Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.062695

ARTICLE

ALCTS—An Assistive Learning and Communicative Tool for Speech and Hearing Impaired Students

Shabana Ziyad Puthu Vedu[1,*], Wafaa A. Ghonaim[2], Naglaa M. Mostafa[3] and Pradeep Kumar Singh[4]

[1]Computer Science Department, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Al Kharj, 11942, Saudi Arabia
[2]Faculty of Science, Al-Azhar University, Cairo, 12111, Egypt
[3]Department of Mathematics, Faculty of Science, Al-Azhar University (Girl's Branch), Cairo, 12111, Egypt
[4]Department of Computer Science and Engineering, Central University of Jammu, Jammu and Kashmir, 181143, India

*Corresponding Author: Shabana Ziyad Puthu Vedu. Email: s.ziyad@psau.edu.sa

Received: 25 December 2024; Accepted: 21 February 2025; Published: 16 April 2025

ABSTRACT: Hearing and Speech impairment can be congenital or acquired. Hearing and speech-impaired students often hesitate to pursue higher education in reputable institutions due to their challenges. However, the development of automated assistive learning tools within the educational field has empowered disabled students to pursue higher education in any field of study. Assistive learning devices enable students to access institutional resources and facilities fully. The proposed assistive learning and communication tool allows hearing and speech-impaired students to interact productively with their teachers and classmates. This tool converts the audio signals into sign language videos for the speech and hearing-impaired to follow and converts the sign language to text format for the teachers to follow. This educational tool for the speech and hearing-impaired is implemented by customized deep learning models such as Convolution neural networks (CNN), Residual neural Networks (ResNet), and stacked Long short-term memory (LSTM) network models. This assistive learning tool is a novel framework that interprets the static and dynamic gesture actions in American Sign Language (ASL). Such communicative tools empower the speech and hearing impaired to communicate effectively in a classroom environment and foster inclusivity. Customized deep learning models were developed and experimentally evaluated with the standard performance metrics. The model exhibits an accuracy of 99.7% for all static gesture classification and 99% for specific vocabulary of gesture action words. This two-way communicative and educational tool encourages social inclusion and a promising career for disabled students.

KEYWORDS: Sign language recognition system; ASL; dynamic gestures; facial key points; CNN; LSTM; ResNet

1 Introduction

Communication is the most powerful means of sharing and knowing each other's views. The exchange of thoughts, ideas, and emotions among individuals always relies on effective communication. However, speech and hearing-impaired children face a constant struggle in communicating with their classmates and teachers in school. According to World Health Organization statistics, 5% of the world's population, around 466 million, suffer from partial or total hearing impairment. Among them, 432 million are adults and 34 million are children. By 2050, 2.5 billion people, one in every four individuals, are at risk of hearing impairment [1]. Speech and hearing-impaired children often face social isolation due to their lack of communicative competence. Sign language is a visual and gestural medium used by speech and hearing-impaired people to communicate with each other. The American Sign Language (ASL) includes

vocabulary and grammar communicated through hand gestures and facial expressions [2]. Children with speech and hearing impairments born to disabled parents benefit from learning sign language at an early age. However, parents without prior knowledge of sign language undergo special training to communicate effectively with their children. Research shows that early communication through sign language develops strong cognitive skills in a child. While sign language is adequate for communication among the disabled, it is useless when communicating with a disabled person [3]. The lack of sign language interpreters to interpret lectures in the classroom and standard mode of education further complicates the learning process for these students, depriving them of quality higher education. The lack of sign language interpreters to bridge this communication gap hinders talented students from pursuing higher studies in universities of their choice. Therefore, speech and hearing-impaired students attend special schools where sign language is the standard mode of education. A research study shows that in an educational institution, 60% of interpreters for the hearing impaired are incompetent to interpret and explain the curriculum [4]. Children with speech and hearing impairments are deprived of higher education in reputable universities. This scenario emphasizes the need to develop an automated two-way communication Artificial Intelligence (AI) tool to ensure effective learning for the hearing impaired. This research study focuses on building an AI-based sign language recognition tool that converts audio signals to sign language video. The live sign language video stream from the speech and hearing-impaired is converted to text format and displayed to the viewer. This research study is a novel framework for American Sign Language (ASL) recognition and translation for static and dynamic gestures. This assistive learning and communication tool for speech and hearing-impaired students (ALCTS) allows the disabled to communicate effectively in a classroom environment.

- Easy to use real-time two-way learning assistive and communication tool for hearing and speech-impaired students.
- The ASL alphabet classification model shows 99% with a custom-built CNN model.
- The developed custom-built ANN model extracted the key points of the face and hands.
- The dynamic gesture classification module showed 99% accuracy with the stacked LSTM model.

2 Literature Survey

Vision-based sign language recognition systems include image acquisition, preprocessing, segmentation, feature extraction, and classification [5]. The image acquisition devices used in sign language recognition systems include the camera, webcam, data glove, Kinect, and leap motion controller. Classifiers such as Support Vector Machine (SVM), k-nearest neighbor, Hidden Markov Model, and Random Forest are machine-learning classification models for sign language recognition. Sign languages include American Sign Language (ASL), Arabic Sign Language (ArSL), Brazilian Sign Language (Libras), British Sign Language (BSL) and German Sign Language (DGS). ASL includes three types of datasets: static, continuous, and isolated. Massey Dataset is static, RWTHBOSTON-104 and RWTHBOSTON-400 are continuous and isolated, and Purdue RVL-SLLL Database is static and isolated [6]. The study proposed a model for hand gesture recognition in ASL with a feature-based algorithm by generating a list of hand gesture features [7]. The Oriented FAST and Rotated BRIEF feature detection (ORB) technique identifies patches from the image and generates vectors for the patches. The classifiers Support Vector Machine, Naïve Bayes, Logistic Regression, K Nearest Neighbor, Random Forest, and Multi-Layer Perceptron are analyzed for classification performance. ORB with Multi-Layer Perceptron exhibits an accuracy of 96.96% [8]. The model implements a glove-embedded gesture classifier where the data is collected from the fingertips. The system can detect the alphabets from French Sign Language with an accuracy score of 92%. SVM, Naive Bayes (NB), Multi-Layer Perceptron model, and Random Forest (RF) classifiers were evaluated and experimental results showed RF with 15 trees was the best classifier [9]. Technological advances have replaced machine learning algorithms

with deep learning algorithms in every domain. Region Proposal Network (RPN) predicts object bounds and their scores at each position. Long Short-Term Memory Network (LSTM) and Gated Recurrent Unit (GRU) recognize signs from the ISL video frames. The model achieves 97% accuracy on 11 signs with a single layer of LSTM and GRU. The model achieves 97% accuracy on 11 signs with a single layer of LSTM and GRU [10]. RPNs generate multiple region proposals that are given as input to the Fast R-CNN for detecting objects [11]. The Kinect system and leap motion controller track hands with a classification accuracy of 82.8% [12]. A multi-headed CNN model has been developed in a study to mitigate overfitting to give an impressive 98.98% test accuracy [13]. A framework with customized AlexNet and a modified Visual Geometry Group 16 (VGG16) model for feature extraction with classification tasks using a multiclass SVM for sign language recognition exhibited an accuracy of 99.82% [14]. A recent study introduces the Multi-Scale and Dual Sign Language Recognition Network (SLR-Net), leveraging a graph convolution network (GCN) on RGB videos. The SLR-Net with Multi-Scale Attention Network (MSA), Multi-Scale Spatiotemporal Attention Network (MSSTA), and Attention Enhanced Temporal Convolution Network (ATCN) modules demonstrated an accuracy of 98.08% on the CSL-500 dataset [15]. An improved selective kernel network-temporal convolutional (SKResNet-TCN) network-based sign language video recognition model was developed to enhance the feature extraction ability at a reduced cost [16]. ShuffleNetv2-YOLOv3 makes the network light and enhances the recognition speed, giving an F1-score of 99.1% [17]. The proposed methodology shows 99.7% accuracy for ASL alphabet recognition and 99% for gesture action classification.

The methodologies discussed in Table 1 extract features from static and dynamic gestures of different sign languages and are classified with neural network models. In the previous research study [9], data gloves were used for hand movement detection. Using data gloves every day for communication is not feasible for the disabled. The research study [10] for the ISL dataset is developed in a control-free environment. The number of samples for each gesture word tested is limited, and the author's future work is to build a larger dataset with larger samples. In the research study [18], the dataset used is static gesture images. The research study [15] discussed in the table is designed for Chinese sign languages and not ASL. The research study [16] focuses on isolated word recognition in an Argentinian data set. The state-of-the-art tools are just one-way communication tools that detect the sign language gestures made by the disabled. The proposed assistive tool is a two-way communication tool for students to communicate with teachers in a classroom environment built with deep learning techniques. The proposed model extracts key points from the hand and face movements to classify the dynamic gestures. The speech given by the teacher in the classroom is converted into a sign language video for the disabled to view. Such a two-way assistive learning and communicative tool eliminates the need for an intermediate human interpreter and, therefore, ensures continuous real-time communication. The proposed model shows high accuracy for ASL classification compared to these existing models. The novelty of this research study is the two-way communication tool that can recognize static and dynamic gestures.

Table 1: State-of-the-art methodologies

Authors	Challenges addressed	Technique	Accuracy	Disadvantages
Mummadi et al. [9]	On-board computation & Sensor fusion approach	Random Forest	92%	Data gloves are not suitable to use on a daily basis

(Continued)

Table 1 (continued)

Authors	Challenges addressed	Technique	Accuracy	Disadvantages
Kothadiya et al. [10]	Real-time application	LSTM and GRU	97%	Limited samples in self-collected dataset
Barbhuiya [14]	Computation power	Alexnet and Modified VGG16	98.8%	Identifies only Static gestures
Meng [15]	Complex spatiotemporal dependencies	Graph Neural Network	99.82%	Chinese dataset
Xu [16]	Feature extraction issues	SKResNet-TCN	100%	Argentinian dataset for isolated words
Sun et al. [17]	Lightweight environment	ShuffleNetv2-YOLOv3	F1-score–99.1%	Addresses static sign language

3 Background

The contribution of Artificial Intelligence to health care is remarkable and significant. Computer-aided detection and diagnosis, targeted treatment, remote patient monitoring, automated medical image diagnosis, robot-assisted surgery, medical assistive devices for disabled people, and patient data management are some of the valuable contributions of Artificial Intelligence to health care. The advent of sensors, deep learning techniques, machine learning algorithms, sensor data processing units, Cloud storage, and ubiquitous internet connectivity led to the emergence of computer-aided diagnostic tools [18]. Versatile machine learning and deep learning algorithms for signal processing, image processing, and text processing have paved the way for recently available aided tools for the disabled. AI-based sign language recognition tools provide valuable support to the speech and hearing impaired in overcoming their communication challenges, thereby facilitating social inclusion. Sign language recognition includes sign language representation and translation. The sign language recognition model is either video-based or sensor-based. Sensor-based sign language recognition tool is implemented by acquiring data from wearable and visual sensors, followed by data processing with robust machine learning algorithms [19]. The sign language recognition system leverages sophisticated sensors, such as wearable and visual sensors. Sensors measure the inertial measurement unit and electromyography signals to detect arm movement [20]. Cameras can record valuable information about sign language gestures and interpret them more accurately than wearable sensors. Artificial intelligence plays a significant role in sign language recognition and translation to ensure effective communication for the speech and hearing impaired [21]. Despite its significance, classic methods of sign language recognition deal with issues such as redundant information, finger occlusion, motion blur, and diverse signing styles that drastically impact model accuracy. Gestures are non-verbal means of communication [22]. Hand shape, gesture position, lip movement, and emotions shown are key factors for gesture recognition. Static gestures are identified by the shape of the hand, and dynamic gesture actions by the movement of the hand. The recognition of the gestures determines the words and the context in which they are used in sign language. The proposed methodology converts static gestures and gesture actions shown in ASL to text format for the viewer.

4 Methodology

The proposed ALCTS model is a two-way assistive learning and communication tool for the hearing impaired to communicate in the classroom. The proposed model aims to ease the communication between disabled students and teachers in a classroom environment. Speech is converted to sign language for the disabled, and the sign language shown by the disabled is converted to text format for the teacher to grasp easily. This ALCTS model is a breakthrough for the disabled to pursue higher education in universities with confidence. The lectures given by the teachers in the classroom are streamed as audio files. The acoustic signals in the audio files are converted to text. The text is broken down into words. The words are classified as static gesture words or gesture actions. The static gesture words are split into letters, and each letter is mapped to the alphabet of ASL sign language. Each letter or alphabet identified in the text is converted to the ASL sign language images. A stack of 2D ASL sign language images representing a single word is converted into video for the hearing disabled. The gesture actions are mapped to the gesture videos in the database. The stack of videos, including ASL static words and gesture action words, is displayed to the student. The proposed sign language recognition model captures the signs with the camera and streams them live to the sign language recognition module. The model converts the input video stream into multiple image frames, and each frame is matched with letters in the ASL dataset.

The CNN model identifies the static gesture images and maps them to the alphabet. The alphabets are now grouped into words. The words are merged into sentences. The video stream includes gesture videos that are recognized with a stacked LSTM classifier model to detect actions as words. The combination of words detected by CNN and LSTM models is converted to the text format and displayed to the viewer. The viewer could be the teacher or friend in the classroom unfamiliar with sign language. Classrooms should be equipped with appropriate devices that allow the disabled to communicate with their teachers and classmates. This ensures greater participation of students in professional and social setups.

4.1 Proposed Methodology for Speech to ASL Video Conversion Model

This section discusses the proposed methodology to convert speech to sign language video format. The model shown in Fig. 1 converts speech to ASL sign language video. In this model, Whisper open AI converts the audio-to-text format [23]. Whisper is an Automatic Speech Recognition (ASR) system developed by OpenAI. ASR systems are designed to convert spoken language into written text. The text is split into words for conversion to sign language. The words in the text are foremost classified as static gesture words or gesture action words. The static gesture words are converted to video, containing a stack of ASL sign images corresponding to each English alphabet. The ASL sign image frames stored in a sequence in a folder are later compiled into a video to avoid misspelling the words. The video writer module can generate videos based on the user speed. The dynamic gesture words are mapped to the corresponding video related to these words from the database. The videos of the static gestures and gesture action words are merged in the correct sequence to be displayed to the disabled. The proposed model is implemented in Python. The labeled dataset for the ASL sign language is publicly available.

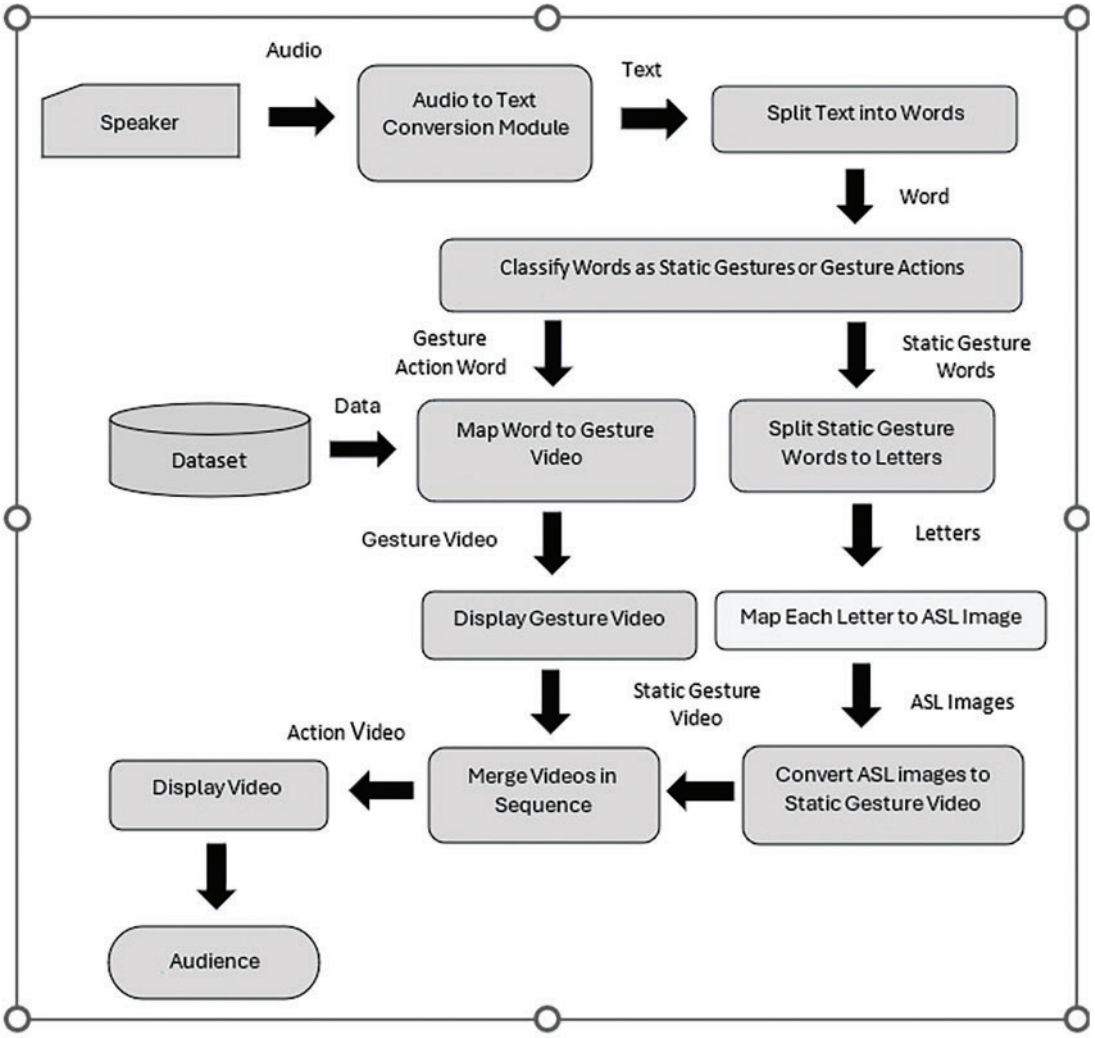

Figure 1: Proposed speech to ASL conversion model for speech and hearing impaired

4.1.1. Algorithm for Converting Speech to Sign Language Video

The algorithm for converting audio to sign language is given in this section. Fig. 2 shows the algorithm for converting audio to sign language videos in the speech-to-video conversion model.

Step 1: The Whisper AI module converts the audio file from speaker to text.

Step 2: Split the text into words and save each word in a string variable S with length n.

Step 3: If the word S_i is a dynamic gesture word, then

Retrieve the corresponding gesture Vdg_i from the database.

Save Vdg_i in the folder V_f

else

Split the word S_i into letters

for each letter L_j in S_i

Retrieve the ASL sign image I_j matching the letter L_j from the ASL dataset.

Store the image I_j in the folder V_i.

Create video Vsg_i from folder V_i with OpenCV VideoWriter() with a time delay.

Save Vsg_i it in a folderV_f.

Step 4: Merge gesture action videos Vdg_i and the static gesture videos Vsg_i to generate a V_o clip.

Step 5: Display the video clip V_o.

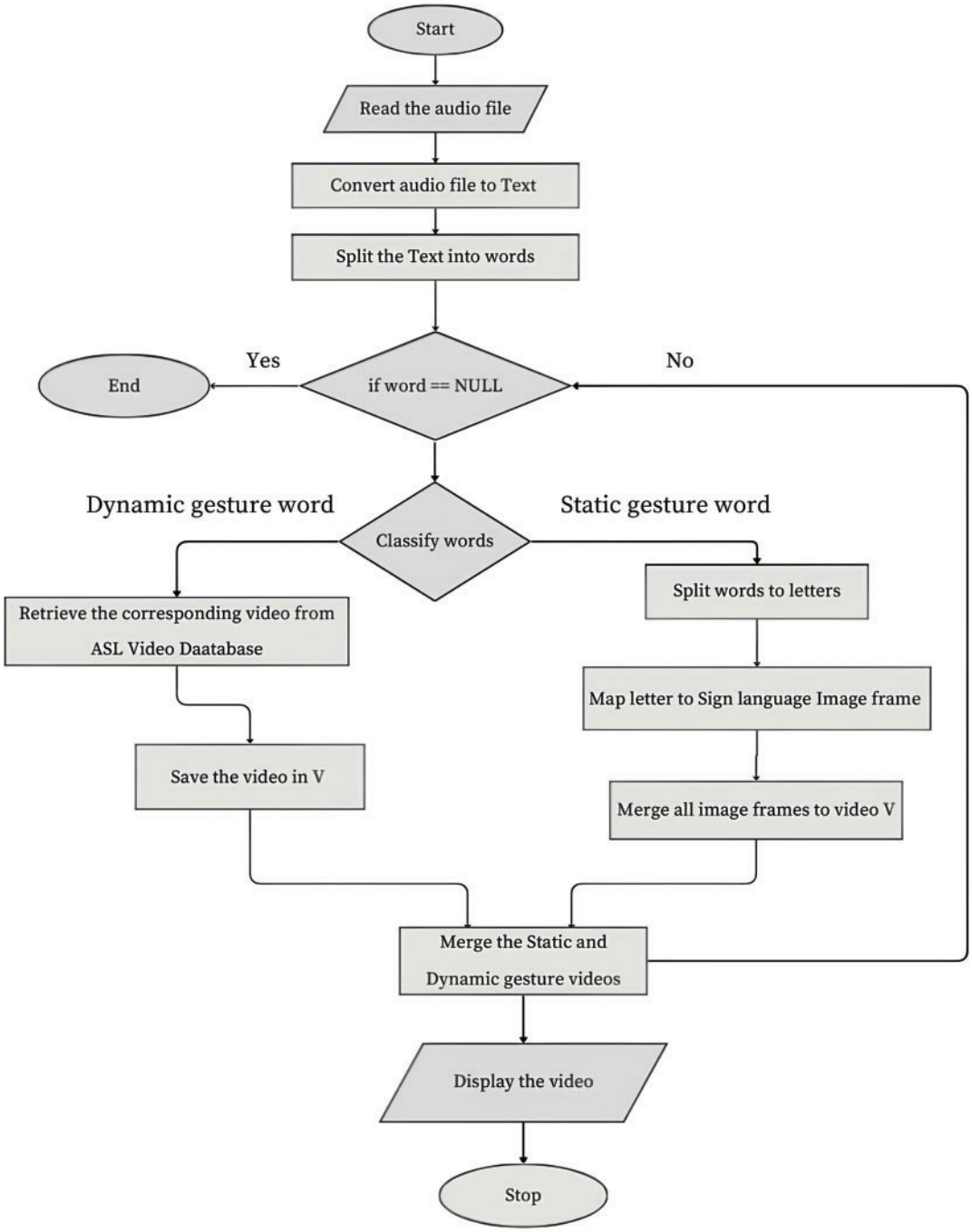

Figure 2: Flow chart for speech to ASL conversion model

4.2 Proposed ASL Recognition System for Communicating with Speech and Hearing Impaired

The proposed sign language recognition model converts the sign language shown by the hearing disabled to text data for the audience. This model uses a visual sensor to capture live videos of the sign language from the disabled. The video stream is converted into a series of image frames. The videos are split into image frames, and each frame is classified as a static or dynamic gesture. If the image frame is detected as a static gesture, then it is classified as one of the English alphabets. These alphabets are stored in the correct order to denote a word. The word is now stored in a queue data structure to maintain the word order in the sentence. When a gesture action is detected in the video clip, the proposed video classification algorithm classifies it as one of the gesture labels. The queue data structure is used to merge the static and gesture action words to maintain the semantic structure of the sentence. Finally, the detected words are clubbed to form a continuous text format to be displayed. Fig. 3 shows the proposed model for converting the live sign language videos from disabled to text data. The accuracy of the proposed ASL recognition system is tested with the LSTM and CNN deep learning models. The experimental study section records the results of the model.

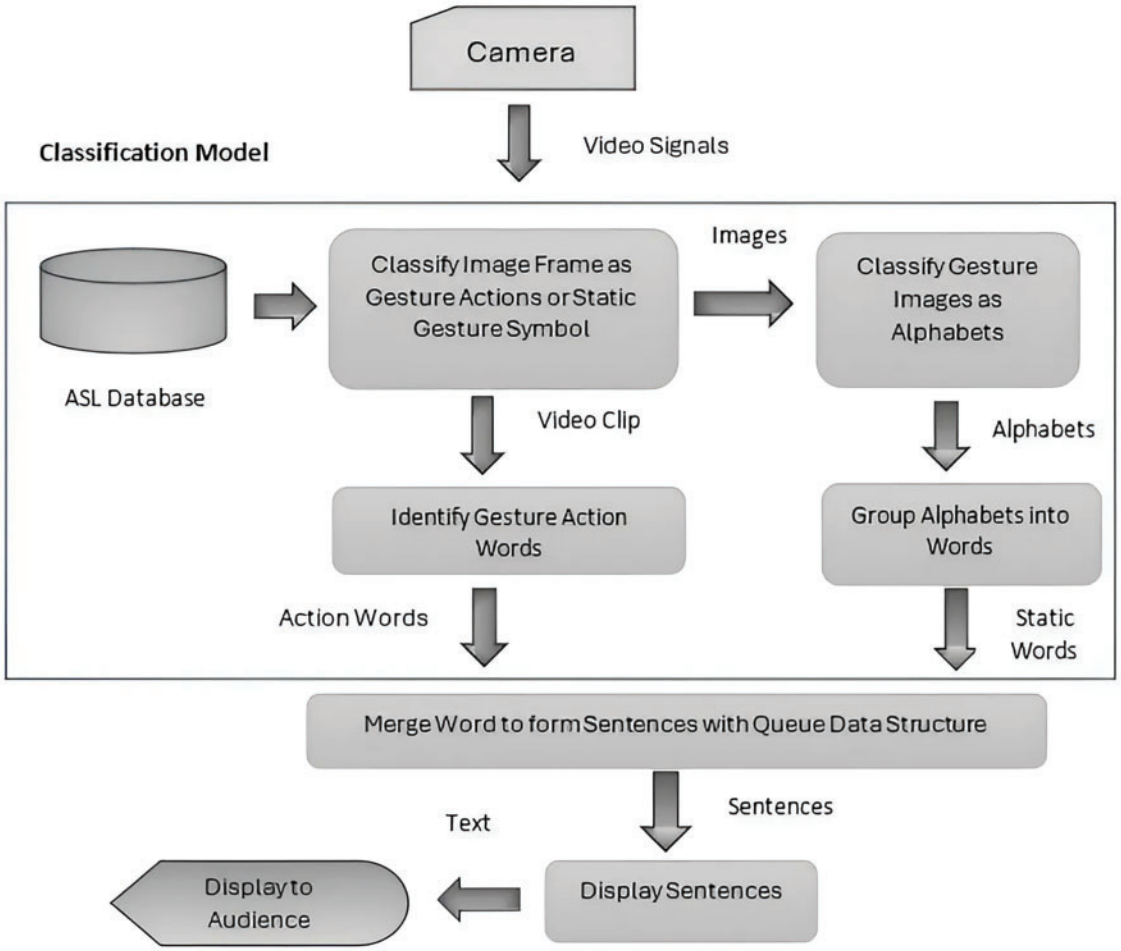

Figure 3: Proposed sign language recognition system for communicating with speech and hearing impaired

4.2.1 Algorithm for American Sign Language Recognition System

The video clipping V is segmented into set of images I where $I = \{I_1, I_2, I_3, \ldots, I_n\}$
for every image I_j in I
{
for every image frame I_j in the video
{
If I_j is an ASL alphabet, then
{Save the classification label for I_j in a string S_j.
S_w = StringConcat (S_w, S_j) // returns static word
else
{Enqueue S_w as element q_i in the queue Q.
Clear String S_w.
if I_{j+1} is frame of a gesture action video then
V_{dg} = Merge frame (V_{dg}, I_j, I_{j+1})
else
{ V_w =Gesture video classification (V_{dg}) . // returns dynamic word
Enqueue the V_w in the queue Q.}
}
Dequeue the elements in the queue Q and concatenate it to the text T.
Display the text data T.

4.2.2. Algorithm for Gesture Video Classification (V_{dg})

Step 1: The live stream video $\boldsymbol{V_{dg}}$ is split into images.

Step 2: The key points of the hand and face in the images are detected.

Step 3: Apply the LSTM model for gesture classification on the key point data.

Step 4: Return the dynamic gesture word.

In the above algorithm, the current and the next image frames in the video are checked in a sequence. If the current and the next frame are static gesture images, then the subsequent frames are grouped as a gesture video to represent a static gesture word. Once detected as a gesture action, the gesture video gets saved in a folder according to the arrival sequence. A queue data structure is used in the algorithm to ensure that the analyzed words are processed in the correct time sequence. Fig. 4 is the flowchart of the ASL sign language.

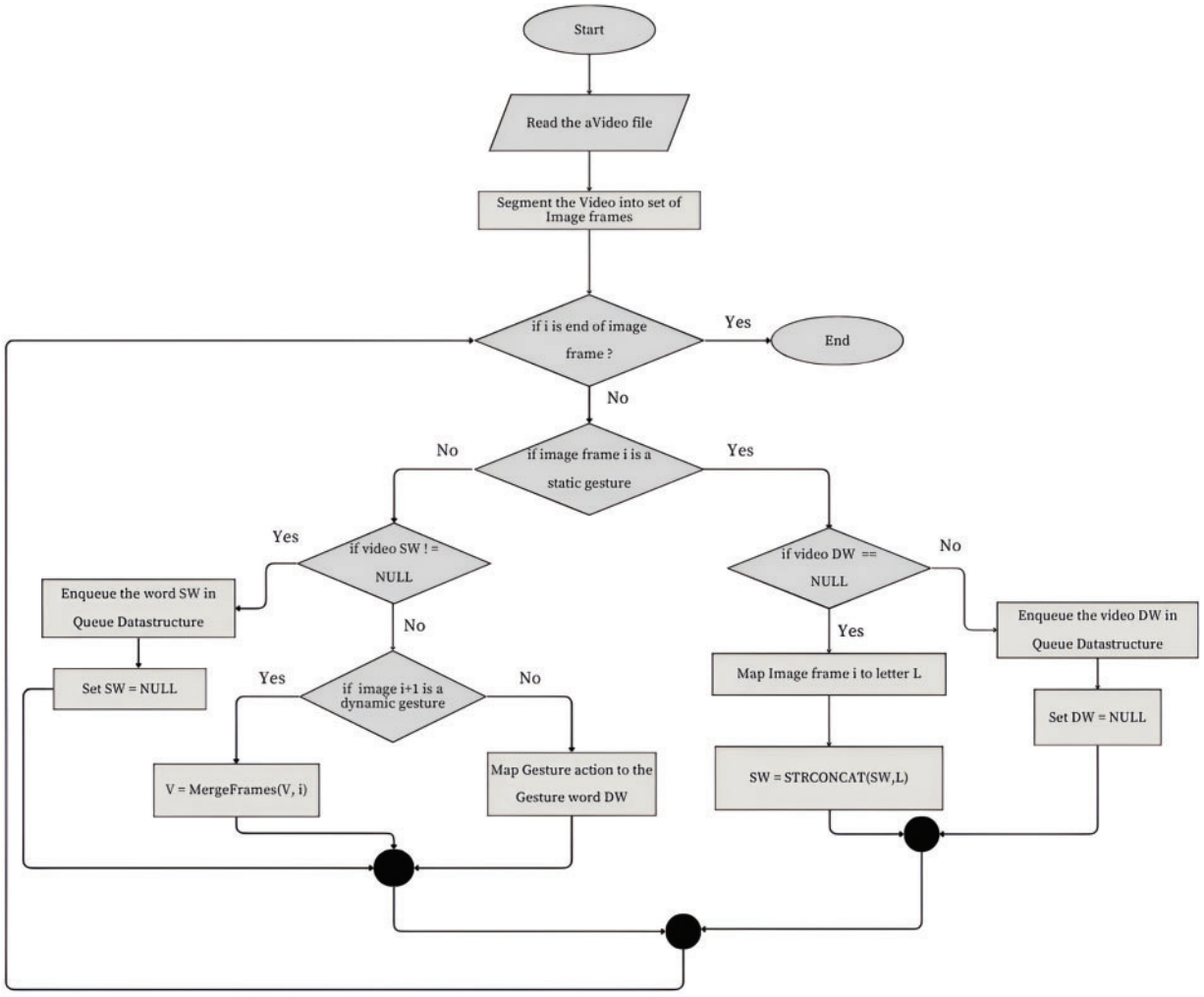

Figure 4: Flow chart for ASL recognition system for the speech and hearing impaired

4.3 CNN Model for Image Classification

The classification of the ASL sign language images is implemented by the CNN deep learning model. The conventional noise removal technique is replaced by the data augmentation in this framework. The noise removal process increases the computational time complexity of the proposed system. The delay in processing the images adversely affects the performance of the proposed sign language recognition system [24]. Therefore, replacing the noise removal step with the data augmentation step before image classification enhances the classification task. The augmentation technique includes geometric transformations such as flipping, rotation, cropping, and translation. Horizontal axis flipping of the images results in multiple versions of the same image. Random cropping resizes the small part of the original image to the image size. Translation preserves the image dimensions, retains the image size, and is preferred over random cropping [25]. The image augmentation technique applied before classification reduces the risk of overfitting through regularization. The convolution layer extracts the intricate features of the image by analyzing the pixel intensities and interesting patterns in an image. The 2D convolution layer in CNN performs the convolution of the input image and kernel to retrieve significant feature information from the image. The zero-padding technique ensures that the center of the kernel does not overlap with the edge of

the image. Strides are set as one to extract every key feature in the image and to avoid downsampling of the images [26]. The size of the kernel, stride value, padding parameter, and the number of kernels directly impact the CNN model. The pooling function retains the most deterministic features in the feature map, eliminating irrelevant information retains the most deterministic features in the feature map eliminating the irrelevant information. Pooling reduces the number of parameters to learn, which minimizes the risk of overfitting. The spatially transformed output from the pooling layer reduces the computation cost of the subsequent layers. Max pooling, which discovers the key feature in the space by downsampling, is apt for the CNN model in this study. MaxPooling identifies the distinct hand features by downscaling the image and removing invariances due to shifts, rotations, and scaling [27]. The varied distribution of inputs to the model reduces the training speed and, thereby, the stability of the training process. This issue is addressed by Batch normalization, which solves the internal covariate shift problem in deep learning models. Reducing gradient dependency creates a smoother gradient flow in the network, improving the learning rates and, subsequently, the risk of divergence. In this study, each training mini-batch unit has a convolution layer followed by pooling, batch normalization, and dropout layers [28]. While trying to learn intrinsic patterns from data, the network learns statistical noise, which leads to overfitting of the model. Overfitting is a condition where the model performs well for train data but suffers in accurate prediction or classification for new data points. In deep neural networks, specific neurons try to fix the misinterpretations of other neurons, which leads to co-adaptation and overfitting. Dropping neurons as individuals or groups allows the remaining neurons to process the inputs more broadly. Dropout of a neuron leads to the dropping of specific neurons along with their connections during the training updates. The selection of neurons to be dropped is either random or based on their energy. The neurons are dropped based on conditions such as spatial locations or functional roles. Specific neurons are more likely to be dropped out, and the probability parameter for the dropout function decides this fact. In this landscape, the dependence of the model performance on specific neurons is reduced, and the learning process is more generalized, building a robust model without overfitting issues. In the fully connected layer, each node is connected to every other node in the subsequent and previous layers to transform the feature space by altering the dimensions or keeping it constant. The proposed CNN model classifies images from different backgrounds, illuminations, and positions with high accuracy.

4.4 ResNet for Gesture Image Classification

ResNet is a deep neural network architecture built to address the problem of vanishing gradients in a deep neural network. The novelty of this research study is the design of the residual blocks that include shortcut connections or skip connections. These shortcut connections allow the network to learn the residual functions. Residual functions learn residual mapping rather than complete mapping. The vanishing gradient problem, a common issue in deep neural networks, is eliminated with ResNet. The structure of the ResNet includes input identity, main path, and skip connection. The input to the block is directly passed through the shortcut connections without any transformations. The main path of the block contains a series of convolutional layers, with batch normalization and activation functions like ReLU. The output of the main path is added elementwise to the input identity, creating the residual connection in the skip connection. The advantage of the ResNet model over other image processing models is that model performance does not degrade when the architecture gets deeper. The weights of ResNet are set with Stochastics Gradient Descent (SGD) with standard momentum parameters. The SGD method incorporates momentum to accelerate the convergence in the training process. The idea behind SGD with Momentum is to introduce a momentum term that accumulates the gradients over time. This helps to smooth out the oscillations in the updates, speeds up convergence, and bypasses the local minima.

4.5 LSTM for Video Gesture Recognition

With the advent of efficient deep learning methodologies for video action recognition, hand-crafted feature extraction with machine learning algorithms is now extinct in video recognition. Video recognition is a challenging task compared to image classification, as the model deals with multiple data types, including spatial and temporal information. The temporal and spatial data are processed separately to extract the significant features and later fused to classify the data. Video classification is broadly classified as uni-modal or multi-modal classification. Uni-modal is a technique where the audio, video, and text are classified separately. Multi-modal is a technique for fusing audio, text, and visual features to classify the data. LSTM classifies the temporal data to improve the model accuracy. LSTM is an enhanced version of RNN that addresses memory loss. LSTM has an architecture that considers both current and previous sequences while making predictions. LSTM includes a memory cell with an input gate, a neuron with a self-recurrent connection, a forget gate, and an output gate. The forget gate decides which information should be removed from the block. The input gate conditionally decides whether the cell should be updated or not. The output gate decides on the data output based on the information in the memory and input. The memory cell retains the input until the 'forget' gate is open and the input gate is closed. LSTM is a proven neural network for human action recognition.

4.6 Architecture of Static Gesture ASL Image Classification Model

CNN model is a sequential model with convolution, pooling, and batch normalization layers in blocks followed by the dense layer. The first convolution layer includes 128 filters, each with kernel size **3** × **3** and activation function ReLU. The pooling function is the MaxPooling2D pooling function of size **2** × **2**. Each convolution layer is followed by batch normalization and dropout layer. The second convolution layer has 64 convolution filters, each with kernel size **3** × **3** and activation function as ReLU. The Max pooling layer in this block is of size **2** × **2**. The third convolution layer has 32 filters, each with kernel size **3** × **3** and activation function ReLU. The dropout parameter is set as 0.25 between each set of CNN layers. The flattened layer is included after the hidden layers. The fully connected layer has two dense layers and a dropout layer in between them. The first dense layer of the model has 128 units and ReLU as an activation function. The second dense layer has 25 units with Softmax as an activation function. The optimizer used is Adam. The loss function is categorical entropy. The model is then compiled with a specific loss function, optimizer, and metrics, followed by training the model. The learning parameters include 10 epochs, with a batch size of 16 and a learning rate of 0.001. Fig. 5 shows the architecture for the classification of ASL images.

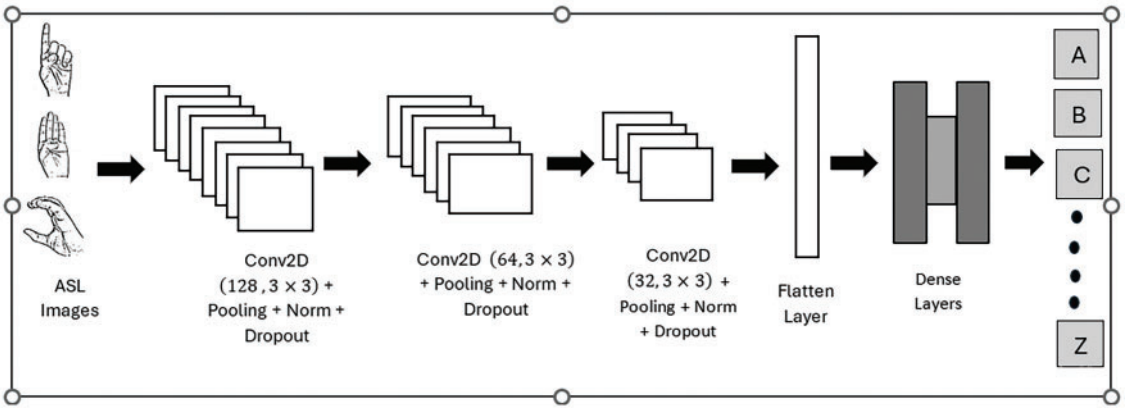

Figure 5: CNN Architecture for ASL alphabet classification

4.7 Architecture of Dynamic Gesture ASL Image Classification Model

The customized ANN model developed for key point detection includes an input layer, dropout layers, and fully connected dense layers. The input layer is of shape 21 × 2 followed by the dropout layer with 0.2 as a parameter. The model is a customized neural network model with two dense layers where the first dense layer is 20 units, with an activation function as ReLU. The second dense layer has 10 units with ReLU as an activation function. The last dense layer uses Softmax as an activation function. The intermediate dropout layer has a dropout parameter of 0.4. This customized model detects the key points in the hand given as input to the LSTM model. The face detection is implemented with a customized model with three stages of hidden layers. Zero padding preserves the fine details in the boundaries of the images. The architecture of the gesture recognition model is showcased in Fig. 5. The convolution layer with kernel size 7 × 7 and stride value as two extracts the significant face features. The activation function used is ReLU, and the max pooling layer is of size 2 × 2 with stride value 2. The second and third layers are Resnet blocks with filter sizes 64 × 64 × 256 and 128 × 128 × 512. The average pooling layer follows this ResNet block. The fully connected layer has a flattened layer followed by a block of three dense layers with ReLU as an activation function and two dropout layers. The loss function is a mean squared error, and the optimizer is Adam. The number of epochs used for training the model is set as 41 to improve the model accuracy. The model output is the summary of facial key points for the gestures. The ResNet model developed for face key point detection has achieved an accuracy of 89%. The face and hand key points detected from the image frames are collected as a dataset for the LSTM model. The stacked LSTM model is proposed and experimentally studied with a user-collected dataset to classify gestures. The stacked model is designed with three LSTM in a sequence followed by three dense layers. The first LSTM layer has an output dimension of 64, with activation function ReLU, return sequence parameter is set as true, and the input shape is 30 × 1662. The second LSTM layer with output dimension 128, with activation function ReLU, the return sequence parameter is true. The LSTM layers return a sequence of vectors of dimensions 64 and 128. ReLU is the activation function in the third layer with an output dimension of 64. In this stacked model of LSTM, the first two layers return the full output sequences, and the last one only returns the output sequence, eliminating the temporal dimension. The first dense layer has 64 units, and the second one has 32 units with activation functions such as ReLU. Gesture recognition includes the key point extraction followed by recognition. The key point extraction of the hands for gesture detection is carried out with a customized sequential model. Researchers have identified and used Twenty-one significant key points to detect hand movements. This study considers four prime key points per hand to detect hand movement. The four key points are 15, 17, 19 and 21 for the left hand and 16, 18, 20, and 22 for the right hand in Fig. 6a.

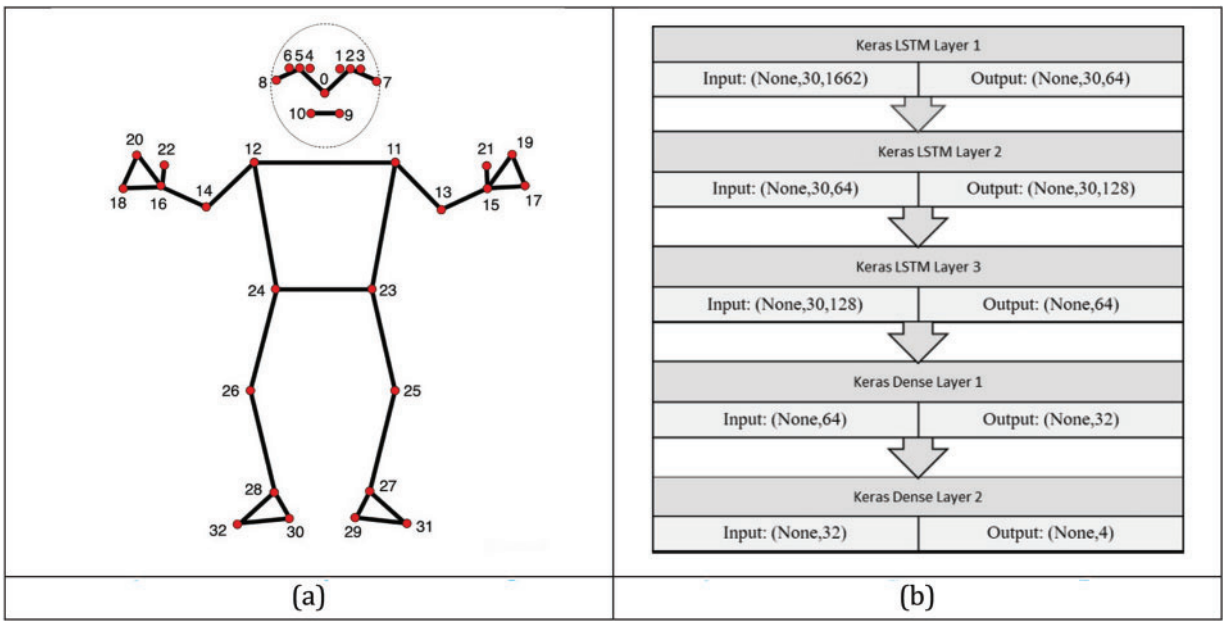

Figure 6: (**a**) Key landmark of the human body [29]; (**b**) Architecture of stacked LSTM model for gesture recognition

5 Results and Discussion

The results and discussion section includes experimental results for static gestures of ASL alphabet classification and experimental results for video gesture classification. The models are evaluated with the confusion matrix and graphs showing the accuracy over epochs during the training and testing phase.

5.1 Experimental Results for Static Gesture ASL Image Classification Model

The model for ASL image classification is trained with the images using the TensorFlow, Keras API python model. Keras is an open-source high-level neural network API written in Python that runs on top of other popular deep learning frameworks, such as TensorFlow. The proposed model shows the highest accuracy compared to state-of-the-art methodologies. The accuracy achieved is 99.7% for the ASL image classification from the data set [30]. Table 2 shows the loss and accuracy values for each epoch during the training phase to demonstrate the performance of the ASL alphabet classification model.

Table 2: Results for the gesture recognition for gesture action classes

Epoch	Loss	Accuracy
1	1.7583	0.4448
2	0.6441	0.7817
3	0.3776	0.8730
4	0.2917	0.9022
5	0.2324	0.9213
6	0.1943	0.9372
7	0.1866	0.9376
8	0.1670	0.9427
9	0.1444	0.9517

(Continued)

Table 2 (continued)

Epoch	Loss	Accuracy
10	0.1386	0.9532
Final test accuracy	0.0123	0.9975

Fig. 7a,b showcases the confusion matrix for the training and testing dataset. In Fig. 7b, diagonal nonzero elements indicate that all the predicted samples are true samples of the same class. In Fig. 7b, nonzero values in the non-diagonal element positions state that there are only negligible numbers of misclassified samples. Hence, the model has a very high accuracy of 99.7%. Figures show misclassification of samples for the letters 'C,' 'D,' and 'P.' Fig. 8a shows that the training accuracy escalates with the number of epochs. Fig. 8b demonstrates the decrease in loss value with each training phase epoch. The curve ends as a plateau, indicating no signs of overfitting. The number of epochs directly impacts the overfitting of the data. If the number of epochs is optimum, the model encounters an early stop condition where the overfitting condition is eliminated. The model has achieved such high accuracy because of the customized CNN model developed to detect the ASL alphabet. The numbers in the ASL language can also be detected with a similar customized CNN model to make the tool highly suitable for the learning environment.

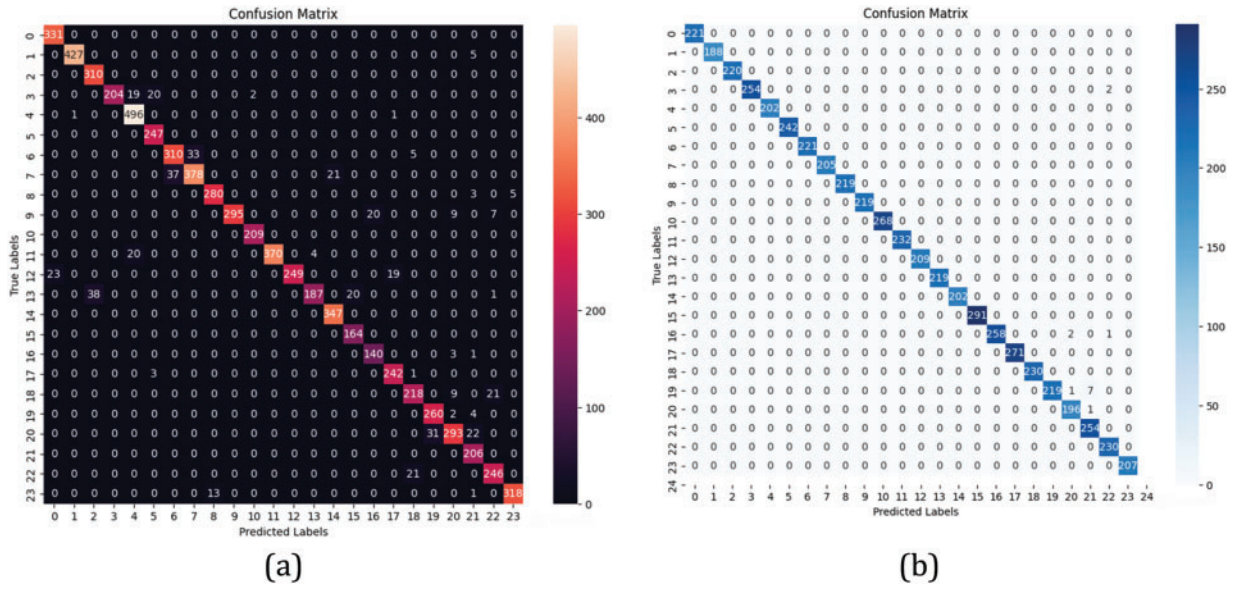

Figure 7: Performance evaluation of ASL alphabet classification model: (**a**) Training data confusion matrix; (**b**) Testing data confusion matrix

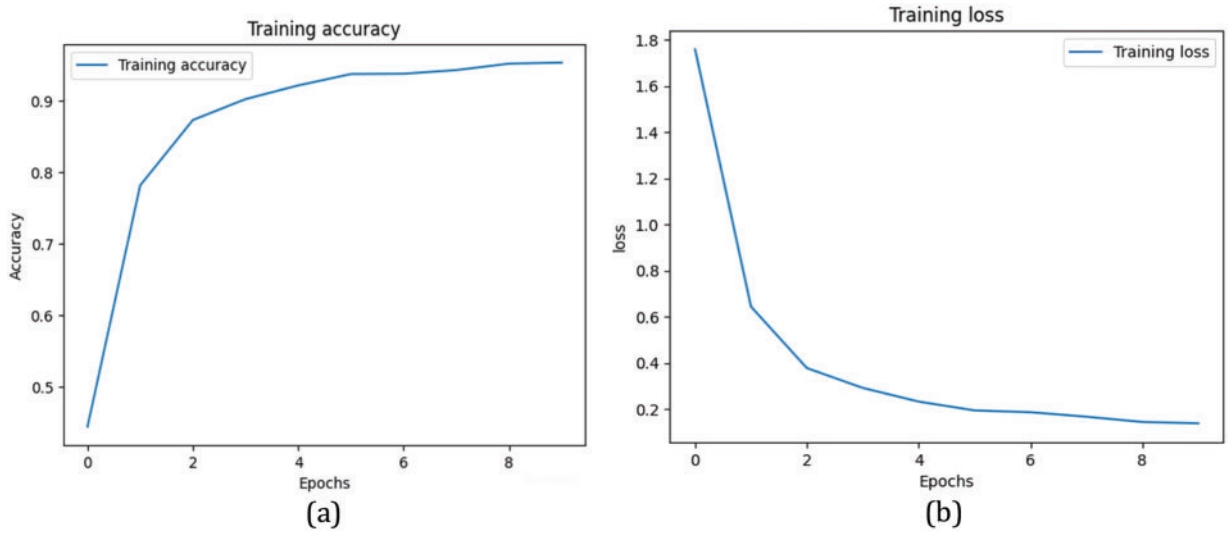

Figure 8: Performance evaluation of ASL alphabet classification model: (**a**) Accuracy vs. Epochs; (**b**) Loss vs. Epochs graph

5.2 Experimental Results for Video Gesture Recognition

The proposed model for gesture action recognition from live videos was experimentally tested with few gesture words. The key points from the hand and face are extracted from the multiple image frames for gesture action recognition. The classification results for the selected gestures in ASL are shown in Table 3. The heatmap for the model is demonstrated in Fig. 9.

Table 3: Results for the gesture recognition for gesture action classes

Classification label	Words	Precision	Recall rate	F1-score
0	Thankyou	0.99	0.99	0.99
1	Alldone	0.98	0.95	0.96
2	Help	0.95	0.98	0.97
3	Please	1.00	0.99	0.99

The proposed methodology shows improved results compared to the other existing research studies. The customized CNN model with data augmentation technique as pre-processing achieves a high accuracy of 99.7% for ASL alphabet classification. Dynamic gesture recognition is implemented by detecting the key points of hands and face with an accuracy of 99%. The model efficiency is improved due to the nature of the LSTM model to forget irrelevant information and retain the required information. LSTM efficiently handles the vanishing gradient problem, a major drawback found in other deep learning models. LSTM outperforms other deep learning models for variable input sequences by dynamically altering the internal state. Therefore, the stacked LSTM model demonstrates high accuracy for sign language recognition problems. In a nutshell, LSTM is apt for real-world applications in the natural language processing problem domain. The downside of deep learning techniques is the high computational cost. However, the feature engineering process has multiple layers of neural networks that encapsulate both the learning and classifying processes.

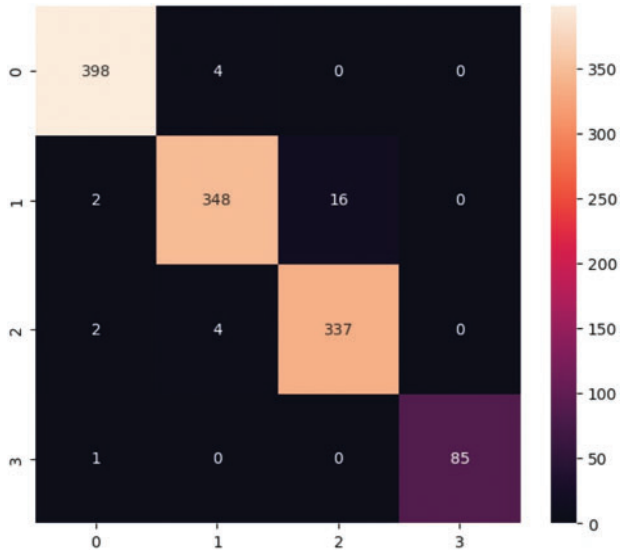

Figure 9: Heatmap for the LSTM gesture action detection model

Compared to other research studies, the proposed model shows high accuracy in detecting dynamic gestures in ASL. The dataset used for dynamic gestures is a user-built dataset for four words. Though the number of words analyzed is only a few, the data samples collected for this research study are significant for each word. The model's high performance is due to the large data sample set of gesture actions collected for each word. Creating such a large train and test data samples for each word in ASL sign language is an overwhelming task. Therefore, the researchers have only experimented with four words. Construction of a larger dataset and testing the model with the built dataset will be done as future work. Training the model on a customized dataset for a particular age group or ethnic group of students can improve the classification accuracy, reduce the model's computational complexity, and escalate the model's performance.

6 Conclusion

ALCTS is an asset for disabled high achievers enabling them to pursue a promising career. ALCTS empowers the speech and hearing impaired to acquire quality education and communicate with non-disabled peers. By integrating ALCTS in classrooms, institutions can provide the disabled with the opportunity to study in prestigious and highly ranked Universities and thereby develop technical skills at par with the non-disabled. ALCTS allows the disabled to communicate freely with teammates and to acquire social skills in any professional environment. Every child, physically or emotionally challenged, has the right to education. Hence every educational institution should make a concerted effort to facilitate students to pursue higher education without hesitations and barriers. The future direction of work will include developing a number system recognition model.

Acknowledgement: We sincerely thank Prince Sattam Bin Abdulaziz University (PSAU) for financial support provided for this research study.

Funding Statement: This project is sponsored by Prince Sattam Bin Abdulaziz University (PSAU) as part of funding for its SDG Roadmap Research Funding Programme project number PSAU-2023-SDG-2023/SDG/31.

Author Contributions: The authors confirm contribution to the paper as follows: conceptualization, Shabana Ziyad Puthu Vedu; methodology, Shabana Ziyad Puthu Vedu, Wafaa A. Ghonaim; software, Shabana Ziyad Puthu Vedu,

Pradeep Kumar Singh; validation, Shabana Ziyad Puthu Vedu, Naglaa M. Mostafa; formal analysis, Shabana Ziyad Puthu Vedu; investigation, Shabana Ziyad Puthu Vedu; data curation, Naglaa M. Mostafa; writing original draft preparation, Shabana Ziyad Puthu Vedu; review and editing, Wafaa A. Ghonaim, Naglaa M. Mostafa; project administration, Shabana Ziyad Puthu Vedu; funding acquisition, Shabana Ziyad Puthu Vedu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: ASL dataset available online for static ASL sign language recognition and user-collected dataset for gestures actions for specific words. https://www.kaggle.com/datasets/datamunge/sign-language-mnist (accessed on 20 February 2025).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Deafness and hearing loss. [cited 2023 Dec 01]. Available from: https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss.
2. What Is American Sign Language (ASL)? NIDCD. [cited 2023 Dec 01]. Available from: https://www.nidcd.nih.gov/health/american-sign-language.
3. Obi Y, Claudio K, Budiman V, Achmad S, Kurniawan A. Sign language recognition system for communicating to people with disabilities. Procedia Comput Sci. 2023;216(5):13–20. doi:10.1016/j.procs.2022.12.106.
4. Schick B, Williams K, Kupermintz H. Look who's being left behind: eeducational interpreters and access to education for deaf and hard-of-hearing students. J Deaf Stud Deaf Educ. 2006;11(1):3–20. doi:10.1093/deafed/enj007.
5. Subburaj S, Murugavalli S. Survey on sign language recognition in context of vision-based and deep learning. Meas: Sens. 2022;23(8):100385. doi:10.1016/j.measen.2022.100385.
6. Adeyanju IA, Bello OO, Adegboye MA. Machine learning methods for sign language recognition: a critical review and analysis. Intell Syst Appl. 2021;12(2):200056. doi:10.1016/j.iswa.2021.200056.
7. Umair MB, Basharat H, Usman A, Arslan T, Iqra T, Muhammad AB, et al. Feature based algorithmic analysis on american sign language dataset. IJACSA. 2019;10(5):100575. doi:10.14569/IJACSA.2019.0100575.
8. Sharma A, Mittal A, Singh S, Awatramani V. Hand gesture recognition using image processing and feature extraction techniques. Procedia Comput Sci. 2020;173(6):181–90. doi:10.1016/j.procs.2020.06.022.
9. Mummadi CK, Leo FPP, Verma KD, Kasireddy S, Scholl PM, Kempfle J, et al. Real-time and embedded detection of hand gestures with an IMU-based glove. Informatics. 2018;5(2):28. doi:10.3390/informatics5020028.
10. Kothadiya D, Bhatt C, Sapariya K, Patel K, Gil-González A-B, Corchado JM. Deepsign: sign language detection and recognition using deep learning. Electronics. 2022;11(11):1780. doi:10.3390/electronics11111780.
11. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell. 2016;39(6):1137–49. doi:10.1109/TPAMI.2016.2577031.
12. Sykora P, Kamencay P, Hudec R. Comparison of SIFT and SURF methods for use on hand gesture recognition based on depth map. AASRI Procedia. 2014;9:19–24. doi:10.1016/j.aasri.2014.09.005.
13. Pathan RK, Biswas M, Yasmin S, Khandaker MU, Salman M, Youssef AAF. Sign language recognition using the fusion of image and hand landmarks through multi-headed convolutional neural network. Sci Rep. 2023;13(1):441 Art. no. 1. doi:10.1038/s41598-023-43852-x.
14. Barbhuiya A, Ram K, Rahul J. CNN based feature extraction and classification for sign language. Multimed Tools Appl. 2021;80(2):3051–69. doi:10.1007/s11042-020-09829-y.
15. Meng L, Li R. An attention-enhanced multi-scale and dual sign language recognition network based on a graph convolution network. Sensors. 2021;21(4):1120. doi:10.3390/s21041120.
16. Xu X, Meng K, Chen C, Lu L. Isolated word sign language recognition based on improved SKResNet-TCN network. J Sens. 2023;2023:9503961. doi:10.1155/2023/9503961.

17. Sun SN, Han LS, Wei J, Hao HM, Huang JH, Xin WB, et al. ShuffleNetv2-YOLOv3: a real-time recognition method of static sign language based on a lightweight network. Signal Image Video Process. 2023;17(6):2721–9. doi:10.1007/s11760-023-02489-z.
18. Zhao R, Yan R, Chen Z, Mao K, Wang P, Gao RX. Deep learning and its applications to machine health monitoring. Mech Syst Signal Process. 2019;115(1):213–37. doi:10.1016/j.ymssp.2018.05.050.
19. Caliwag AC, Hwang HJ, Kim SH, Lim W. Movement-in-a-video detection scheme for sign language gesture recognition using neural network. Appl Sci. 2022;12(20):10542. doi:10.3390/app122010542.
20. Amangeldy N, Ukenova A, Bekmanova G, Razakhova B, Milosz M, Kudubayeva S. Continuous sign language recognition and its translation into intonation-colored speech. Sensors. 2023;23(14):6383. doi:10.3390/s23146383.
21. Papastratis I, Chatzikonstantinou C, Konstantinidis D, Dimitropoulos K, Daras P. Artificial intelligence technologies for sign language. Sensors. 2021;21(17):5843. doi:10.3390/s21175843.
22. Goldin-Meadow S, Brentari D. Gesture, sign, and language: the coming of age of sign language and gesture studies. Behav Brain Sci. 2017;40:e46. doi:10.1017/S0140525X15001247.
23. Introducing Whispe. [cited 2024 Mar 11]. Available from: https://openai.com/research/whisper.
24. Momeny M, Latif AM, Agha SM, Sheikhpour R, Zhang YD. A noise robust convolutional neural network for image classification. Results Eng. 2021;10:100225. doi:10.1016/j.rineng.2021.100225.
25. Shorten C, Khoshgoftaar TM. A survey on image data augmentation for deep learning. J Big Data. 2019;6(1):60. doi:10.1186/s40537-019-0197-0.
26. Yamashita R, Nishio M, Do RKG, Togashi K. Convolutional neural networks: an overview and application in radiology. Insights Imaging. 2018;9(4):4. doi:10.1007/s13244-018-0639-9.
27. Zafar A, Aamir M, Mohd Nawi N, Arshad A, Riaz S, Alruban A, et al. A comparison of pooling methods for convolutional neural networks. Appl Sci. 2022;12(17):8643. doi:10.3390/app12178643.
28. Ioffe S, Szegedy C. Accelerating deep network training by reducing internal covariate shift. 2015. doi:10.48550/arXiv.1502.03167.
29. Google for Developers. Pose landmark detection guide | MediaPipe. [cited 2024 Mar 14]. Available from: https://ai.google.dev/edge/mediapipe/solutions/vision/pose_landmarker.
30. Sign language MNIST; 2024 Mar 11. [cited 2024 Mar 14] Available from: https://www.kaggle.com/datasets/datamunge/sign-language-mnist.