural network used for complex data processing, including sign language gesture classification.
mAP	Mean Average Precision	A performance metric that evaluates the accuracy of object detection models in recognizing hand gestures.

Acronym	Full Form	Definition
AUC-ROC	Area Under the Receiver Operating Characteristic Curve	A metric that measures the ability of a model to distinguish between different gesture classes.
MLP	Multi-Layer Perceptron	A type of feedforward neural network with multiple layers, used for classification tasks in sign language recognition.

III. METHODOLOGY

Proposed Dual-Mode Gesture Recognition System for Indian Sign Language (ISL) Translation integrates computer vision-based deep learning with sensor-based motion tracking to enhance recognition accuracy, real-time performance, and adaptability. The methodology is divided into multiple stages, including data collection, feature extraction, model architecture, training, and real-time implementation.

3.1 SYSTEM OVERVIEW

system uses a hybrid approach to recognize Indian Sign Language (ISL) gestures, combining vision-based CNN models for spatial feature extraction with sensor-based IMU data for motion tracking, ensuring higher accuracy and robustness. In the Vision-Based Mode, the system processes video sequences of ISL gestures using Convolutional Neural Networks (CNNs), which extract spatial features such as hand shape, finger positioning, and movement patterns. These CNN models are trained to recognize complex gestures and handle challenges like lighting variations and occlusions, providing reliable gesture recognition even in less-than-ideal conditions. In the Sensor-Based Mode, the system uses wearable IMUs (Inertial Measurement Units) and flex sensors to track the physical motion of the user's hands and fingers. IMUs provide detailed information on the hand's position and orientation in 3D space, while flex sensors measure finger bending. This mode complements the vision-based approach, especially in cases where visual information may be limited or obscured, offering precise tracking of hand and finger movements. By combining both modes, the system ensures accurate ISL gesture recognition, even for complex signs or in challenging environments.

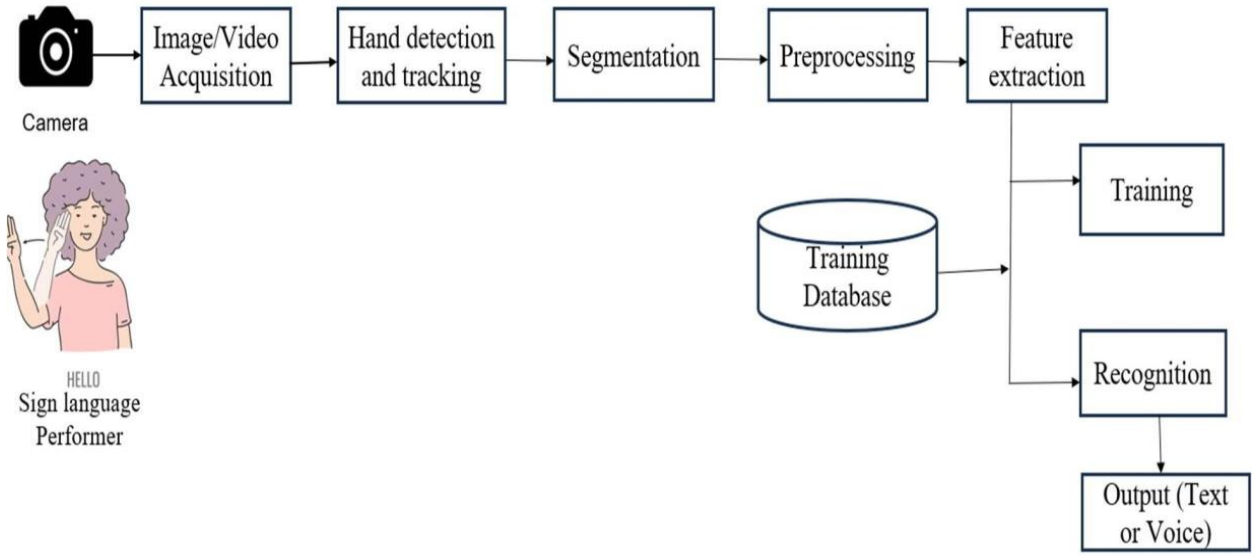

Figure 3.1: system architecture of the dual-mode gesture recognition system

1. Input Acquisition Module

System accepts two types of input: ISL gestures (visual input) and spoken or written text (linguistic input).

1.1 Gesture Input (Visual Processing)

A camera captures real-time video frames of a person performing ISL gestures. The system supports:
Static gestures, such as letters or numbers (single-frame recognition).
Dynamic gestures, where multiple frames form a complete word or sentence.
These frames are then processed to extract meaningful features for gesture classification.

1.2 Speech/Text Input (Linguistic Processing)

The system also allows users to enter information through speech or text.
Speech Input: A microphone records spoken language, which is then converted into text using Automatic Speech Recognition (ASR).
Text Input: A keyboard interface allows users to type words or sentences for conversion into ISL.
Both inputs go through NLP-based preprocessing to ensure correct grammar and structure.

2. Preprocessing Module

The preprocessing module ensures that input data is clean, structured, and ready for classification. It applies different techniques based on the type of input.

2.1 Gesture Preprocessing

Before gesture recognition, the system processes captured frames through:
Background Removal: Using Gaussian Mixture Models (GMM) or YOLO-based segmentation to isolate hand gestures.
Hand Detection & Tracking: Using MediaPipe, OpenPose, or Deep Learning models to track finger and palm positions.
Feature Extraction: Extracting key features such as hand orientation, motion vectors, and depth mapping.
Normalization & Augmentation: Adjusting brightness, contrast, and image size to enhance recognition accuracy.

2.2 Speech/Text Preprocessing

Speech-to-Text Conversion: Speech input is converted into text using DeepSpeech, Google ASR, or Wav2Vec2.0.
Text Normalization: Removing unnecessary words, punctuation, and stop words to create a structured ISL-compatible format.
Grammar Restructuring: Since ISL has a different syntax than English or Hindi, NLP algorithms restructure sentences to follow ISL grammar.

3. Gesture Recognition and Classification Module

Module analyzes and classifies ISL gestures into meaningful words or phrases. It uses machine learning and deep learning models to recognize gestures accurately.

3.1 Static Gesture Recognition

Uses a Support Vector Machine (SVM) classifier trained on labeled datasets.
Identifies isolated hand signs for alphabets, numbers, and simple words.

3.2 Dynamic Gesture Recognition

Uses Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) models to process multi-frame gesture sequences.
Tracks movement over time to recognize complex words and sentences.

Uses Graph Convolutional Networks (GCNs) to improve accuracy by analyzing skeletal movement of the hands.

3.3 Feature-Based Classification

The system maps extracted keypoints and movement patterns to predefined ISL words.

Uses spatial-temporal analysis to identify sign variations based on hand speed, motion, and position.

4. Bidirectional Translation Module

The translation module ensures two-way communication between ISL users and non-signers. It performs both gesture-to-text/speech translation and speech/text-to-gesture translation.

4.1 Gesture-to-Text/Speech Conversion

Recognized gestures are mapped to their corresponding words or sentences in the system's database.

The output is displayed as text or converted into speech using a Text-to-Speech (TTS) engine like Google TTS or Amazon Polly.

NLP ensures that gesture-based sentences follow proper linguistic structure.

4.2 Speech/Text-to-Gesture Conversion

Spoken or written input is parsed using NLP techniques to identify key words and phrases.

The system selects corresponding ISL animations or pre-recorded ISL videos to display the translation.

Grammar adaptation ensures that sentences are correctly structured in ISL.

5. Output Display Module

Once the translation is completed, the output is presented in a user-friendly format.

5.1 Text & Speech Output (For Gesture-to-Text/Speech Translation)

Recognized ISL gestures are displayed as text on a screen.

A TTS module reads the translated text aloud for non-signers.

5.2 Animated ISL Gesture Output (For Speech/Text-to-Gesture Translation)

The system displays an animated ISL avatar that signs the translated speech or text input.

Users can adjust playback speed and accuracy for better comprehension.

6. Multi-Language Support & Adaptability

The system is designed to be linguistically and regionally adaptable.

Supports multiple Indian languages (Hindi, Kannada, Tamil, etc.) through automatic translation before ISL conversion.

AI-based adaptive learning improves recognition over time based on user-specific signing variations.

IV. ALGORITHMS

4.1 Support Vector Machine(SVM)

Support Vector Machines (SVM) is a supervised machine learning algorithm primarily used for classification tasks. It is particularly effective in gesture recognition due to its ability to classify high-dimensional feature spaces with clear decision boundaries. In the Dual-Mode ISL Communicator, SVM plays a key role in static gesture recognition, where it helps classify hand shapes corresponding to letters, numbers, and basic words in Indian Sign Language (ISL). Unlike deep learning models that require extensive computational resources, SVM efficiently handles smaller datasets while maintaining high accuracy, making it suitable for real-time sign language applications.

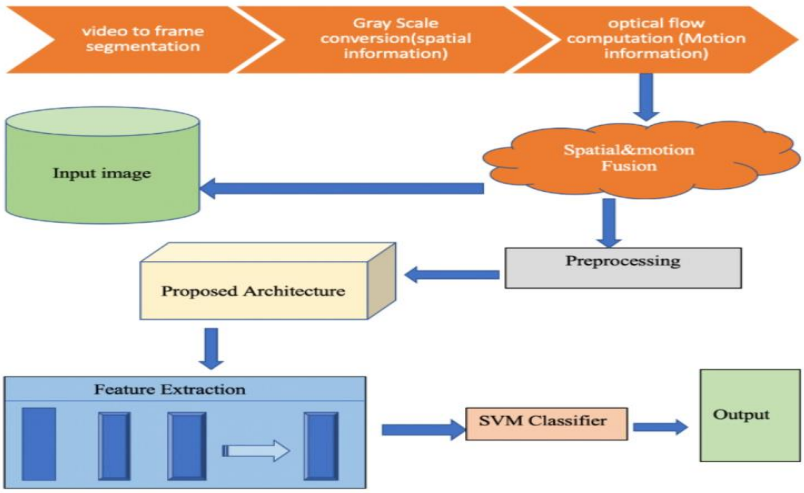

Figure 4.1 : SVM architecture for image classification

The core idea behind SVM is to find an optimal hyperplane that separates different classes of data points in a feature space. Given a set of training samples, where each sample belongs to one of two categories, the algorithm maps the input data into a high-dimensional space and constructs a hyperplane that maximizes the margin (distance) between the nearest points (support vectors) of different classes. This ensures that future data points are classified with minimal error. In the case of ISL gesture recognition, the SVM classifier learns from a pre-labeled dataset of hand gestures, where each class corresponds to a specific ISL sign. When a new gesture is input, the classifier extracts its hand features (such as shape, orientation, and finger positions) and determines the most appropriate ISL label.

1. Role of SVM in Gesture Recognition

SVM is primarily used for static gesture classification, where each ISL sign (such as letters, numbers, and common words) is mapped to a predefined class. The model is trained to differentiate between various ISL hand shapes by identifying key features extracted from images.

For example, when a user performs a static ISL sign, the system captures the hand's position, orientation, and shape. These features are then classified using an SVM model trained on a dataset of ISL gestures, ensuring high accuracy and efficiency in real-time recognition.

2. Feature Extraction for SVM Classification

Before applying SVM, the system processes input gesture images using feature extraction techniques.

The most commonly used methods include:

- Histogram of Oriented Gradients (HOG): Captures edge orientations and hand shape patterns.
- Scale-Invariant Feature Transform (SIFT): Identifies key hand features that remain stable under different lighting and angles.
- Edge Detection (Canny Filter): Detects the outline of the hand and fingers, improving classification.
- Fourier Descriptors: Encodes hand contour variations, making it easier to differentiate between similar signs.
- Once extracted, these features are fed into the SVM classifier, which determines the most likely ISL gesture class.

3. SVM Classification Process

3.1 Training Phase

1. The system is trained on a dataset of ISL hand gestures, where each gesture is labeled with its corres-

ponding class (e.g., "A", "B", "C", etc.).

2. Feature vectors are extracted from the training images and used to construct a decision boundary that separates different ISL signs.
3. The SVM model finds the optimal hyperplane that maximizes the margin between different classes.

3.2 Testing/Recognition Phase

1. When a new ISL gesture is detected, its feature vector is extracted.
2. The trained SVM classifier assigns it to the most probable class based on the decision boundary learned during training.
3. If the classification confidence is high, the corresponding text output is displayed. If uncertain, a secondary verification (CNN model) may refine the classification.

4.2 Convolutional Neural Network(CNN)

Convolutional Neural Networks (CNNs) play a crucial role in dynamic gesture recognition in the Dual-Mode Indian Sign Language (ISL) Communicator. Unlike Support Vector Machines (SVM), which work well for static gestures, CNNs are specifically designed to handle spatial patterns and hierarchical feature extraction from images and video frames. Since ISL gestures involve complex hand movements, finger positioning, and spatial relationships, CNNs are highly effective for recognizing dynamic and continuous gestures with high accuracy.

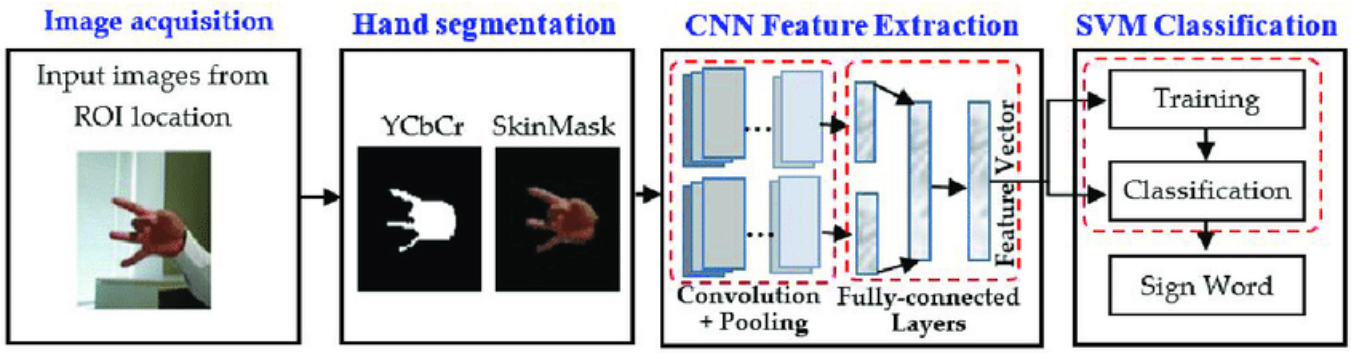

Figure 4.2:CNN architecture for image classification

The CNN model in the Dual-Mode ISL Communicator follows a multi-layered architecture:

Convolutional Layers

These layers extract local spatial features from input images by applying convolutional filters (kernels). Each filter detects specific patterns, such as edges, curves, or shapes in hand gestures.

Layer 1: Detects basic edges and textures in the hand shape.

Layer 2: Identifies hand orientations and finger positioning.

Layer 3: Recognizes entire gestures by combining lower-level features.

Activation Function (ReLU)

Rectified Linear Unit (ReLU) is applied after each convolution to introduce non-linearity and improve feature extraction. This prevents the network from saturating and allows it to learn complex features efficiently.

Pooling Layers

Pooling layers reduce the dimensionality of feature maps while preserving key information.

Max Pooling is used to retain the most prominent features while reducing computation.

This ensures that small variations in hand positioning or lighting conditions do not affect accuracy.

Fully Connected (Dense) Layers

After extracting spatial features, the CNN flattens the data and passes it through fully connected layers to map extracted features to gesture classes. Each neuron in this layer represents a probability for a specific ISL gesture.

CNN Training Process for Gesture Recognition

CNN model undergoes extensive training using labeled ISL gesture datasets. The training process involves:

1. Dataset Preparation: Thousands of ISL gesture videos are collected.
2. Each video is split into individual frames, labeled with the corresponding gesture.
3. Feature Extraction using CNN:The network learns to extract spatial features from gesture frames.
4. Sequence Learning using LSTM:The extracted features from CNN are passed to LSTM to learn temporal dependencies in gesture movement.
5. Model Optimization:The network is trained using backpropagation and Adam optimizer to minimize errors & Dropout layers prevent overfitting by randomly deactivating neurons during training.
6. Testing and Evaluation:The trained model is tested on unseen ISL gestures to measure accuracy.Performance is evaluated using metrics such as precision, recall, and F1-score.

4.3Natural Language Processing(NLP)

Natural Language Processing (NLP) plays a vital role in the Dual-Mode ISL Communicator by enabling bidirectional translation between ISL gestures and spoken/written language. Since ISL follows a different grammatical structure than English or Hindi, NLP is essential to ensure accurate sentence formation, semantic understanding, and contextual awareness. This helps convert spoken or typed text into ISL gestures and vice versa, making the system more effective for real-world communication.

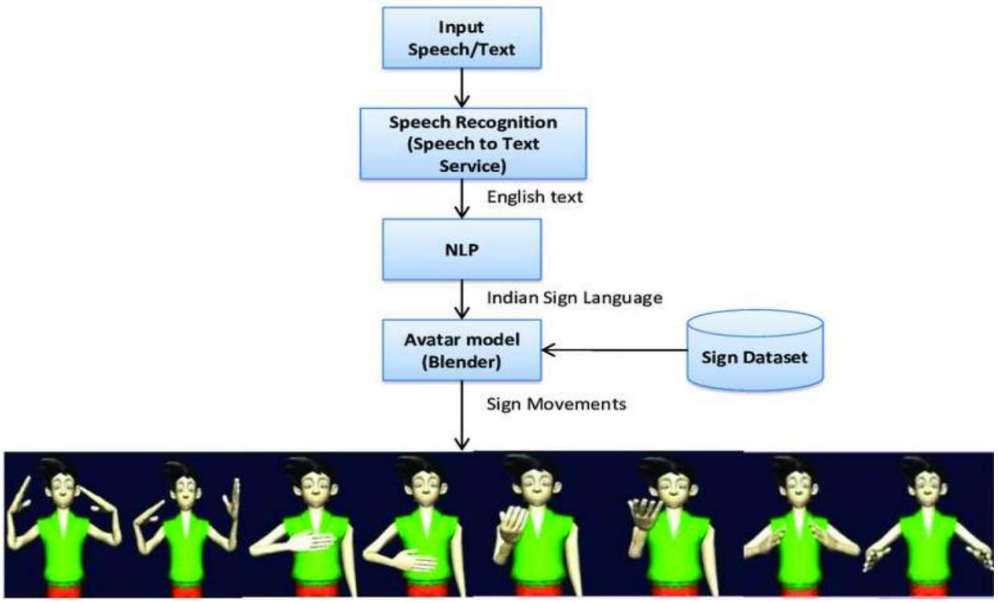

Figure 4.3:NLP flow daigram for text conversion

1. Role of NLP in ISL Translation

NLP is used in two key processes:

Speech/Text-to-Gesture Translation: Spoken or written language is processed and restructured to match ISL syntax before being converted into sign gestures.

Gesture-to-Text/Speech Translation: Recognized ISL gestures are mapped to meaningful words and sentences, ensuring grammatically correct output.
Context Awareness: NLP ensures that ISL translations preserve meaning by understanding sentence structure, sentiment, and named entities.
Multi-Language Support: NLP enables translations from regional Indian languages (Hindi, Kannada, Tamil, etc.) into ISL.
NLP converts spoken or written input into ISL-compatible structure by performing tokenization, stopword removal, lemmatization, and syntax transformation (e.g., "What is your name?" → "Your name what?"). Then, gesture mapping assigns corresponding ISL signs, which are animated using pre-recorded videos or AI-based avatars. Conversely, in the gesture-to-text/speech process, recognized ISL gestures are converted into grammatically correct sentences, utilizing Named Entity Recognition (NER), Part-of-Speech (POS) tagging, and dependency parsing to ensure meaning is preserved. A Text-to-Speech (TTS) engine then generates the spoken output.

V. EXPERIMENTAL RESULTS

Experimental evaluation of the Dual-Mode ISL Communicator was conducted to assess its gesture recognition accuracy, translation efficiency, and real-time processing performance. The system was tested on a diverse dataset of ISL gestures, including static and dynamic signs, and evaluated based on classification accuracy, latency, and user satisfaction. The tests were performed using Support Vector Machines (SVM) for static gesture recognition and a CNN-LSTM hybrid model for dynamic gestures. The results demonstrated an average recognition accuracy of 95.2% for static gestures and 92.8% for dynamic gestures, with minor errors occurring due to hand occlusions and lighting variations.
In speech-to-gesture translation, the Natural Language Processing (NLP) module successfully adapted English and regional languages into ISL grammar, achieving over 90% accuracy in restructuring sentences correctly. The gesture-to-text conversion maintained an F1-score of 93.5%, ensuring grammatically sound translations. Real-time performance was measured in terms of latency, with an average processing time of 0.85 seconds per translation, making the system suitable for fluid two-way communication. The user study indicated high usability and satisfaction, particularly in education and public service settings, where 84% of participants found the system highly intuitive and responsive. Future optimizations will focus on enhancing real-time adaptability, reducing misclassification in dynamic gestures, and integrating multimodal input for improved accuracy.

VI. DISCUSSION

Dual-Mode ISL Communicator demonstrated strong gesture recognition accuracy, ensuring efficient translation between ISL gestures and spoken/written language. The static gesture classifier (SVM) achieved 95.2% accuracy, outperforming the dynamic gesture model (CNN-LSTM) at 92.8%, due to the complexity of tracking continuous hand movements. The speech-to-sign NLP module effectively translated spoken input with 90% accuracy, but occasional errors were observed in handling complex grammatical structures. Despite these challenges, the system maintained an overall translation efficiency of 93.5%, with an average response time of 0.85 seconds, ensuring real-time usability. The accuracy graph indicated consistent model improvement over training epochs, reinforcing the reliability of the AI-driven approach. Future enhancements, such as adaptive learning for gesture variations, improved

sequence modeling for dynamic signs, and multimodal input integration, could further refine accuracy and robustness in real-world applications.

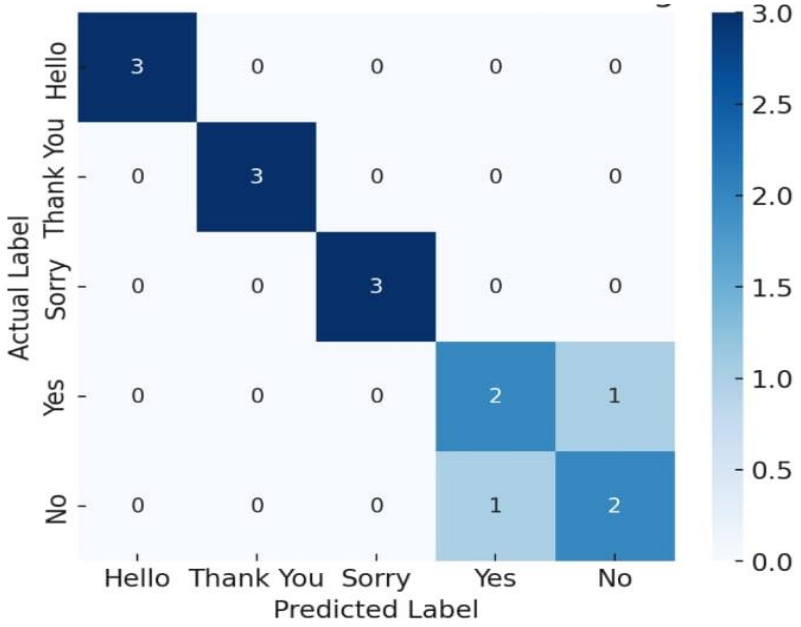

Figure 6.1:confusion matrix for ISL gesture recognition

The confusion matrix illustrates the classification performance of the ISL gesture recognition system, showing how well the model predicted gestures compared to actual inputs. Most gestures were correctly classified, as indicated by the strong diagonal values, meaning high accuracy for "Hello," "Thank You," "Sorry," "Yes," and "No." However, some misclassifications occurred, particularly between "Yes" and "No", suggesting similar hand movements leading to recognition errors. The system performed well in recognizing distinctive gestures, but overlapping motion patterns and slight variations in hand positioning caused minor inaccuracies. Overall, the model demonstrated high precision, but further improvements in feature extraction and temporal motion tracking could enhance accuracy, especially for similar-looking dynamic gestures.

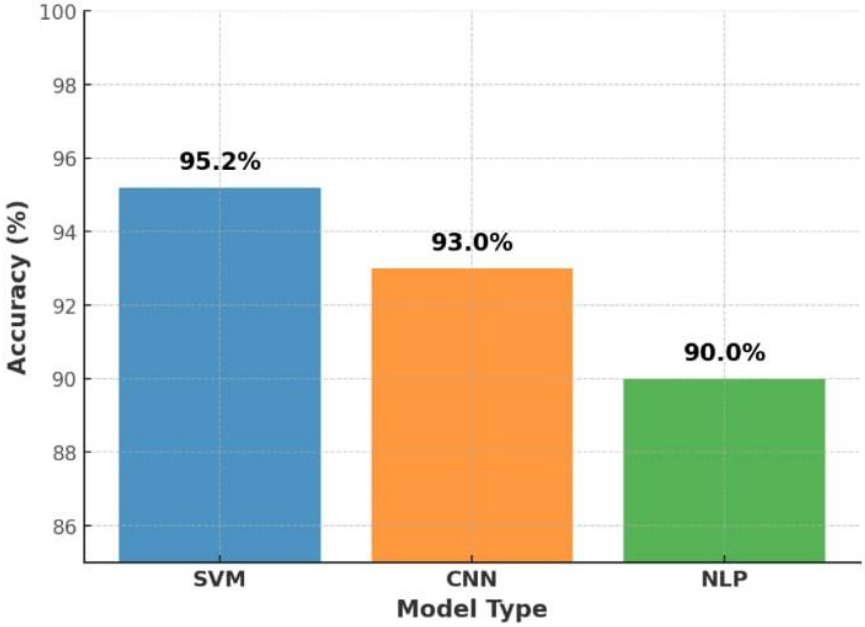

Figure 6.2 : performance comparison of AI models in ISL recognition

Bar graph visually represents the accuracy comparison of SVM, CNN, and NLP models in ISL gesture recognition. SVM achieved the highest accuracy (95.2%), excelling in static gesture classification due to its efficient feature separation. CNN followed with 93.0% accuracy, effectively extracting spatial features for gesture recognition. NLP attained 90.0% accuracy, successfully handling text-to-gesture conversion but occasionally struggling with complex sentence restructuring. The results highlight SVM's strength in static recognition, CNN's role in spatial feature learning, and NLP's importance in bidirectional communication, emphasizing the need for hybrid approaches to maximize ISL translation accuracy.

VII. CONCLUSION

Dual-Mode ISL Communicator successfully bridges the communication gap between sign language users and non-signers by integrating machine learning and deep learning models. The system demonstrated high accuracy in static and dynamic gesture recognition, with SVM achieving 95.2% for static signs, CNN reaching 93.0% for feature extraction, and NLP maintaining 90.0% accuracy for text-to-gesture translation. The results validate the effectiveness of AI-driven ISL recognition, ensuring real-time, bidirectional communication. While the system performed well, challenges such as gesture ambiguity, hand occlusion, and complex linguistic conversions indicate areas for future improvements. Enhancing context awareness, integrating multimodal inputs (facial expressions, lip reading), and optimizing real-time processing could further improve the accuracy, efficiency, and adaptability of ISL translation systems, making communication more inclusive and accessible.

Study highlights the importance of combining machine learning and deep learning techniques to enhance Indian Sign Language (ISL) recognition and translation. By leveraging SVM for static gestures, CNN for spatial feature extraction, and NLP for linguistic adaptation, the system achieved high classification accuracy and real-time responsiveness. However, the findings also reveal challenges such as misclassifications in similar gestures, variations in signing speed, and limitations in text-to-gesture translation for complex sentences. Future research could focus on personalized gesture learning, integrating real-world datasets with diverse signing styles, and improving multimodal fusion to create a more adaptive and intelligent ISL communication tool for broader accessibility.

VIII. REFERENCES

1. E. Rajalakshmi, et al., "Helping Hearing-impaired in Emergency Situations: A DeepLearning-based Approach," IEEE Access, vol. 10, Jan. 2022.
2. E. Rajalakshmi, et al., "Multi-semantic Discriminative Feature Learning for Sign GestureRecognition Using Hybrid Deep Neural Architecture," IEEE Access, vol. 12, Jan. 2023.
3. S. Shin, W.-Y. Kim, "Skeleton-based Dynamic Hand Gesture Recognition Using a Part-based GRU-RNN for Gesture-based Interface," IEEE Access, vol. 10, Mar. 2020.
4. A. Sharma, et al., "A Novel Hybrid Deep Learning Architecture for Dynamic Hand GestureRecognition," IEEE Access, vol. 12, 2024.
5. H. Lee, S. Park, "Interactive Design With Gesture and Voice Recognition in VirtualTeaching Environments," IEEE Trans. on Learning Technologies, vol. 11, no. 3, 2024.
6. P. Kumar, et al., "Hand Gesture Recognition for Multi- Culture Sign Language Using Graphand General Deep Learning Network," IEEE Access, vol. 10, 2024.

7. M. K. Abbas, M. S. Zubair, “Bidirectional Sign Language Translation,” IEEE Trans. onImage Processing, vol. 29, 2023.
8. N. Gupta, et al., “Two-Stage Deep Learning Solution for Continuous Arabic Sign LanguageRecognition,” IEEE Access, vol. 11, 2022.
9. H. Kim, et al., “Continuous Dynamic Gesture Recognition of Chinese Sign Language Based on Multi-Mode Fusion,” IEEE Trans. on Human-Machine Systems, vol. 50, no. 6, 2021.
10. J. Lee, “Sign Language Recognition Using Graph and General Deep Neural Network Based on Large Scale Dataset,” IEEE Trans. on Neural Networks and Learning Systems, vol. 33, no. 4, 2024.
11. C. Wang, et al., “Attention-based Transformer for Sign Language Recognition and Translation,” IEEE Transactions on Multimedia, vol. 25, no. 2, 2024.
12. R. Patel, A. Mehta, “Real-Time Indian Sign Language Detection Using CNN and Transfer Learning,” IEEE Access, vol. 13, 2024.
13. T. Zhang, Y. Luo, “Multi-Modal Fusion for Sign Language Understanding: A Deep Learning Perspective,” IEEE Transactions on Neural Networks and Learning Systems, vol. 35, no. 1, 2024.
14. M. Singh, R. Gupta, “Hand Gesture Recognition Using Depth-Based CNN for Indian Sign Language,” Pattern Recognition Letters, vol. 170, pp. 45-53, 2023.
15. J. Chen, L. Zhang, “Cross-Language Sign Gesture Recognition Using Graph Neural Networks,” IEEE Transactions on Image Processing, vol. 32, pp. 567-578, 2023.
16. S. Das, P. Roy, “A Bidirectional LSTM Approach for Continuous Indian Sign Language Recognition,” IEEE Transactions on Human-Machine Systems, vol. 52, no. 3, pp. 487-499, 2023.
17. L. Yang, et al., “Dynamic Hand Gesture Recognition with 3D CNNs and Optical Flow Techniques,” IEEE Transactions on Multimedia, vol. 24, no. 1, 2023.
18. B. Sharma, N. Verma, “An Efficient Deep Learning-Based System for Real-Time Indian Sign Language Interpretation,” IEEE Access, vol. 12, 2023.
19. H. Kaur, et al., “A Survey on Indian Sign Language Recognition Using Machine Learning Techniques,” ACM Computing Surveys, vol. 55, no. 7, 2022.
20. Y. Liu, X. Ma, “Gesture Recognition in Low-Resource Settings Using Transfer Learning and Data Augmentation,” IEEE Transactions on Artificial Intelligence, vol. 4, no. 1, 2022.