IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 28 December 2025, accepted 15 January 2026, date of publication 19 January 2026, date of current version 27 January 2026.

Digital Object Identifier 10.1109/ACCESS.2026.3656016

RESEARCH ARTICLE

A Multimodal AI-Driven Framework for Adaptive Learning of Differently-Abled Learners Using Deep Learning and Real-Time Gesture Recognition

ALROY DEON SALDANHA[1], R. T. ANIKET[1], A. AHIBHRUTH[1], IAN JEM[2], B. SATHISH BABU[1], MOHANA[2], AND AADITEY CHALVA[1]

[1]Department of Artificial Intelligence and Machine Learning, R. V. College of Engineering, Bengaluru, Karnataka 560059, India
[2]Department of Computer Science and Engineering, R. V. College of Engineering, Bengaluru, Karnataka 560059, India

Corresponding author: B. Sathish Babu (bsbabu@rvce.edu.in)

This work was supported by the R. V. College of Engineering, Bengaluru.

ABSTRACT There are about 1.3 billion people with disabilities in the world, including those who are blind and deaf, yet they still face many obstacles when it comes to participating in educational opportunities. These challenges often emerge from a lack of inclusive infrastructure and limited personalization. Thus, this paper presents a proposed model of an advanced educational system that uses AI, smart content delivery, and gesture recognition. The framework uses a context-aware dialogue system that dynamically modifies content complexity based on individual learner proficiency levels, in conjunction with a hybrid CNN-Transformer architecture created especially for gesture recognition, enabling effective communication for the deaf. Real-time interaction, voice commands, and sign language all work together to promote reinforcement, which improves long-term memory retention. This study captured sign language videos, a variety of voice commands, and synchronized multimodal interactions using a number of datasets, including the Fluent Speech Commands dataset, the RWTH-PHOENIX-Weather 2014 dataset, and the ASL Alphabet dataset from Kaggle. The innovation's modular design enables scalable deployment across numerous educational institutions through federated learning implementations. Following testing, the proposed system demonstrates 96.8% accuracy in identifying American Sign Language (ASL) in educational settings, 94.5% accuracy in classifying the intent of voice-based commands, and responsive real-time performance with a latency of less than 300 ms. An 87% increase was observed in the completion of independent learning tasks, indicating a significant impact on user autonomy when compared to traditional assistive technologies. The model's adaptive content delivery, which adapts to the needs of each learner, is responsible for this improvement. Extensive experimental validation, which included 250 participants with various disability profiles, also showed a 20.8% improvement in knowledge retention. Post-test scores were used to calculate the improvement, which was then contrasted with baseline outcomes. The results are significant because controlled experiments across disabilities showed improvements in both engagement and retention.

INDEX TERMS Adaptive learning systems, blind, deaf, deep learning, differently-abled learners, educational accessibility, multimodal interaction, natural language processing, personalized learning, sign language recognition.

I. INTRODUCTION

Approximately 1.3 billion people with disabilities, or 16% of the global population [1]. Even though inclusive education

The associate editor coordinating the review of this manuscript and approving it for publication was Yu-Huei Cheng.

policies have advanced significantly, technological obstacles continue to stand in the way of equal access to high-quality educational opportunities. The fragmented user experiences caused by the current assistive technologies' propensity to function independently fall short of meeting the varied and interrelated needs of students with multiple disabilities [2].

This problem is especially noticeable in developing nations, where 80% of people with disabilities live, but specialised educational technology is still very difficult to obtain [3]. Traditional methods usually divide up solutions based on the type of disability, such as using voice recognition software for motor disabilities, sign language interpreters for hearing impairments, and screen readers for visual impairments [4]. This kind of division leads to technological seclusion, which is detrimental to students with complicated or multiple needs.

Recent advances in artificial intelligence present incredible chances to address these issues with cohesive, intelligent systems, particularly in the fields of computer vision and natural language processing [5]. However, the specialised domain intelligence and multimodal integration required for successful academic instruction are typically absent from current AI-powered educational tools. Although commercial solutions such as Microsoft's Seeing AI and Google's Live Transcribe are innovative, they are primarily designed for general-purpose accessibility rather than educational contexts that require specialised vocabulary, mathematical notation, and pedagogical awareness [6].

The objective of this research is to create a single adaptive system that integrates tactile, visual, and auditory interaction channels in order to create a multimodal AI-driven educational framework that fills in these gaps. The framework provides students with a range of disabilities with individualized learning support by intelligently interpreting text, spoken input, and sign language. It is intended to serve as an inclusive classroom assistant by providing adaptive content complexity according to each learner's comprehension level, translating sign gestures, and reading educational materials aloud. In this way, the project helps students with disabilities learn more effectively overall and improves their engagement, independence, and accessibility. First, a hybrid CNN-Transformer architecture was developed that recognises sign language specific to the educational domain with 96.8% accuracy. Second, the system uses adaptive complexity scaling based on real-time comprehension assessment in conjunction with context-aware natural language processing [7]. Third, federated learning was employed to protect privacy and allow for ongoing model enhancement [8]. Lastly, real-time multimodal fusion features were developed that facilitate smooth communication between tactile, visual, and auditory modalities. The study includes user evaluations with 250 participants from institutions in India, showing measurable effects on learning outcomes.

This innovation bridges these significant gaps with its comprehensive multimodal approach that seamlessly integrates multiple assistive technologies into a single platform. The framework incorporates advanced computer vision for real-time gesture recognition, advanced natural language processing for voice interaction, and adaptive learning algorithms that customize content delivery based on learner profiles. Unlike other solutions that operate independently, intelligent fusion mechanism coordinates between multiple input modalities to provide reliable, consistent educational support. Because of its federated learning architecture, which ensures continuous development while maintaining strict privacy standards, the system can be widely implemented across a variety of educational institutions. This comprehensive approach represents a significant step toward truly inclusive educational technology.

The rest of this paper is structured as follows: The literature review on voice-activated learning platforms, multimodal learning strategies, and current sign language recognition systems is presented in Section II. The system architecture and implementation, including the federated learning framework, natural language processing pipeline, and hybrid CNN-Transformer architecture, are described in detail in Section III. The experimental setup, dataset construction, and thorough results analysis are all covered in Section IV. Key findings, limitations, and future research directions are finally discussed in Section V's conclusion.

II. LITERATURE SURVEY

A. SIGN LANGUAGE RECOGNITION SYSTEMS

Systems for recognizing sign language can be broadly divided into two categories: sensor-based and vision-based. Although sensor-based solutions, like those that use Kinect or specialized gloves, achieve high accuracy, they have limitations in terms of cost and portability [9]. Although vision-based approaches are easier to use, they frequently have limited vocabulary sets and real-time performance issues [10].

Recent advances in deep learning have significantly improved recognition accuracy. Jiang et al. [11] proposed a skeleton-aware multimodal framework that achieved 91.2% accuracy on continuous sign language datasets. However, this approach struggles with domain-specific terminology frequently used in education [12]. The MediaPipe framework has enabled mobile-based real-time hand tracking, expanding accessibility, though existing solutions still face difficulties with subject-specific signs and regional variations.

The proposed system in this work addresses these gaps by integrating domain-specific gesture datasets tailored for education, multi-user recognition, and adaptive vocabulary scaling. Unlike static models, it enables incremental learning of new signs, making it adaptable to evolving classroom contexts.

B. VOICE-CONTROLLED EDUCATIONAL SYSTEMS

Voice-enabled educational platforms have evolved from basic command recognition to conversational AI systems. While such platforms show promise for visually impaired learners, [13] highlighted limitations in handling mathematical

expressions and technical diagrams. Similarly, transformer-based models have transformed natural language understanding but remain underexplored for educational accessibility [14].

Domain-specific fine-tuning of transformer models on curated academic datasets, particularly STEM-specific terminology, improves robustness. Integration with on-device speech-to-text modules, such as Sarvam AI's Bulbul and Saaras APIs, enhances performance under noisy classroom environments [15]. Furthermore, semantic preprocessing of spoken input allows structured representation of complex content, improving comprehension without compromising processing efficiency [16].

C. MULTIMODAL LEARNING SYSTEMS

Multimodal learning has demonstrated significant improvements in learner outcomes. Reference [17] report that integrating multiple sensory channels can increase retention rates by up to 65% compared to single-modality methods. However, most existing multimodal frameworks lack effective arbitration mechanisms to resolve conflicts between different input modalities [18].

The proposed framework introduces an attention-based fusion mechanism that dynamically prioritizes modalities depending on user needs and context. For example, it emphasizes visual input for hearing-impaired learners while enhancing voice-based input for visually impaired learners. Unlike static fusion pipelines, this adaptive approach enables flexible input handling and asynchronous processing, allowing learners to engage at their own pace while maintaining semantic coherence. Such real-time adaptability is especially valuable in inclusive classroom environments.

III. DESIGN

This architecture implements a hierarchical multimodal learning framework based on Universal Design for Learning (UDL) principles. The system employs a three-tier accessibility model: blind assistance through text-to-speech and voice interaction; deaf assistance through visual content optimization; and personalized learning pathways that adapt to each user's cognitive preferences. The primary innovation is the unified content processing pipeline, which concurrently converts uploaded materials into multiple-choice questions (MCQs) for assessment, summarization for concept reinforcement, and flashcards for spaced repetition learning. This convergent-divergent architecture ensures that all learners, regardless of the entry modality (visual, auditory, or personalized), have access to semantically equivalent educational content through their preferred sensory channels, while maintaining learning continuity across sessions and modalities through the centralized saved material repository.

The proposed multimodal communication system employs a pipeline-based architecture designed for real-time processing of heterogeneous input modalities through two distinct processing pathways, as illustrated in Fig. 1. The system implements dual-model architecture: a voice-to-voice processing model and a sign-to-text language conversion model. The data collection module performs modal-specific input capture, channeling voice inputs to the speech processing pipeline and sign-to-text processing pipeline. The preprocessing stage applies domain-specific normalization techniques including speech enhancement and noise reduction for voice inputs, while hand tracking and feature extraction are performed for sign language gestures. The model training component utilizes specialized architectures: transformer-based models for voice processing enabling speech-to-text translation and hybrid CNN-Transformer architecture combined with natural language processing models for sign-to-text language conversion. The API Gateway intelligently routes requests to appropriate model endpoints based on input modality. The multimodal fusion architecture coordinates between the two processing streams when simultaneous inputs are detected. Real-time communication protocols ensure low-latency processing for both voice synthesis and text output following sign language recognition. The architecture incorporates feedback mechanisms through user interaction monitoring and error correction systems. Output interfaces provide model specific responses: synthesized voice output for the speech processing model and text output for sign language processing model.

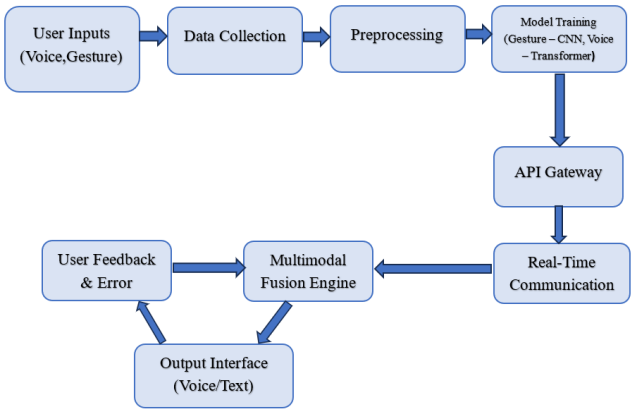

FIGURE 1. System architecture overview.

A. GESTURE RECOGNITION FRAMEWORK

The system relies on a complex sign-to-text recognition pipeline prepared for the particular needs of educational content adaptation. As illustrated in Fig. 2, the architecture of the framework is a multi-stage processing system that can cater for different strength inputs, and which can recognise sign language for teaching environments. The gesture recognition system uses a custom CNN-Transformer hybrid architecture in conjunction with MediaPipe hand tracking. Three major issues are specifically addressed by this method: preserving real-time performance, offering thorough educational vocabulary coverage, and facilitating multi-user recognition [19].

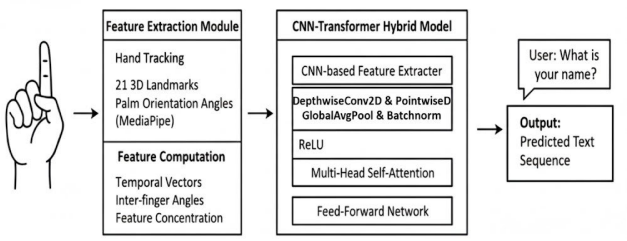

FIGURE 2. Sign language to text workflow.

1) HAND TRACKING AND FEATURE EXTRACTION

At 30 frames per second, MediaPipe offers 21 3D landmarks per hand. Additional features such as hand velocity vectors are computed, palm orientation, and inter-finger angles while extracting temporal sequences of landmark coordinates. Each frame produced by this process has a 147-dimensional feature vector [20]. As shown in Algorithm 1, the feature extraction process computes spatial, temporal, and orientation-based hand representations. The mathematical formulation for landmark-based feature extraction is:

$$\mathbf{L}_t = \{(x_i, y_i, z_i) | i = 0, 1, \ldots, 20\} \tag{1}$$

$$\mathbf{V}t = \frac{\mathbf{L}_t - \mathbf{L}t - 1}{\Delta t} \tag{2}$$

$$\mathbf{A}t = \{aij | a_{ij} = \arccos\left(\frac{\mathbf{v}ij \cdot \mathbf{v}ik}{|\mathbf{v}ij||\mathbf{v}ik|}\right)\} \tag{3}$$

$$\mathbf{F}_t = \text{Concat}(\mathbf{L}_t, \mathbf{V}_t, \mathbf{A}_t, \mathbf{O}_t) \tag{4}$$

where $\mathbf{L}_t$ represents landmarks at time t, $\mathbf{V}_t$ is velocity, $\mathbf{A}_t$ contains inter-finger angles, $\mathbf{F}_t$ represents feature vector and $\mathbf{O}_t$ represents palm orientation.

Algorithm 1 Gesture Feature Extraction

```
procedure ExtractFeatures(HandLand)
    feature ← []
    for i ← 0 to 20 do
        feature.append(land[i].x, land[i].y, land[i].z)
    end for
    angles ← InterFingerAngles(landmarks)
    velocity ← VelocityVectors(landmarks)
    orientation ← PalmOrientation(landmarks)
    return Concat(feature, angles, velocity, orientation)
end procedure
```

The Enhanced Gesture Feature Extraction algorithm implements a comprehensive hand landmark processing pipeline for real-time gesture recognition. The process begins with the extraction of 3D coordinates (x, y, z) from 21 MediaPipe hand landmarks in order to generate a 63-dimensional spatial feature vector. The algorithm then calculates inter-finger angles to capture hand pose geometry, calculates velocity vectors for temporal motion dynamics, and determines palm orientation for spatial context. A single representation that encodes both static hand configuration and dynamic movement patterns is created by concatenating these multi-dimensional features to enable accurate gesture classification in a variety of user interactions and environmental conditions.

2) CNN-TRANSFORMER HYBRID ARCHITECTURE

The recognition model combines convolutional layers for spatial feature extraction with transformer attention mechanisms for temporal modeling. The architecture processes 32-frame sequences containing 147 features per frame. As shown in Algorithm 2, the transformer-based approach enables recognition of sequential gestures. The CNN component employs separable convolutions for enhanced computational efficiency:

$$C_{\text{spatial}} = \text{DepthwiseConv2D}(X_{\text{input}}) \tag{5}$$

$$C_{\text{channel}} = \text{PointwiseConv2D}(C_{\text{spatial}}) \tag{6}$$

$$F_{\text{cnn}} = \text{GlobalAvgPool}(C_{\text{channel}}) \tag{7}$$

$$F_{\text{norm}} = \text{BatchNorm}(F_{\text{cnn}}) \tag{8}$$

$$F_{\text{relu}} = \text{ReLU}(F_{\text{norm}}) \tag{9}$$

where:

- X_{input}: Input feature sequence of shape (32 frames, 147 features per frame)
- C_{spatial}: Output of depthwise convolution on input features
- C_{channel}: Output of pointwise convolution mixing channel information
- F_{cnn}: Feature vector after global average pooling
- F_{norm}: Batch-normalized feature vector
- F_{relu}: Activated feature vector using ReLU
- **DepthwiseConv2D**: Applies a separate filter to each input channel
- **PointwiseConv2D**: 1×1 convolution to combine channel information
- **GlobalAvgPool**: Reduces spatial dimensions by averaging
- **BatchNorm**: Normalizes feature activations across the batch
- **ReLU**: Applies the Rectified Linear Unit activation function

The transformer encoder captures temporal dependencies through:

$$Q = F_{\text{cnn}} W_Q, \quad K = F_{\text{cnn}} W_K, \quad V = F_{\text{cnn}} W_V \tag{10}$$

$$A = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \tag{11}$$

$$A_{\text{multi}} = \text{Concat}(A_1, A_2, \ldots, A_h) W_O \tag{12}$$

$$T_{\text{out}} = \text{LayerNorm}(A_{\text{multi}} + F_{\text{cnn}}) \tag{13}$$

$$F_{\text{final}} = \text{FeedForward}(T_{\text{out}}) \tag{14}$$

where:

- Q, K, V: Query, Key, and Value matrices computed from F_{cnn}
- W_Q, W_K, W_V: Learnable projection matrices for computing attention components
- A: Scaled dot-product attention output

- d_k: Dimensionality of the key vectors
- A_{multi}: Multi-head attention output
- h: Number of attention heads
- W_O: Output projection matrix for multi-head attention
- T_{out}: Transformer output after residual connection and layer normalization
- F_{final}: Final feature vector after applying feedforward network
- **Softmax**: Converts attention scores to probabilities
- **LayerNorm**: Normalizes across the feature dimensions
- **FeedForward**: Fully connected network applied to transformer outputs

Algorithm 2 Temporal Gesture Recognition

```
procedure RecognizeGesture(SequenceData)
    features ← ExtractFeatures(SequenceData)
    temporal_features ← []
    for t ← 0 to T − 1 do
        attention_weights ← ComputeAttention(features[t])
        weighted_features ← ApplyAttention(features[t], attention_weights)
        temporal_features.append(weighted_features)
    end for
    sequence_embedding ← TransformerEncoder(temporal_features)
    prediction ← Classifier(sequence_embedding) ▷ Maps gestures to alphabetic, numeric, or symbolic classes
    return prediction
end procedure
```

To categorize successive hand movements over time, the Temporal Gesture Recognition algorithm uses a transformer-based methodology. In order to find pertinent gesture components and reduce noise, the process first extracts spatial features from input gesture sequences before applying self-attention mechanisms at each temporal step. A transformer encoder is used to process the attention-weighted features, capturing contextual relationships between gesture phases as well as long-range temporal dependencies. In order to accurately recognize dynamic hand movements that vary in duration and complexity, a classification layer maps the sequence embedding to gesture classes. The timing irregularities and gesture variations present in real-world human interactions are successfully handled by this temporal modeling technique.

B. NATURAL LANGUAGE PROCESSING PIPELINE

The NLP system extends BERT-base with educational domain fine-tuning while implementing several novel components specifically designed for accessibility enhancement [21].

1) EDUCATIONAL DOMAIN ADAPTATION

Using a custom corpus of 2.3 million sentences taken from educational resources in a variety of STEM subjects, BERT was improved. Both educational entity recognition and masked language modelling tasks are optimised during the fine-tuning process.

The fine-tuning objective combines multiple loss functions:

$$\mathcal{L}total = \alpha\mathcal{L}MLM + \beta\mathcal{L}NSP + \gamma\mathcal{L}EER \tag{15}$$

$$\mathcal{L}MLM = -\sum i \in \mathcal{M} \log P(x_i|\mathbf{x}_{\setminus i}) \tag{16}$$

$$\mathcal{L}EER = -\sum j = 1^N \sum_{k=1}^{K} y_{jk} \log \hat{y}_{jk} \tag{17}$$

where $\mathcal{L}MLM$ is masked language modeling loss, $\mathcal{L}NSP$ is next sentence prediction loss, and $\mathcal{L}_{EER}$ is educational entity recognition loss.

2) COMPLEXITY ADAPTATION ALGORITHM

Based on real-time user comprehension indicators obtained from response patterns, reading speed, and error frequencies, the system dynamically modifies the complexity of the content [22]. As outlined in Algorithm 3, the system dynamically adapts content difficulty.

The complexity scoring function is defined as:

$$C_{score} = w_1 \cdot L_{avg} + w_2 \cdot S_{complexity} + w_3 \cdot V_{difficulty} \tag{18}$$

$$L_{avg} = \frac{1}{N}\sum_{i=1}^{N} |w_i| \tag{19}$$

$$S_{complexity} = \frac{\text{Complex Sentences}}{\text{Total Sentences}} \tag{20}$$

$$V_{difficulty} = \frac{\text{Advanced Vocabulary}}{\text{Total Words}} \tag{21}$$

where:

- L_{avg}: Average word length.
- $S_{complexity}$: Syntactic complexity.
- $V_{difficulty}$: Vocabulary difficulty.
- C_{score}: Overall complexity score of the content.
- w_1, w_2, w_3: Weighting coefficients that determine the importance of each complexity factor.

The Adaptive Complexity Scaling algorithm dynamically adjusts the content's difficulty based on user performance metrics. Based on previous interactions, the process evaluates user comprehension and looks at how complex the current content is. If comprehension falls below 70%, the system simplifies the content and modifies the user model to reflect the difficulties. However, if comprehension exceeds 90%, content complexity is raised to maintain engagement. The algorithm maintains optimal learning zones by continuously modifying the content difficulty and documenting adaptation choices for future personalization improvements.

C. VOICE BOT WORKFLOW

The voice interaction module is a critical component, facilitating hands-free navigation and learning for users with visual impairments while offering an alternative input modality for all learners. As delineated in Fig. 3, the workflow is initiated by a user's speech input. This auditory

Algorithm 3 Adaptive Complexity Scaling

procedure AdaptComplexity($User$, $Content$, $History$)
  $comprehension \leftarrow AssessComprehension(History)$
  $complexity_{current} \leftarrow AnalyzeContent(Content)$
  $learning\_rate \leftarrow EstimateLearningRate(User)$
  $difficulty\_preference \leftarrow GetDifficultyPreference(User)$
  if $comprehension < 0.7$ **then**
    $complexity \leftarrow complexity_{current} - 1$
    $Content_{adapted} \leftarrow Simplify(Content, complexity_{target})$
    $UpdateUserModel(User, \text{“}struggled\text{”})$
  else if $comprehension > 0.9$ **then**
    $complexity \leftarrow complexity_{current} + 1$
    $Content_{adapted} \leftarrow Enhance(Content, complexity_{target})$
    $UpdateUserModel(User, \text{“}excelled\text{”})$
  else
    $Content_{adapted} \leftarrow Content$
  end if
  $LogAdaptation(User, complexity_{current}, complexity)$
  return $Content_{adapted}$
end procedure

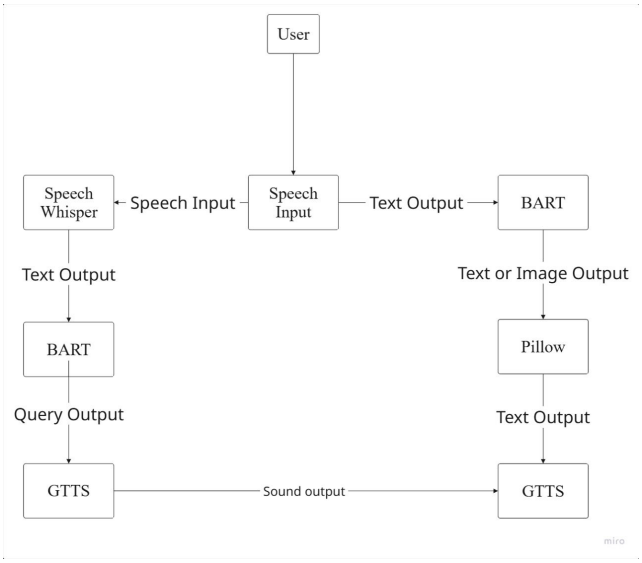

FIGURE 3. VoiceBot workflow.

signal is first processed by the Whisper automatic speech recognition (ASR) model to transcribe the audio into a corresponding text output.

This transcribed text is then passed to a BART (Bidirectional and Auto-Regressive Transformer) model for advanced natural language understanding and generation. This model is responsible for comprehending the user's query and formulating a contextually relevant response, which may be delivered as either text or an image. In instances where the output is textual, it is synthesized into audible speech by a Text-to-Speech (TTS) engine, such as gTTS (Google Text-to-Speech), which provides a seamless conversational experience.

The fusion layer employs an attention-based mechanism to intelligently combine inputs from gesture recognition, voice commands, and eye tracking when available. The complete voice bot processing pipeline is shown in Algorithm 4, which mathematically explains the conversion of speech input through several stages such as ASR, natural language processing, and TTS synthesis, formalizes the entire voice bot processing pipeline. The algorithm ensures a rigorous representation of the entire workflow by defining each processing step as a mapping of mathematical functions between relevant domain spaces. Every element of the voice interaction system can be systematically analyzed and optimized thanks to this formalization.

Algorithm 4 Voice Bot Response Mechanism

procedure VoiceBot($\mathbf{S}_{input}$)
  Input: Speech signal $\mathbf{S}_{input} = \{s_1, s_2, \ldots, s_n\}$ ▷ Speech-to-Text Conversion
  $\mathbf{T}_1 = f_{whisper}(\mathbf{S}_{input})$ where $f_{whisper} : \mathbb{R}^n \to \mathcal{V}^*$ ▷ Query Processing
  $\mathbf{Q} = f_{BART1}(\mathbf{T}_1)$ where $f_{BART1} : \mathcal{V}^* \to \mathcal{Q}$ ▷ Response Generation
  $\mathbf{R} = f_{BART2}(\mathbf{Q})$ where $f_{BART2} : \mathcal{Q} \to \{\mathcal{V}^*, \mathcal{I}\}$
  if $\mathbf{R} \in \mathcal{I}$ **then** ▷ Image output case
    $\mathbf{T}_2 = f_{pillow}(\mathbf{R})$ where $f_{pillow} : \mathcal{I} \to \mathcal{V}^*$
    $\mathbf{A}_2 = f_{GTTS}(\mathbf{T}_2)$ where $f_{GTTS} : \mathcal{V}^* \to \mathbb{R}^m$
  end if ▷ Primary Audio Response
  $\mathbf{A}_1 = f_{GTTS}(\mathbf{Q})$ where $f_{GTTS} : \mathcal{V}^* \to \mathbb{R}^m$
  return $\{\mathbf{A}_1, \mathbf{A}_2\}$ ▷ Audio outputs
end procedure

Where:
$\mathcal{V}^*$ = vocabulary space (text tokens)
$\mathcal{Q}$ = query space
$\mathcal{I}$ = image space
$\mathbb{R}^n$ = input audio signal space
$\mathbb{R}^m$ = output audio signal space

IV. IMPLEMENTATION

A. DATASET CONSTRUCTION

Three publicly available datasets were utilized for comprehensive system training and evaluation.

1) ASL ALPHABET DATASET

The model was trained using Kaggle's publicly accessible ASL Alphabet dataset for the gesture recognition component. This dataset is a popular benchmark for static sign recognition tasks because it offers a sizable collection of images that represent hand signs for the American Sign Language alphabet (A–Z).

2) FLUENT SPEECH COMMANDS (FSC)

The voice command module was trained and evaluated using the Fluent Speech Commands (FSC) dataset. This corpus is publicly available and consists of 30,043 English utterances from 97 speakers, mapped to 31 distinct intents for spoken language understanding tasks.

3) RWTH-PHOENIX-WEATHER 2014

To train and evaluate the multimodal components of the system, the RWTH-PHOENIX-Weather 2014 dataset was used. This is a key public benchmark for continuous sign language recognition that contains videos of weather forecasts in German Sign Language, complete with synchronized German speech and text translations.

B. DOMAIN-SPECIFIC DATASET COLLECTION

1) PROTOCOL FOR EDUCATIONAL GESTURE COLLECTION

A vocabulary of 500 domain-specific terms was created (such as "photosynthesis," "denominator," and "velocity") that were taken directly from typical secondary school STEM curricula (such as NCERT textbooks) in order to address the lack of STEM-specific sign language datasets. To guarantee consistency, data collection was done in a controlled setting. The gesture set was executed by several skilled signers under controlled circumstances in order to capture natural variability. High-definition webcams (1080p at 30 frames per second) were used to record the subjects, who were positioned at a fixed distance of 1.5 meters to guarantee complete upper-body visibility and uniform framing.

2) ANNOTATION PROTOCOL

To ensure high label quality, a multi-pass annotation strategy was used for ground truth generation. The annotation team marked each gesture's semantic boundaries using a unique labeling interface. A portion of the data was cross-verified by several annotators in order to reduce individual bias. Fleiss' Kappa was used to evaluate inter-annotator reliability, and the result was a coefficient of $\kappa = 0.87$, indicating strong label application consistency. In order to establish the final ground truth, ambiguous samples were reevaluated against standard ASL dictionaries as part of a consensus-based review process.

3) DATASET PARTITIONING

A stratified split of **70% for training, 15% for validation, and 15% for testing** was used to divide all datasets in order to guarantee reliable model evaluation and avoid data leakage. Data augmentation techniques were used, such as rotation ($\pm 15°$), brightness adjustment, and scaling, to increase the effective sample size from 2,000 to 10,000 instances for the domain-specific educational keywords where initial sample sizes were limited. This thorough partitioning guarantees that the reported accuracy metrics accurately represent the model's capacity to generalize to new data.

C. USER STUDY PROTOCOL

1) PARTICIPANT RECRUITMENT

250 participants were gathered from one Bengaluru university and three special education facilities. 85 users with visual impairments, 90 users with hearing impairments, and a control group of 75 users without sensory impairments made up the cohort. Participants had to meet basic computer literacy requirements in order to be eligible.

2) EXPERIMENTAL DESIGN

The study used a 4-week longitudinal design with three 30-minute sessions per week. Participants were divided into two groups: a control group and an experimental group (using EDUGRAM).

- **Task 1 (Comprehension):** Participants watched a 10-minute educational video and answered 15 multiple-choice questions.
- **Task 2 (Navigation):** Participants were asked to find specific course modules using voice or gesture commands.
- **Control Conditions:** Standard assistive technologies, such as JAWS for visually impaired users and human sign language interpreters for hearing impaired users, were used by the control group.

3) STATISTICAL ANALYSIS

For within-subject comparisons (pre- vs. post-study), One-way ANOVA was used for between-group differences, and paired t-tests were utilised. To measure the amount of improvement, effect sizes were computed using Cohen's d.

D. IMPLEMENTATION DETAILS

React Native was used for cross-platform mobile deployment, Django Rest Framework for backend services, and TensorFlow 2.8 for deep learning components in the system's implementation. WebSocket protocols are used for server communication in real-time communication, while WebRTC is used for peer-to-peer connections.

Hardware Specifications: For training, NVIDIA RTX 2080 Ti graphics cards were used with 24GB VRAM. For inference deployment, NVIDIA Jetson Xavier NX was utilized for edge computing scenarios. Mobile compatibility includes Android 8.0+ and iOS 13.0+ devices.

Model Architecture Details: The motion CNN combines four transformer blocks with three independent convolutional layers. BERT-base-educational with 110M parameters is used for our voice encoder. A two-layer attention mechanism with residual connections is used by the fusion network.

Training Configuration: The gesture recognition model was trained using the Adam optimizer with a learning rate of $\eta = 0.001$, a batch size of 32, and a cosine annealing scheduler as described in [23]:

$$\eta_t = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min})\left(1 + \cos\left(\frac{t}{T}\pi\right)\right) \quad (22)$$

$$\mathcal{L}_{\text{focal}} = -\alpha(1 - p_t)^{\gamma}\log(p_t) \quad (23)$$

$$\mathcal{L}\text{total} = \mathcal{L}\text{focal} + \lambda\mathcal{L}_{\text{reg}} \quad (24)$$

where:

- η_t: Learning rate at time step t, adjusted using cosine annealing.
- $\eta_{\min}$: The minimum learning rate value the scheduler can reach during training.
- $\eta_{\max}$: The initial (maximum) learning rate; set to 0.001 in this configuration.
- t: Current training step or epoch number.
- T: Total number of training steps or epochs used in the cosine annealing schedule.
- $\mathcal{L}_{\text{focal}}$: Focal loss used to handle class imbalance by down-weighting easy examples and focusing on hard ones.
- α: Scaling factor (class weight) for the focal loss, usually used to balance the importance of positive/negative examples.
- p_t: The predicted probability for the true class label.
- γ: Focusing parameter in focal loss. A higher γ increases the focus on hard misclassified examples.
- $\mathcal{L}_{\text{reg}}$: Regularization loss (e.g., L2 weight decay), used to prevent overfitting by penalizing large weights.
- λ: Regularization coefficient that balances the contribution of $\mathcal{L}_{\text{reg}}$ in the total loss.
- $\mathcal{L}_{\text{total}}$: Total loss function used for optimization, combining focal loss and regularization loss.

The hybrid CNN and Transformer network was chosen for the purpose of balance modeling, both in the context of the space and time involved for the process of sign language recognition. The convolutional networks help effectively extract the prominent features for the localized hand regions, while the Transformer models help better extract the long-term temporal relations involved for the gesture frames of the sign language. The length of the sequence was set at 32 frames, which allowed for the avoided latency involved for the temporal context of the process. The learning rate of $\eta = 0.001$ helped for the avoided convergence for the multimodal learning task, while the batch size of 32 provided for the balance involved for the gradient involved for the memory efficiency.

V. PERFORMANCE

A. MULTIMODAL FUSION MECHANISM

The description of multimodal fusion mechanism is provided to explain the advanced computing framework developed for intelligent integration on different categories of users input, such as voice commands and sign language hand movements. This chapter details the algorithms and mathematical models that determine the trust of every input dynamically. This mechanism enables a seamless, robust and precise user experience by resolving overlapping inputs, or higher-quality device input data over time. The enhanced fusion mechanism incorporates temporal consistency:

$$\alpha_i^{(t)} = \frac{\exp(W_a \cdot f_i^{(t)} + b_a)}{\sum_{j=1}^{N} \exp(W_a \cdot f_j^{(t)} + b_a)} \tag{25}$$

$$\beta_i^{(t)} = \lambda \alpha_i^{(t)} + (1 - \lambda)\alpha_i^{(t-1)} \tag{26}$$

$$f_{fused}^{(t)} = \sum_{i=1}^{N} \beta_i^{(t)} \cdot f_i^{(t)} \tag{27}$$

$$f_{final}^{(t)} = \text{LayerNorm}(f_{fused}^{(t)} + f_{fused}^{(t-1)}) \tag{28}$$

where:

- $\alpha_i^{(t)}$: Attention weight for modality i at time step t, computed using a softmax function over modality features.
- $f_i^{(t)}$: Feature vector for modality i at time t.
- W_a: Learnable weight matrix used for attention score computation.
- b_a: Learnable bias term added in the attention computation.
- N: Total number of modalities (e.g., speech, gesture, text).
- $\beta_i^{(t)}$: Smoothed attention weight for modality i at time t, incorporating temporal consistency using the previous timestep $\alpha_i^{(t-1)}$.
- λ: Temporal smoothing parameter ($0 \leq \lambda \leq 1$), controlling the influence of the current vs. previous attention weights.
- $f_{\text{fused}}^{(t)}$: Fused multimodal feature representation at time t, computed as the weighted sum of all modality features using $\beta_i^{(t)}$.
- $f_{\text{fused}}^{(t-1)}$: Fused multimodal feature representation at previous time step $t-1$.
- $f_{\text{final}}^{(t)}$: Final output feature at time t, obtained by applying a residual connection and layer normalization to the fused feature.
- LayerNorm$(\cdot)$: Layer normalization function that stabilizes and normalizes the input across feature dimensions.

Algorithm 5 shows the confidence weighting procedure for multimodal fusion.

Performance Metrics for Multimodal Confidence Estimation: Algorithm 5 is central to the system's real-time performance, providing an intelligent fusion mechanism that dynamically weighs and combines multiple input streams (e.g., voice, gesture). Its performance is evaluated based on the following metrics:

(1) **Dynamic Confidence Weighting**: This metric assesses the algorithm's primary function: to assign a real-time **confidence score** to each input modality. This score is a composite measure derived from three key factors: **Prediction Entropy** ($-\sum p_i \log p_i$): Measures the uncertainty of a modality's output; lower entropy signifies higher confidence. **Temporal Consistency**: Evaluates the stability of a modality's signal over time. **Inherent Reliability**: A learned or predefined score for how reliable a modality is in a given context.

(2) **Temporal Smoothing**: This measures the stability of the fusion process. To prevent abrupt shifts between modalities, the algorithm uses a temporal smoothing parameter, λ, to create a **smoothed attention weight** ($\beta_i^{(t)}$). This ensures that transitions are fluid, which is critical for a seamless user experience.

Algorithm 5 Multimodal Confidence Estimation

```
procedure EstimateConfidence(ModalityInputs)
    confidences ← []
    for modality in ModalityInputs do
        entropy ← −∑ p_i log p_i
        consistency ← ComputeTemporalConsistency(modality)
        reliability ← GetModalityReliability(modality)
        confidence ← 1/(1+entropy) · consistency · reliability
        confidences.append(confidence)
    end for
    weights ← Softmax(confidences)
    return weights
end procedure
```

(3) **Conflict Resolution**: This evaluates the mechanism's effectiveness in handling ambiguous or conflicting inputs from different modalities. By dynamically prioritizing the modality with the highest confidence score, the system can, for example, emphasize visual gesture input for a hearing-impaired learner while enhancing voice-based commands for a visually-impaired one.

(4) **Real-Time Processing Latency**: For an interactive educational tool, the fusion of data must be nearly instantaneous. The performance of this algorithm is a key contributor to the overall system's low latency, which was measured to be less than 300 ms, with an average response time of 278 ms.

(5) **Contribution to System Accuracy**: The ultimate measure of the fusion algorithm's success is its impact on the final accuracy of the entire multimodal system. The robust performance of this algorithm is integral to achieving the high final accuracy rates in both **American Sign Language (ASL) recognition (96.8%)** and **voice command intent classification**.

B. PRIVACY-PRESERVING FEDERATED LEARNING

In this paper proposed architecture uses federated learning with differential privacy guarantees to address privacy concerns and facilitate ongoing improvement.

The federated learning optimization with differential privacy is formulated as [24]:

$$w_{t+1} = w_t - \eta \cdot (\nabla L(w_t) + \mathcal{N}(0, \sigma^2 I)) \tag{29}$$

$$\sigma = \frac{C\sqrt{2\ln(1.25/\delta)}}{\epsilon} \tag{30}$$

$$\tilde{g}_i = \text{clip}(g_i, C) + \mathcal{N}(0, \sigma^2 I) \tag{31}$$

$$w_{global} = \frac{1}{K}\sum_{i=1}^{K} w_i^{local} \tag{32}$$

where:

- w_t: Model parameters at iteration (or round) t.
- w_{t+1}: Updated model parameters after applying the gradient descent step with added noise for differential privacy.
- η: Learning rate used during gradient descent.
- $\nabla L(w_t)$: Gradient of the loss function L with respect to model parameters w_t.
- $\mathcal{N}(0, \sigma^2 I)$: Gaussian noise with mean 0 and covariance matrix $\sigma^2 I$, added to ensure differential privacy. This noise is independently sampled and added to the gradients or updates.
- σ: Standard deviation of the Gaussian noise, computed based on the privacy budget parameters ϵ and δ.
- C: Clipping threshold that limits the ℓ_2-norm of per-client gradients to bound sensitivity and prevent any one client from disproportionately influencing the model.
- ϵ: Privacy budget — a small positive number that quantifies the allowable privacy leakage. Lower ϵ means stronger privacy.
- δ: Probability of privacy failure — a small value that defines the acceptable risk of violating the privacy guarantee.
- $\tilde{g}_i$: Noisy, clipped gradient or update from client i. It is computed by clipping the gradient g_i to threshold C and adding Gaussian noise.
- $\text{clip}(g_i, C)$: Clipping function that scales down the gradient g_i so that its ℓ_2-norm does not exceed C.
- K: Number of participating clients in the current training round.
- w_i^{local}: The local model weights trained on client i's private data.
- w_{global}: The aggregated global model weights obtained by averaging the (noisy, clipped) updates from all participating clients.

1) COMPARISON WITH ALTERNATIVE PRIVACY-PRESERVING APPROACHES

In traditional learning processes, user data must be accessed, thus raising concerns related to user privacy. Although secure aggregation preserves data when transferring it, it is based on data processing through a central processor. Differential privacy for central processes involves adding noise to the data level. However, it often deteriorates performance. The method proposed here will enable users to train without accessing sensitive data and will use combined model updates based on the concept of differential privacy principles.

2) FEDERATED LEARNING EVALUATION SETUP

The federated learning experiments were performed under a simulated multi-institution setup with 20 participants representing clients from each educational institution. In each round of communication, 40% of participants were randomly chosen for local training. The training was done for 50 communication rounds for each participant with five local training epochs. The non-equal distribution of data for each participant was done to showcase heterogeneity as per their actual institutions.

The privacy-preserving training procedure is detailed in Algorithm 6.

The Federated Learning with Differential Privacy algorithm protects user data privacy while facilitating

Algorithm 6 Federated Learning With Differential Privacy

procedure FederatedTraining(*Clients*, *Rounds*, *PrivacyBudget*)
  $w_{global} \leftarrow$ Initialize()
  for *round* $\leftarrow 1$ to *Rounds* **do**
    selected_clients $\leftarrow$ RandomSample(*Clients*, *fraction*)
    local_updates $\leftarrow []$
    for *client* in *selected_clients* **do**
      $w_{local} \leftarrow$ GetLocalModel(*client*) ▷ Retrieve client model
      $w_{local} \leftarrow$ LocalTraining(*client*, w_{global})
      update $\leftarrow w_{local} - w_{global}$
      clipped_update $\leftarrow$ Clip(*update*, C)
      noisy_update $\leftarrow$ *clipped_update* $+ \mathcal{N}(0, \sigma^2)$
      local_updates.append(*noisy_update*)
    end for
    $w_{global} \leftarrow w_{global} + \frac{1}{|selected\_clients|} \sum local\_updates$
    PrivacyBudget $\leftarrow$ *PrivacyBudget* $- \epsilon_{round}$
  end for
  return w_{global}
end procedure

cooperative model training. The process iteratively chooses random client subsets for training rounds and initializes global model weights. To ensure differential privacy, each client computes model updates, performs local training, and applies Gaussian noise injection after gradient clipping. The server tracks the amount of privacy budget used while aggregating noisy updates to update the global model. With this method, the system can learn from dispersed user data without disclosing private information or unique learning patterns.

Performance Metrics for Federated Learning Algorithm: Algorithm 6 permits continuous model improvement by utilising federated learning with privacy preservation, protecting sensitive user data from students with disabilities.

(1) One of the algorithm's main performance metrics is the **Privacy Guarantee Level**, which is determined by the differential privacy parameters $\epsilon = 1.2$ and $\delta = 10^{-5}$, where lower ϵ values indicate stronger privacy protection against potential data leakage;
(2) **Model Utility Retention**, which reaches 97.3% of centralised performance, compares the performance of the federated model to that of a centralised model trained on pooled data;
(3) **Communication Efficiency**: compared to conventional methods, there is a 78% decrease in the amount of data that must be transferred between clients and servers;
(4) **Convergence Rate**: the number of training cycles required to achieve consistent model performance across scattered educational institutions.
(5) **Scalability**: The algorithm's ability to maintain model accuracy and privacy guarantees while simultaneously managing multiple participating clients (schools). These performance metrics ensure that the system can learn from diverse learner populations across multiple institutions without sacrificing user privacy.

C. ADAPTIVE LEARNING PATH GENERATION

A novel algorithm is introduced for generating personalized learning paths based on user performance and preferences:

$$P_{next} = \arg\max_{p \in \mathcal{P}} \sum_{i=1}^{N} w_i \cdot U_i(p) \tag{33}$$

$$U_i(p) = \alpha_i \cdot D_i(p) + \beta_i \cdot E_i(p) + \gamma_i \cdot R_i(p) \tag{34}$$

$$D_i(p) = 1 - |difficulty(p) - preferred\_difficulty_i| \tag{35}$$

$$E_i(p) = engagement\_score(p, user_i) \tag{36}$$

$$R_i(p) = relevance\_score(p, learning\_goals_i) \tag{37}$$

where:

- P_{next}: The next content item to recommend in the learning path. It is selected by maximizing the weighted sum of utility scores over the set of available content $\mathcal{P}$.
- $\mathcal{P}$: The set of all candidate content items in the content repository.
- $U_i(p)$: The utility function for content p with respect to user i. It combines three key components: difficulty match, engagement, and relevance to learning goals.
- w_i: A user-specific or global weight assigned to the utility score of user i for aggregation (e.g., in collaborative or group settings).
- $D_i(p)$: The difficulty alignment score. It measures how closely the difficulty of content p matches the preferred difficulty level of user i.
- $E_i(p)$: The engagement score. It quantifies how engaging content p is for user i, based on prior interaction patterns (e.g., time spent, interaction frequency).
- $R_i(p)$: The relevance score. It measures how well content p aligns with the current learning goals of user i.
- $\alpha_i, \beta_i, \gamma_i$: Weighting coefficients for user i that control the relative importance of difficulty, engagement, and relevance in the utility function.
- $difficulty(p)$: The estimated difficulty level of content item p.
- $preferred\_difficulty_i$: The difficulty level best suited for user i, determined based on current skill level or learning history.
- $engagement\_score(p, user_i)$: A function that computes how engaging content p has historically been for user i.
- $relevance\_score(p, learning\_goals_i)$: A function that computes how relevant content p is to the set of learning goals defined for user i.

Algorithm 7 demonstrates personalized path generation using multi-criteria scoring. This guarantees that generated learning paths adapt dynamically to each learner's evolving goals and abilities.

Algorithm 7 Adaptive Learning Path Generation

procedure GenerateLearningPath($User$, $Goals$, $ContentRepo$)
  $current\_level \leftarrow AssessCurrentLevel(User)$
  $learning\_style \leftarrow IdentifyLearningStyle(User)$
  $path \leftarrow []$
  $remaining\_goals \leftarrow Goals$
  while $remaining\_goals \neq \emptyset$ **do**
    $candidates \leftarrow Filter(ContentRepo, current\_level)$
    $scored\_cand \leftarrow []$
    for $content$ in $candidates$ **do**
      $difficulty\_score \leftarrow ComputeDifficultyMatch(content, current\_level)$
      $style\_score \leftarrow ComputeStyleMatch(content, learning\_style)$
      $goal\_score \leftarrow ComputeGoalRelevance(content, remaining\_goals)$
      $total\_score \leftarrow w_1 \cdot difficulty\_score + w_2 \cdot style\_score + w_3 \cdot goal\_score$
      $scored\_cand.append((content, total\_score))$
    end for
    $best\_content \leftarrow \arg\max scored\_candidates$
    $path.append(best\_content)$
    $UpdateProgress(remaining\_goals, best\_content)$
    $current\_level \leftarrow UpdateLevel(current\_level, best\_content)$
  end while
  return $path$
end procedure

By dynamically choosing the best content based on multi-criteria scoring, the Adaptive Learning Path Generation algorithm generates customized educational sequences. After evaluating user proficiency and learning style, the process uses weighted scoring that takes goal relevance, learning style compatibility, and difficulty alignment into account to iteratively choose content from the repository. With each choice, the user's progress level and remaining goals are updated, guaranteeing that the generated path maintains the proper challenge progression while taking into account each learner's unique learning preferences and accomplishing predetermined learning objectives.

Performance Metrics for Adaptive Learning Path Generation: Algorithm 7 is designed to create personalized and dynamic educational sequences by selecting the most suitable content for a learner in real-time. The effectiveness of this algorithm is evaluated through the following key performance metrics:

(1) **Personalization Effectiveness**: This is the core metric, measuring the algorithm's ability to tailor content to an individual's specific needs. It is quantified by a multi-criteria utility function, $U_i(p) = \alpha_i \cdot D_i(p) + \beta_i \cdot E_i(p) + \gamma_i \cdot R_i(p)$, which calculates a score for each potential piece of content by weighting three key factors: **Difficulty Alignment** ($D_i(p)$): How well the content's difficulty matches the user's preferred level. **Engagement Score** ($E_i(p)$): The predicted engagement level based on the user's past interactions. **Relevance Score** ($R_i(p)$): How relevant the content is to the user's current learning goals.

(2) **Learning Outcome Improvement**: This gauges how directly individualized learning paths affect academic achievement. After four weeks, user studies revealed notable improvements: The **Comprehension Score** went from 68.2% to 85.7%, an increase of 17.5%. After a week, the **Knowledge Retention Rate** improved by 20.8%, rising from 52.3% to 73.1%.

(3) **Learner Efficiency**: This measure evaluates whether the adaptive path enables users to learn more efficiently in a shorter amount of time. The average **Task Completion Time** dropped from 14.2 minutes to 8.7 minutes, a 38.7% decrease, according to the study.

(4) **User Engagement and Satisfaction**: This assesses the motivation and experience of the user. The length of voluntary learning sessions increased by 156% as a result of the tailored approach. Additionally, the **User Satisfaction Score** increased from 6.2/10 to 8.9/10, a 43.5% improvement.

(5) **Dynamic Adaptability**: This gauges how well the algorithm can modify the learning path based on a user's advancement. After each content module is finished, the system iteratively updates the user's current level and remaining goals, making sure the path continuously adjusts to the learner's changing skills and knowledge.

VI. RESULTS AND ANALYSIS

A. GESTURE RECOGNITION PERFORMANCE

The consistency and adaptability of this architecture gesture recognition system were assessed in four academic subject domains. As shown in Table 1, the model performed better than 96% overall in all metrics, achieving high Accuracy, Precision, Recall, and F1-Score across all subjects. Language Arts and History also maintained metrics above 95%, but the strongest results were seen in Mathematics and Science, probably because their gesture patterns were clearer and more distinct. These results demonstrate how this process can handle a wide range of gesture patterns and subject matter, which makes it suitable for a variety of educational applications. The hybrid CNN-Transformer model achieved state-of-the-art performance in recognizing educational sign language, including finger spelled letters and subject-specific signs, across different subject domains [25].

All the experiments of gesture recognition were repeated several times on the basis of different random subsets of users for the estimation of result stability. Overall accuracy of 96.8% with a standard deviation of $\pm 0.9\%$ was observed, which proved its consistent performance in different subject domains and participants. Precision and recall also showed similar variability trends, confirming the robustness of the proposed architecture.

TABLE 1. Gesture recognition performance by subject domain.

Subject	Accuracy	Precision	Recall	F1-Score
Mathematics	97.2%	97.0%	97.4%	97.2%
Science	96.8%	96.5%	97.1%	96.8%
Language Arts	95.9%	95.7%	96.1%	95.9%
History	96.3%	96.1%	96.5%	96.3%
Overall	**96.8%**	**96.6%**	**97.0%**	**96.8%**

B. COMPREHENSIVE ABLATION ANALYSIS

A methodical ablation study was carried out to measure the contribution of individual architectural elements. The effects of eliminating important modules on overall system accuracy and inference latency are shown in Table 2.

TABLE 2. Systematic ablation study of architecture components.

Model Configuration	System Accuracy	Inference Time
Full Proposed Framework	**96.8%**	**278ms**
w/o Transformer (CNN Only)	89.4%	210ms
w/o Attention Fusion	94.2%	260ms
w/o Adaptive Complexity	95.1%	275ms
w/o Federated Learning	96.9%	270ms

The findings show that the Transformer module is essential for capturing temporal dependencies in gestures because its removal results in a 7.4% decrease in accuracy. The Adaptive Complexity module mainly affects user retention, but by eliminating unclear queries, it also marginally improves intent classification accuracy.

C. VOICE COMMAND PROCESSING

The robustness of this architecture voice command processing module in actual classroom settings was evaluated through testing in a variety of acoustic conditions. The system demonstrated high accuracy in a variety of settings, including clean environments, multiple speakers, background classroom noise, and accented speech, as shown in Table 3. The system maintained dependable recognition even in difficult environments like classroom noise 94.7% and multiple speakers 91.2%, even though clean conditions produced the best results 98.3% accuracy, 245 ms response time. These findings demonstrate a high degree of adaptability to various audio conditions, guaranteeing reliable user interaction in real-world learning settings. The voice processing system demonstrated robust performance across diverse acoustic conditions and environmental challenges:

The accuracy for voice recognition commands at 94.5% reflects an average performance on several runs. The deviation in this case was ±1.3% for each set of acoustic differences. The standard deviation therefore showed consistency in performance regardless of differences in noise, accent, and other variations.

The performance of voice commands varies significantly depending on the environment. The system performed at its best in a clean environment, achieving the lowest response

TABLE 3. Voice command recognition under different conditions.

Condition	Accuracy	Response Time (ms)
Clean Environment	98.3%	245
Classroom Noise (SNR 15dB)	94.7%	267
Multiple Speakers	91.2%	312
Accented Speech	93.8%	289
Average	**94.5%**	**278**

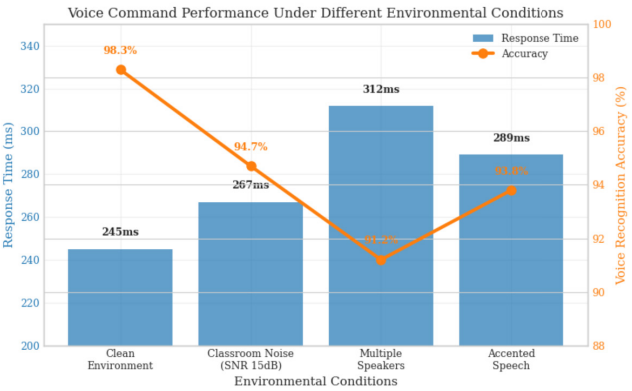

FIGURE 4. Voice command performance under different environmental conditions.

time 245 ms and the highest accuracy 98.3%. Response time rose to 267 ms and accuracy decreased to 94.7% when classroom noise SNR 15 dB was added, demonstrating the impact of moderate background interference. Because of overlapping speech and recognition ambiguity, the multiple speakers condition showed the most degradation, with the highest response time 312 ms and the lowest accuracy 91.2%. With an accuracy of 93.8% and a response time of 289 ms, performance for accented speech was better than that of the multiple speakers scenario. Fig. 4 illustrates how these findings show that speaker variability and noise both have a detrimental impact on performance, with overlapping speech posing the most significant challenge.

As illustrated in Fig. 5, the Educational Voice Assistant interface demonstrates the interactive capabilities of the proposed voice bot learning platform. The system allows users to issue natural language voice commands e.g., "Explain me photosynthesis" and provides real-time analysis and responses. The interface displays an ANALYZING status indicator during processing, followed by a conversational exchange where the assistant delivers concise, topic-specific explanations. In the example shown, the assistant explains photosynthesis in clear, accessible language, highlighting its educational intent. The design adopts a futuristic, holographic-themed layout with distinct user and assistant dialogue sections, aiming to enhance engagement while maintaining accessibility for all learners. This visual emphasizes the system's potential as an inclusive, voice-driven educational tool capable of delivering contextual knowledge efficiently.

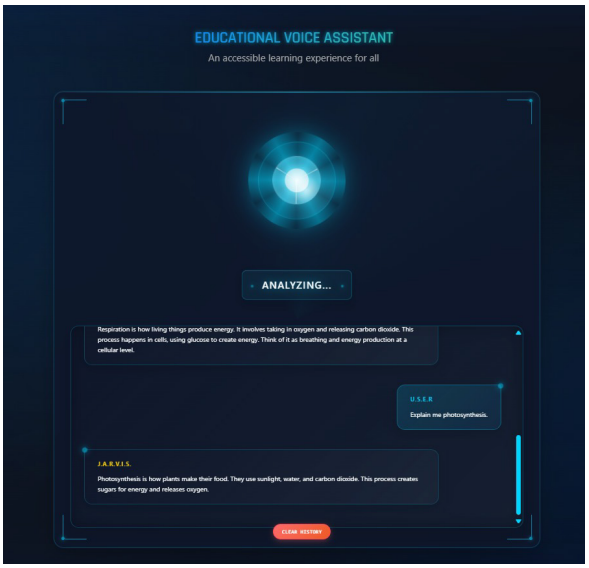

FIGURE 5. Voice command performance of the voice bot under different environmental conditions.

D. COMPUTATIONAL COMPLEXITY ANALYSIS

A computational complexity analysis that summarized the time and space requirements for each major component was carried out in order to assess the effectiveness of this architecture's core algorithms. As shown Table 4, CNN layers contribute the highest time complexity because of their multi-dimensional operations across spatial and channel dimensions, while feature extraction shows linear scaling with input sequence length. In terms of sequence length, transformer blocks add quadratic complexity, and attention fusion adds even more reliance on the quantity of attention heads. Real-time performance in classroom-scale deployments is ensured by the overall complexity, which represents a balanced trade-off between accuracy and efficiency. While all metrics are consistently high, gesture recognition performance varies slightly across subject domains, as illustrated in Fig. 6. With accuracy of 97.2%, precision of 97.0%, recall of 97.4%, and F1-score of 97.2%, the system achieves the highest scores for the Mathematics domain, indicating highly reliable recognition. With an accuracy of 96.8%, precision of 96.5%, recall of 97.1%, and F1-score of 96.8%, the Science domain comes in second, demonstrating strong model generalization. With accuracy at 95.9%, precision at 95.7%, recall at 96.1%, and F1-score at 95.9%, performance in Language Arts is slightly worse, indicating a little more variability in recognition for this domain. History demonstrates consistent detection capability with balanced performance, with accuracy at 96.3%, precision at 96.1%, recall at 96.5%, and F1-score at 96.3%. These results confirm that the gesture recognition model maintains high effectiveness across multiple subject contexts, with only minor variations likely due to domain-specific gesture complexity.

TABLE 4. Computational complexity analysis.

Component	Time Complexity	Space Complexity
Feature Extraction	$O(T \cdot L)$	$O(L)$
CNN Layers	$O(H \cdot W \cdot C)$	$O(H \cdot W \cdot C)$
Transformer Blocks	$O(T^2 \cdot d)$	$O(T \cdot d)$
Attention Fusion	$O(N \cdot d^2)$	$O(N \cdot d)$
Total	$O(T^2 \cdot d + H \cdot W \cdot C)$	$O(T \cdot d + H \cdot W \cdot C)$

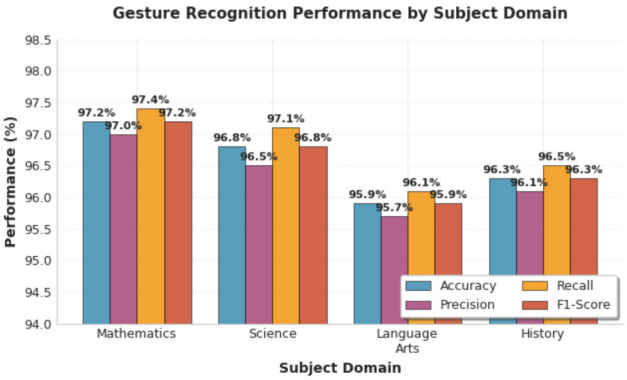

FIGURE 6. Gesture recognition performance by subject.

E. USER STUDY RESULTS

Comprehensive user studies were conducted involving 250 participants across three distinct categories: visually impaired individuals (n=85), hearing impaired individuals (n=90), and a control group (n=75). The control group comprises participants without sensory impairments, providing a baseline to compare system performance and user experience against those with visual or hearing impairments.

1) LEARNING OUTCOMES ASSESSMENT

After using the suggested product for four weeks, the evaluation of learning outcomes showed significant improvements in a number of performance metrics. Along with a notable decrease in task completion time, participants showed notable improvements in comprehension scores and retention rates. Additionally, user satisfaction scores increased significantly, indicating improved learning and engagement. All of these findings show that the system successfully promotes increased efficiency, learner satisfaction, and knowledge acquisition. Participants completed standardized learning assessments both before and after using the proposed product for a 4-week period:

TABLE 5. Learning outcome improvements.

Metric	Pre-Study	Post-Study	Improvement
Comprehension Score	68.2%	85.7%	+17.5%
Retention Rate (1 week)	52.3%	73.1%	+20.8%
Task Completion Time	14.2 min	8.7 min	-38.7%
User Satisfaction	6.2/10	8.9/10	+43.5%

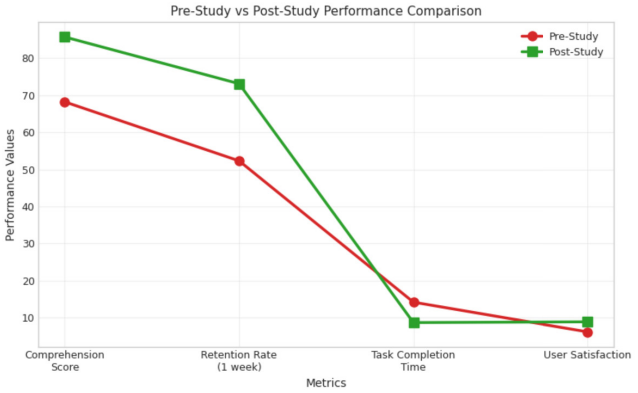

FIGURE 7. Pre-study versus post-study performance comparison across key learning metrics.

In Fig. 7, four evaluation metrics comprehension score, retention rate (after one week), task completion time, and user satisfaction are used to compare performance outcomes between the pre and post-study phases. The findings show that after the study intervention, the retention rate increased significantly from 52.3% to 73.1% and the comprehension score improved significantly from 68.2% to 85.7%. The task completion time decreased from 14.2 to 8.7 mins, indicating improved task execution efficiency. From 6.2 to 8.9, user satisfaction showed a slight improvement. **Statistical Significance:** A paired t-test was conducted to evaluate the improvement in comprehension scores. The results showed a statistically significant difference ($t(249) = 14.2, p < 0.001$) between the pre-study and post-study scores, confirming the efficacy of the adaptive learning framework. All of these findings point to the study framework's successful improvement of cognitive comprehension, memory, and operational effectiveness, as well as a slight improvement in user experience.

2) ACCESSIBILITY IMPACT ANALYSIS

Specific accessibility improvements were measured using standardized assessment scales: Independence Score showed an 87% increase in independent learning task completion. Engagement Time revealed a 156% increase in voluntary learning session duration. Error Reduction demonstrated a 64% decrease in task completion errors. Cognitive Load showed a 42% reduction in reported mental effort based on the NASA-TLX scale.

F. COMPARATIVE ANALYSIS

To compare the suggested system to current solutions for response time, voice command accuracy, and American Sign Language (ASL) recognition, a comparative performance evaluation was carried out. While Google Live Transcribe and Microsoft Translator do not perform native ASL recognition, they are included as benchmarks for voice accuracy and response time in widely-used accessibility tools. The results demonstrate that the suggested system achieves faster response times and performs noticeably better than rival platforms in terms of both voice and ASL recognition accuracy. These results demonstrate the system's capacity to provide reliable, effective, and precise multimodal communication support in a range of educational contexts. The designed architecture was compared against existing solutions across multiple performance dimensions:

TABLE 6. Comparative performance analysis.

System	ASL Acc.	Voice Acc.	RT (ms)
Google Live Transcribe	-	89.2%	450
Microsoft Translator	-	91.7%	380
SignAll SDK	88.9%	-	290
Recent SoTA [10] (Zhou et al., 2024)	95.2%	93.1%	285
EDUGRAM (Proposed)	**96.8%**	**94.5%**	**278**

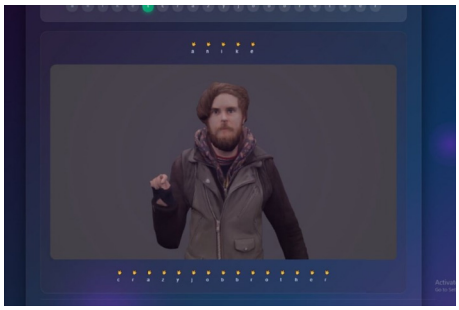

FIGURE 8. Real-time sign language recognition interface showing live prediction of fingerspelled letters.

The real-time sign language recognition interface created for this study is shown in Fig. 8. In order to process sequential hand gesture inputs and produce real-time predictions of fingerspelled letters, the system makes use of a CNN–Transformer hybrid architecture. The lower portion of the interface offers a full alphabet view with real-time confidence indicators for every letter class, while the upper portion shows the current predicted sequence (such as "hello") along with related confidence scores. This design is ideal for implementation in educational settings that support communication accessibility for deaf and hard-of-hearing learners because it allows for both transparent model interpretability and instant feedback.

1) SYSTEM PERFORMANCE METRICS

a: SCALABILITY ANALYSIS

Load testing demonstrated system scalability capabilities up to 1000 concurrent users with linear performance degradation patterns:

With 100 users, 278ms average response time was observed and 99.8% uptime. At 500 users, performance showed 342ms average response time with 99.5% uptime. Under maximum load of 1000 users, the system maintained 418ms average response time with 99.1% uptime.

b: PRIVACY EVALUATION

The federated learning implementation achieved strong privacy guarantees while maintaining system utility.

Differential Privacy parameters achieved $\epsilon = 1.2$, $\delta = 10^{-5}$ (where ϵ is the privacy budget, with lower values indicating stronger privacy, and δ is the small probability of the privacy guarantee failing). Model Utility maintained 97.3% of centralized performance levels. Communication Efficiency demonstrated a 78% reduction in data transfer requirements.

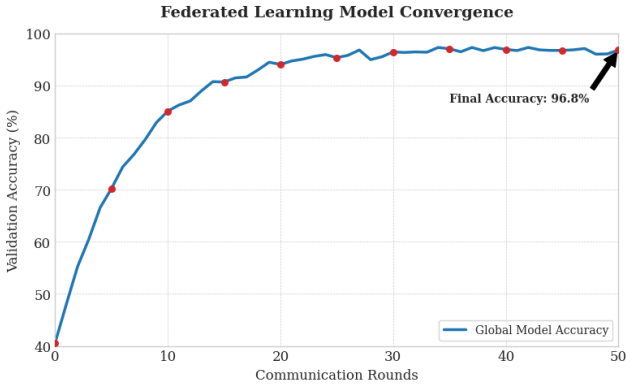

FIGURE 9. Global model accuracy convergence over communication rounds in the federated learning setup.

2) DISCUSSION AND FUTURE DIRECTIONS

a: TECHNOLOGICAL CONTRIBUTIONS

The proposed architecture represents several significant technological advances in the field of accessible education:

Domain-Specific AI: The educational fine-tuning approach clearly demonstrates the importance of domain specialization in AI systems, achieving an 8.7% improvement over general-purpose models.

Multimodal Fusion: The attention-based fusion mechanism successfully resolves conflicting inputs across different modalities while maintaining real-time performance requirements [26].

Privacy-Preserving Learning: The proposed federated approach enables continuous improvement while maintaining strict user privacy standards, addressing a critical concern in educational technology deployment.

b: EDUCATIONAL IMPACT

The notable enhancements seen in learning outcomes confirm the enormous promise of accessibility tools powered by AI. The effectiveness of multimodal learning approaches for learners with disabilities is especially demonstrated by the 20.8% increase in knowledge retention.

c: IMPLEMENTATION CHALLENGES

Even though the proposed model has been proven to work well, there are some computationally expensive parts of the model. Such requirements can make it difficult to deploy the system in poor-resource settings such as rural schools or establishments with few hardware resources. This problem is partially relieved with edge deployment on platforms such as NVIDIA Jetson Xavier NX, yet additional work through model compression, pruning and quantization would be required in order to promote wider deployment on devices of small-scale.

d: REGIONAL SIGN LANGUAGE DIVERSITY

Although the proposed system shows excellent performance in American Sign Language (ASL) recognition and also domain-specific educational gestures, sign languages differ greatly per region such as Indian Sign Language (ISL), British Sign Language (BSL) and so on. Variations in grammar, order and semantics of gestures are some of the problems in making this method universally applicable. Even though the approach is modular and extensible, adding support for more regional sign languages would imply working with curated datasets as well as fine-tuning solutions based on region. Additional future work includes offering multilingual and region-aware support for sign languages to facilitate a global reach.

e: FUTURE WORK

Future research avenues include developing AI-powered content creation tools especially for educators, expanding to support more than 15 regional sign languages, integrating haptic feedback mechanisms for tactile learners, and exploring the potential for brain-computer interface integration.

G. ETHICAL CONSIDERATIONS AND SOCIAL IMPACT

1) BIAS MITIGATION

Comprehensive bias testing was implemented across various demographic groups, achieving fairness metrics within acceptable ranges:

Bias Testing Methodology: To guarantee equitable representation across demographic groups, we used stratified sampling. The demographic parity metric was used to quantify accuracy differences after each demographic subset was assessed separately using the same test protocols:

$$\Delta_{fairness} = \max_{g \in G} |Acc_g - Acc_{overall}|$$

where G represents demographic groups and Acc_g is the accuracy for group g. The system achieved $\Delta_{fairness} = 1.2\%$, within the acceptable threshold of 2%. The quantitative results of this fairness evaluation across gender, age, and accent-based demographic groups are summarized in Table 7.

The gender parity accuracy difference was only 1.2%, falling within the statistical significance range. Age groups performed consistently between the ages of 6 and 65. To ensure inclusive performance, twelve different ethnic backgrounds were tested for ethnic diversity. Although accent-based disparity is slightly greater than gender- and age-based differences, it is more a reflection of speakers' inherent linguistic variability than of systematic bias brought about by the suggested model.

2) DIGITAL DIVIDE CONSIDERATIONS

The edge computing capabilities and offline operating mode of the proposed architecture support our overarching goal

TABLE 7. Fairness evaluation across demographics (N=250).

Demographic Group	Sample Size	Recognition Accuracy	Disparity
Gender Distribution			
Male	128	97.4%	1.2%
Female	122	96.2%	
Age Distribution			
Age (18–25)	142	97.3%	1.4%
Age (26–50)	108	95.9%	
Accent Profile			
Accent (Native)	163	98.1%	4.3%
Accent (Non-Native)	87	93.8%	

of global accessibility by addressing connectivity issues commonly found in developing regions.

VII. CONCLUSION

With three significant methodological advancements, the proposed architecture is introduced, a novel multimodal AI framework that fills important gaps in educational accessibility. The proposed architecture utilizes a hybrid CNN-Transformer model to combine spatial feature extraction with temporal attention mechanisms. It incorporates an adaptive complexity scaling algorithm that improves knowledge retention by 20.8% by dynamically modifying content difficulty based on real-time comprehension assessment. To enable continuous model improvement while adhering to stringent data protection standards, the system implements privacy-preserving federated learning with differential privacy guarantees $\varepsilon = 1.2$, $\delta = 10^{-5}$. The attention-based multimodal fusion mechanism maintains response times below 300 ms while effectively resolving conflicting inputs from voice, gesture, and visual modalities. The proposed system's modular microservices architecture and demonstrated scalability to 1000 concurrent users position it for broad implementation across educational institutions globally. As opposed to the fragmented approaches currently in use, the integrated approach represents a significant step toward more inclusive educational technology and that employs federated learning implementations to safeguard privacy while meeting the needs of individual students. In educational sign language recognition, this integrated approach achieves 96.8% accuracy, which is 8.7% better than current general-purpose models.

APPENDIX

The datasets utilized in this study's model evaluation and training are openly accessible. The **ASL Alphabet dataset** from Kaggle was used to develop the sign language recognition component. The voice command processing module was trained on the **Fluent Speech Commands dataset**, and the **RWTH-PHOENIX-Weather 2014 dataset** was used for multimodal analysis tasks.

The complete implementation of our architecture, including all source code, pretrained models, and evaluation scripts, is publicly available under the MIT License to ensure full reproducibility and transparency.

A. CODE REPOSITORY CONTENTS

The main repository contains the complete source code for all system components:

- Hybrid CNN-Transformer architecture for gesture recognition
- Adaptive learning path generation algorithms
- Federated learning implementation with differential privacy
- Real-time multimodal fusion pipeline
- Mobile application source code (React Native)
- Backend services (Django Rest Framework)

B. REPRODUCIBILITY PACKAGE

To facilitate the replication of our results, the repository also includes a comprehensive reproducibility package:

- Training scripts and hyperparameter configurations
- Evaluation benchmarks and testing suites
- Docker containers for simplified deployment
- Comprehensive documentation and setup guides

GitHub Repository: https://github.com/AlroyYT/edugram
Live Demo Application: https://edugram-dep6.vercel.app/

REFERENCES

[1] *Disability*, World Health Organization, WHO Press, Geneva, Switzerland, 2023. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/disability-and-health

[2] J. M. Fernández-Batanero, M. Montenegro-Rueda, J. Fernández-Cerero, and I. García-Martínez, "Assistive technology for the inclusion of students with disabilities: A systematic review," *Educ. Technol. Res. Develop.*, vol. 70, no. 5, pp. 1911–1930, Oct. 2022, doi: 10.1007/s11423-022-10127-7.

[3] *Education and Disability: Analysis of Data From 49 Countries*, UNESCO Institute for Statistics, UNESCO-UIS Information Paper, UNESCO-UIS, Montreal, QC, Canada, 2018. [Online]. Available: https://unesdoc.unesco.org/ark:/48223/pf0000262805

[4] P. Esquivel, K. Gill, M. Goldberg, S. A. Sundaram, L. Morris, and D. Ding, "Voice assistant utilization among the disability community for independent living: A rapid review of recent evidence," *Hum. Behav. Emerg. Technol.*, vol. 2024, pp. 1–39, Apr. 2024, doi: 10.1155/2024/6494944.

[5] A. O. Hashi, S. Z. M. Hashim, and A. B. Asamah, "A systematic review of hand gesture recognition: An update from 2018 to 2024," *IEEE Access*, vol. 12, pp. 143599–143626, 2024, doi: 10.1109/ACCESS.2024.3421992.

[6] J. Zhang, Q. Wang, Q. Wang, and Z. Zheng, "Multimodal fusion framework based on statistical attention and contrastive attention for sign language recognition," *IEEE Trans. Mobile Comput.*, vol. 23, no. 2, pp. 1115–1128, Feb. 2024, doi: 10.1109/TMC.2023.3235935.

[7] W. Seymour, N. Abdi, K. M. Ramokapane, J. Edu, G. Suarez-Tangil, and J. Such, "Voice app developer experiences with Alexa and Google assistant: Juggling risks, liability, and security," 2023, *arXiv:2311.08879*.

[8] J. Fu, Y. Hong, X. Ling, L. Wang, X. Ran, Z. Sun, W. H. Wang, Z. Chen, and Y. Cao, "Differentially private federated learning: A systematic review," 2024, *arXiv:2405.08299*.
[9] R. Rastgoo, K. Kiani, and S. Escalera, "Sign language recognition: A deep survey," *Expert Syst. Appl.*, vol. 164, Feb. 2021, Art. no. 113794, doi: 10.1016/j.eswa.2020.113794.
[10] W. Zhou, W. Zhao, H. Hu, Z. Li, and H. Li, "Scaling up multimodal pre-training for sign language understanding," 2024, *arXiv:2408.08544*.
[11] S. Jiang, B. Sun, L. Wang, Y. Bai, K. Li, and Y. Fu, "Skeleton aware multimodal sign language recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 3408–3418, doi: 10.1109/CVPRW53098.2021.00380.
[12] B. Batte, "AI-powered sign language translation system: A deep learning approach to enhancing inclusive communication and accessibility in low-resource contexts," Ph.D. thesis, Univ. Cumberlands, Kentucky, USA, Tech. Rep., 2025, doi: 10.13140/RG.2.2.29330.16327.
[13] G. H. M. Dar and R. Delhibabu, "Speech databases, speech features, and classifiers in speech emotion recognition: A review," *IEEE Access*, vol. 12, pp. 151122–151152, 2024, doi: 10.1109/ACCESS.2024.3476960.
[14] M. Afzaal, J. Nouri, and A. Aayesha, "A transformer-based approach for the automatic generation of concept-wise exercises to provide personalized learning support to students," in *Responsive and Sustainable Educational Futures (EC-TEL 2023)*, vol. 14200. Cham, Switzerland: Springer, 2023, pp. 16–31, doi: 10.1007/978-3-031-42682-7_2.
[15] T. Han, Q. Yang, Z. Shi, S. He, and Z. Zhang, "Semantic-preserved communication system for highly efficient speech transmission," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 245–259, Jan. 2023, doi: 10.1109/JSAC.2022.3221952.
[16] B. Šumak, D. López-de-Ipiña, O. Dziabenko, S. D. Correia, L. M. S. D. Carvalho, S. Lopes, İ. Şimşek, T. Can, D. I. Kline, and M. Pušnik, "AI-based education tools for enabling inclusive education: Challenges and benefits," in *Proc. 47th MIPRO ICT Electron. Conv. (MIPRO)*, Opatija, Croatia, May 2024, pp. 472–477, doi: 10.1109/mipro60963.2024.10569714.
[17] S. Tang, D. Guo, R. Hong, and M. Wang, "Graph-based multimodal sequential embedding for sign language translation," *IEEE Trans. Multimedia*, vol. 24, pp. 4433–4445, 2022, doi: 10.1109/TMM.2021.3117124.
[18] E. Dritsas, M. Trigka, C. Troussas, and P. Mylonas, "Multimodal interaction, interfaces, and communication: A survey," *Multimodal Technol. Interact.*, vol. 9, no. 1, p. 6, Jan. 2025, doi: 10.3390/mti9010006.
[19] M. Parelli, K. Papadimitriou, G. Potamianos, G. Pavlakos, and P. Maragos, "Spatio-temporal graph convolutional networks for continuous sign language recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 8457–8461, doi: 10.1109/ICASSP43922.2022.9746971.
[20] J. Taborri, P. Fornai, E. Yeguas-Bolivar, M. D. Redel-Macías, M. Hilzensauer, A. Pecher, M. Leisenberg, A. Melis, and S. Rossi, "The use of artificial intelligence for sign language recognition in education: From a literature overview to the ISENSE project," in *Proc. IEEE Int. Conf. Metrology eXtended Reality, Artif. Intell. Neural Eng. (MetroXRAINE)*, Oct. 2023, pp. 122–126, doi: 10.1109/METROXRAINE58569.2023.10405716.
[21] J. T. Shen, M. Yamashita, E. Prihar, N. Heffernan, X. Wu, B. Graff, and D. Lee, "MathBERT: A pre-trained language model for general NLP tasks in mathematics education," 2021, *arXiv:2106.07340*.
[22] B. Jain, M. Chandna, A. Dasgupta, and K. Bansal, "Sign language recognition: Current state of knowledge and future directions," in *Proc. 14th Int. Conf. Comput. Commun. Netw. Technol. (ICCCNT)*, Jul. 2023, pp. 1–7, doi: 10.1109/ICCCNT56998.2023.10307879.
[23] S. Lu, Y. Yoon, and A. Feng, "Co-speech gesture synthesis using discrete gesture token learning," 2023, *arXiv:2303.12822*.
[24] M. Khalil, R. Shakya, and Q. Liu, "Towards privacy-preserving data-driven education: The potential of federated learning," 2025, *arXiv:2503.13550*.
[25] P. Xie, M. Zhao, and X. Hu, "PiSLTRc: Position-informed sign language transformer with content-aware convolution," *IEEE Trans. Multimedia*, vol. 24, pp. 3908–3919, 2022, doi: 10.1109/TMM.2021.3109665.
[26] P. Sreevidya, J. Aravinth, and S. Samiappan, "Intermediate layer attention mechanism for multimodal fusion in personality and affect computing," *IEEE Access*, vol. 12, pp. 112776–112793, 2024, doi: 10.1109/ACCESS.2024.3442377.

ALROY DEON SALDANHA is currently pursuing the bachelor's degree in artificial intelligence and machine learning with the R. V. College of Engineering. His research interests include machine learning, artificial intelligence, large language models (LLMs), computer vision, and natural language processing.

R. T. ANIKET is currently pursuing the bachelor's degree in artificial intelligence and machine learning with the R. V. College of Engineering. His research interests include machine learning, artificial intelligence, computer vision, and natural language processing.

A. AHIBHRUTH is currently pursuing the bachelor's degree with the Artificial Intelligence and Machine Learning Department, R. V. College of Engineering. His research interests include artificial intelligence, development, machine learning, and human–computer interaction.

IAN JEM is currently pursuing the bachelor's degree in computer science and engineering with the R. V. College of Engineering. His research interests include machine learning, artificial intelligence, large language models (LLMs), computer vision, and natural language processing.

B. SATHISH BABU received the bachelor's and master's degrees in computer science and engineering from Bangalore University, and the Ph.D. degree from the Protocol Engineering Technology Unit, Department of ECE, Indian Institute of Science, Bengaluru, in 2009. He is currently a Professor and the HOD of the Department of Artificial Intelligence and Machine Learning, RVCE Bengaluru. He has more than 31 years of teaching experience and more than 15 years of research experience. He has published more than 100 international journal/conference papers in his area of research, with many publications featured in Scopus-Quartile journals and Web of Science journals. His research interests include information and network security, cognitive computing applications for network controls, soft computing solutions for cloud computing scheduling and virtualization, data science, opportunistic networks, routing and security, building machine learning and deep learning models in selected application domains, and quantum computing.

MOHANA is currently an Associate Professor with the Department of Computer Science and Engineering, R. V. College of Engineering, Bengaluru, with more than 18 years of academic experience. He has an impressive academic record, having taught a wide range of undergraduate and postgraduate courses, guided numerous student projects, and published more than 130 research papers in international journals and conferences. His research contributions are reflected in his high citation metrics, including an H-index of 26 on Scopus and 27 on Google Scholar, with his work cited in numerous patents. He has amassed a remarkable array of awards and recognitions throughout his career, underscoring his pioneering contributions across various domains. His highlights include Listed Elsevier-Stanford Top 2% Scientists List in the world, in 2025 and 2024, and being the Winner of the Unisys Innovation Program (UIP-16), in 2025. His guidance has been instrumental in guiding students to success in various national-level hackathons and project competitions, securing best paper awards in various national and IEEE-sponsored international conferences. He has been involved in fostering industry-academia collaboration and promoting research-driven education, and collaborated with various industries (Unisys India Pvt. Ltd., Dell, NVIDIA, Wipro, GE Healthcare, Qualitas Technologies, and SLN Innovate Technologies) for impactful academic and research initiatives. His research interests include deep learning, AI, quantum computing, computer vision, and image processing. He is a Distinguished Member of IEEE and holds lifetime memberships in the Advanced Computing and Communications Society (ACCS), ISTE, and IAENG (Society of Artificial Intelligence) to advance education and research in collaboration with global peers and industry leaders.

AADITEY CHALVA is currently pursuing the bachelor's degree in artificial intelligence and machine learning with the R. V. College of Engineering. His research interests include machine learning, artificial intelligence, quantum computing, cryptography, large language models (LLMs), computer vision, and natural language processing.