International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 03 | March - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Smart Assistive Technology Using Machine Learning (ML) and Internet of Things (IOT) for Healthcare and Communication.

[1] M. Baritha Begum, [2]Subha Shree. S, [3]Shivani. M, [4] Yogeshwari. V, [5] Swetha. R

[1]B.E Associate Professor, Department of Electronics and Communication, Saranathan College of Engineering, Tiruchirapalli, Tamilnadu, India.

[2]B.E Student, Department of Electronics and Communication, Saranathan College of Engineering, Tiruchirapalli, Tamilnadu, India.

[3]B.E Student, Department of Electronics and Communication, Saranathan College of Engineering, Tiruchirapalli, Tamilnadu, India.

[4]B.E Student, Department of Electronics and Communication, Saranathan College of Engineering, Tiruchirapalli, Tamilnadu, India.

[5]B.E Student, Department of Electronics and Communication, Saranathan College of Engineering, Tiruchirapalli, Tamilnadu, India.

[1]barithaegum-ece@saranathan.ac.in,[2] subha1414shree@gmail.com, [3]Shivanimurugananantham@gmail.com, [4]yogavasu7373@gmail.com, [5]swetharajagopal04@gmail.com

---***---

Abstract- **Communication is a very difficult task for those who are not able to speak and hear properly. In many situations they are not able to express their requirements or request assistance during emergency situations. Therefore, to assist such people our work is proposing two simple assistive systems using IoT and Machine Learning techniques.**

The first proposed system is an IoT- based communication glove. In this glove a motion sensor is fixed on the glove or mount on the glove to detect hand movement. When a specific movement is made by the hand it recognizes the movement and displays a specific message on the screen. At the same time it sends the message to the caregivers through mobile application,here we use blynk. The proposed system also has a fall-detection feature. If a person falls down or makes an unusual movement while in unconscious situation it detects the movement and sends the message to the caregivers like the person is in emergency situation.

The second proposed system is based on machine learning techniques for hand gesture recognition using a webcam. The machine learning model recognizes the hand gesture by identifying the hand landmarks. The model predicts the gesture and displays the corresponding alphabet on the screen with a voice

Keywords:- **Assistive Communication System, Internet of Things (IoT), Machine Learning, Hand Gesture Recognition, Fall Detection, Caregiver Alert Notification.**

1. INTRODUCTION

Communication is an important part of human life because it helps people express their needs, ideas, and feelings to others. However, people who are unable to speak and hear often face many difficulties in communicating in their daily life. According to recent news of World Health Organization more than one billion people around the world experience some level of hearing loss and around four thirty million people suffer from disabling hearing loss that requires treatment.

It is also expected that this number may increase in the coming years due to various health and environmental factors.

While In India, a certain number of people are affected by hearing and speech disabilities. Many individuals

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 03 | March - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

depend on gestures or sign language to communicate with others. However, most people in society are not familiar with sign language, which creates a communication gap between disabled individuals and others. According to census reports, a large number of

people in the country live with hearing and speech impairments. In Tamil Nadu, there are more than two lakh people affected by hearing impairment and nearly eighty thousands people with speech disabilities. Because of these challenges, it becomes difficult for them to express their basic needs such as asking for help, requesting food, or informing others during emergency situations.

To reduce these difficulties, assistive technologies can be developed using modern technologies such as the Internet of Things (IoT) and Machine Learning. IoT technology allows devices and sensors to connect with each other through the internet and transmit information in real time. In the proposed work, an IoT-based communication glove is designed to help users communicate using simple hand movements. A motion sensor attached to the glove detects the hand movement and the corresponding message is displayed on an LCD screen. At the same time, the message is also sent to caregivers through a mobile application using the Blynk platform. The system also includes a fall detection feature that can detect unusual movements and send an emergency alert if the user falls down or becomes unconscious.

Along with the IoT system, a machine learning based gesture recognition system is also developed. In this system, a webcam is used to capture hand gestures and identify hand landmark point and these landmark features are used to train a machine learning model that can recognize different gestures. The trained model predicts the gesture and displays the corresponding alphabet on the screen along with voice output. This approach helps users communicate their message more clearly and effectively.

Thus, the proposed work focuses on developing simple and practical assistive systems using IoT and machine learning technologies to support people with hearing and speech disabilities and improve their communication ability and safety.

2. A REVIEW OF THE LITERATURE

Recent progress in assistive communication technologies has centered on the creation of intelligent systems designed to facilitate effective communication for individuals with speech and hearing impairments. To make it easier for people and machines to work together, people have suggested different ways to do it, like using wearable sensors, gesture recognition, machine learning algorithms, and IoT-based communication systems. Zhang and Chen put forward a system that uses voice and gesture-based human-machine interfaces for communication help. Their work talks about how to use AI to make it easier for people and machines to talk to each other. The study focuses on gesture-based communication techniques that enable individuals with disabilities to engage with digital systems more effectively.

Li, Wang, and Zhou made a system that uses wearable sensors to recognize sign language for people who can't speak. Their study shows how machine learning algorithms and wearable motion sensors can work together to recognize hand gestures and turn them into useful commands. The study shows that sensor-based gesture recognition works well in real-time communication systems.

Kumar and Verma looked into how to use MEMS accelerometer sensors and machine learning to recognize hand gestures. Their research demonstrated that accelerometer like the ADXL345 can precisely identify motion patterns and categorize gestures without the necessity for intricate flex sensor arrays. This method makes it easier to implement hardware while still ensuring reliable gesture recognition performance.

Patel and Shah suggested an IoT-based system for detecting falls and sending emergency alerts that uses wearable accelerometer sensors. The system uses motion data from wearable devices to find sudden falls and send emergency alerts over IoT communication networks. Their study shows how important it is to combine wearable sensors with IoT technologies for monitoring and safety in real time.

Singh and Kaur performed a survey on assistive technologies based on artificial intelligence for people who have trouble hearing or speaking. Their work shows how AI and the Internet of Things (IoT) are becoming more important in making wearable communication aids that are cheap. The research bolsters the impetus for systems such as SilentSpeak, which seek to enhance communication accessibility through intelligent embedded technologies.

Sarma et al. developed a real-time American Sign Language recognition system utilizing MediaPipe and machine learning methodologies. The system uses MediaPipe to find hand landmarks and then uses several machine learning classifiers, including Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Trees, and Random Forest, to accurately recognize gestures in real time.

Sharma and Gupta also looked into using a Random Forest classifier to recognize hand gestures. Their research demonstrates that ensemble learning methods enhance gesture classification precision and mitigate over fitting through the integration of multiple decision trees. The Random Forest method worked well for finding complicated gesture patterns in systems that use gestures to communicate.

The analysis of these studies indicates that wearable sensors, accelerometers, machine learning algorithms, and IoT communication technologies are integral to the advancement of gesture-based communication systems. But a lot of the systems that are already out there use either camera-based vision techniques or complicated sensor arrays. The proposed SilentSpeak system seeks to overcome these constraints by combining an accelerometer-based gesture recognition glove with an embedded IoT communication framework, facilitating real-time communication support for individuals with speech and hearing impairments.At the first occurrence of an acronym, spell it out followed by the acronym in parentheses, e.g., charge-coupled diode (CCD).

3. METHODOLOGY

A. Existing Systems (Existing Work)

Most sign language translation systems today use sensor-based smart gloves with built-in controllers to pick up hand gestures and turn them into text or speech. In a lot of cases, flex sensors are put on the fingers of a wearable glove to measure how much each finger bends. These sensors work on the piezoresistive principle, which means that the sensor's electrical resistance goes up as it bends. A voltage divider circuit turns the change in resistance into a voltage signal, and the microcontrollers Analog-to-Digital Converter (ADC) reads the resulting analog signal.

The built-in controller reads the values from the analog sensors all the time and compares them to pre-set threshold patterns to find specific sign language gestures. When the system recognizes a gesture, it turns it into text, speech, or a display message using output interfaces like LCD displays, mobile apps, or voice modules. These systems give people with speech and hearing problems a basic way to communicate by turning hand gestures into words that make sense.

Gesture recognition systems that use flex sensors are easy to use and cheap, but they have some technical problems. Flex sensors can only tell when fingers are bending, they can't tell when a hand is tilted, oriented, or moving. Because of this, these systems have trouble recognizing complicated gestures that people use when they talk to each other in sign language. Also, the resistance change in flex sensors isn't always directly related to the bending angle, which means that the response isn't always linear. This could make gesture recognition less accurate.

Another problem is sensor drift, which happens when sensors bend too much or when the weather changes. This drift causes sensor readings to change over time, so the system needs to be recalibrated often. If you keep bending the sensor strip, it can also damage the conductive material inside, which shortens the sensor's operational lifespan. Also, flex sensor analog signals are often affected by electrical noise which can cause ADC measurements to be wrong and make the whole system less reliable.

Once the extracted features are obtained, they can be used to train machine learning models for gesture classification. Gesture recognition is an area where many algorithms have been employed, including K-Nearest Neighbours (KNN), Support Vector Machines (SVM) and Random Forest classifiers. More sophisticated methods use hand landmark detection techniques to identify hand feature points from captured images, therefore providing key features for machine learning models and allowing for better accuracy of gesture recognition and fewer errors from background noise.

Although vision-based systems offer several advantages, they also have several drawbacks. Their accuracy may decline when the environment does not provide adequate lighting, has complex backgrounds, or when the camera view is obstructed. Additionally, some systems require a large amount of training data and a large amount of computational power to perform well, thus adding to the complexity of the entire system.

B. Proposed Work

The proposed system is an embedded system/IoT-based gesture recognition system. The system converts the pattern of hand motions into communication. The sensor component of the system is based on the ADXL345 Accelerometer, which is a three-dimensional MEMS accelerometer. The accelerometer is capable of measuring the acceleration of the object on the X, Y, and Z axes. The accelerometer is continuously sensing the orientation, tilt, and motion of the hand while the gesture is being performed. The accelerometer is connected to the processing component of the system using the I²C Serial Interface, which is used to transfer digital signals between the sensor and the processor with less noise.

The processing unit embedded within the system is ESP8266, which is a microcontroller unit integrated with a Wi-Fi transceiver and a TCP/IP protocol suite. The controller is used to acquire real-time acceleration data from the ADXL345 sensor and process the motion signals using firmware algorithms embedded within the controller unit. The sensor signals are sampled and analyzed to detect specific motion signals corresponding to a particular gesture pattern.

The outputs of the feedback messages generated by the system are based on the communication requirements. In the case of an emergency motion pattern detected by the system, the system will generate the message "Emergency! Help needed." In the case of an assistance request for basic needs, the system will generate the message "I need food or water." In the case of a medical assistance gesture detected by the system, the system will generate the message "I need medicine." In the case of the system failing to detect the motion pattern, the system will generate the message "Can't understand. Write it!" to indicate that the system is not able to recognize the gesture.

The system is able to provide real-time interpretation of gestures with the help of MEMS motion sensing, signal processing, and IoT technology-based wireless communication.

In this work, a machine learning based hand gesture recognition system is proposed to assist individuals with speech and hearing impairments in communicating effectively. The system is designed to recognize sign language gestures and convert them into meaningful text and voice output in real time. By integrating computer vision and machine learning techniques the proposed system aims to provide an efficient and user friendly communication tool.

The system captures hand gesture images using a webcam which serves as the primary input device. The captured frames are processed using image preprocessing techniques such as resizing and normalization to ensure consistent image quality and reduce variations caused by lighting conditions and background noise and image blurred. These preprocessing steps improve the reliability of the gesture recognition process.

To extract meaningful gesture information the system utilizes the MediaPipe framework for hand landmark detection. MediaPipe identifies 21 key landmark points on the hand, including the fingertips, finger joints, and wrist. These landmarks represent the geometric structure of the hand and help in accurately distinguishing different gestures. The spatial coordinates of these landmarks are used to generate numerical features that represent the gesture pattern.

The extracted landmark coordinates are converted into a structured feature vector and used as input for the machine learning model. In this work, the Random Forest classifier is used for gesture classification. Random Forest is an ensemble learning algorithm that constructs multiple decision trees and combines their predictions to improve classification accuracy and reduce overfitting. This algorithm is well suited for handling structured feature data and provides reliable performance for gesture recognition tasks.

During the training phase, the dataset is divided into training and testing subsets to evaluate the performance of the model. The trained model learns the patterns associated with different gesture classes and predicts the corresponding alphabet when a new gesture is provided as input.

In the real-time implementation, the webcam continuously captures the user's hand gestures. MediaPipe detects the hand landmarks from the captured frames, and the extracted features are passed to the trained Random Forest model for prediction. The predicted gesture is then displayed as text on the screen. In addition, the recognized text is converted into speech using a Text-to-Speech module, allowing the system to provide audio output as well.

The proposed system therefore enables real-time interpretation of sign language gestures and transforms them into both text and voice outputs. This approach helps bridge the communication gap for individuals who are unable to speak or hear, making interaction easier and more accessible.

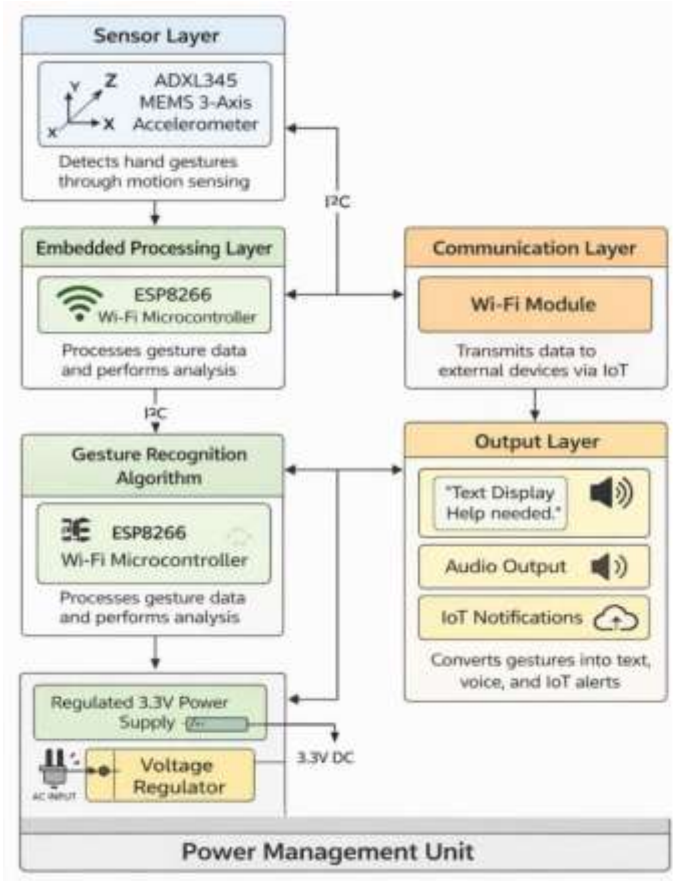
Once a gesture pattern is recognized by the system, the system generates multiple feedback signals through different interface modules. The visual feedback is generated by displaying the interpreted message as text on a display interface module. The voice feedback is generated using an audio output module that converts the recognized gesture into voice communication signals. Apart from the above local feedback signals, the system also generates IoT notification signals using the Wi-Fi transceiver integrated within the ESP8266 controller unit. Once a gesture is recognized by the system, the

corresponding notification signals are transmitted to other devices using internet communication protocols.

4. SYSTEM ARCHITECTURE

The system architecture is designed to detect hand gesture and transform it to a meaningful communication signal through the integration of various embedded technologies. The system architecture integrates motion sensors, embedded systems, IoT communication, output feedback, and power management to enable real-time gesture interpretation and system stability.

The system architecture is composed of five layers: the sensor layer, embedded processing layer, communication layer, output layer, and power management unit (PMU). The system architecture works sequentially to detect hand motion, interpret the gesture, transmit the message, and provide feedback and power stability to the system hardware.

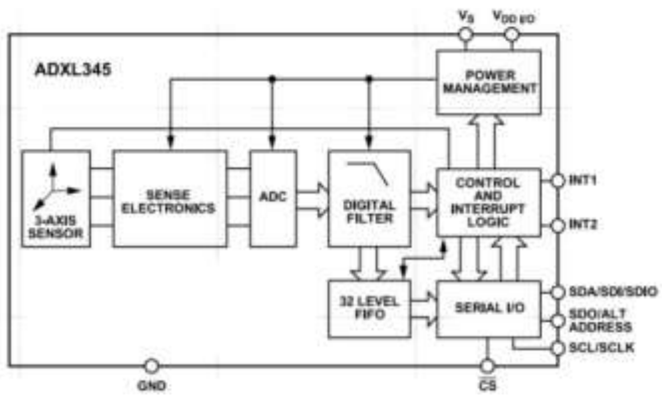


A. Sensor Layer

The role of the sensor layer is to detect the actual motion of the user's hand during the gesture's execution. The motion of the gesture is detected by the ADXL345.

The accelerometer detects the acceleration of the object over the X-axis, Y-axis, and Z-axis, allowing the detection of hand orientation, tilt, and motion. This motion signal corresponds to the actual movement of the hand during gesture execution.

The sensor produces a digital signal, which is sent to the embedded controller through the I2C protocol, allowing for the transfer of the signal with minimum electrical noise.



Functional block diagram

When the user moves their hand or fingers, acceleration is generated due to the motion of the hand. The ADXL345 accelerometer sensor detects this motion through its internal MEMS capacitive sensing structure. Inside the sensor, tiny capacitive plates are arranged in a micro-electromechanical structure. When the hand moves, the internal proof mass of the sensor shifts slightly, causing a change in the distance between these capacitive plates. This variation results in a change in capacitance. The change in capacitance is then converted into a small analog electrical signal that represents the acceleration of the motion along the X, Y, and Z axes. This signal is further processed by the embedded controller to interpret the gesture movement.

ADXL345 sensing flow

B. Embedded Processing Layer

The embedded processing layer is responsible for the real-time acquisition and analysis of motion data. The embedded processing unit used in the system is the ESP8266.

The microcontroller is used to continuously read the acceleration values from the sensor, and the motion signals are processed by the embedded firmware algorithms. The data obtained is analyzed to detect specific gesture patterns by observing the variations in the acceleration values for the three axes.

The identified gesture is then mapped to a specific message, which is stored in the firmware of the system.

1. Digital Filtering

The raw data obtained from the motion sensor in the form of acceleration values may not be reliable due to noise in the data caused by vibration in the hand and environmental factors. Therefore, to make the gesture detection process more reliable, a digital filter is used to filter the data obtained from the sensor in the ESP8266 controller's firmware. In this gesture recognition system, a low-pass filter is used to filter the acceleration data obtained from the sensor.

The low-pass filter works on the principle of taking a weighted average of the previous filtered value and the current value obtained from the sensor. This filtering makes the sensor's output stable and reduces sudden changes in the output, thus making the motion detection process stable even when the gesture is being performed. The formula for the low-pass filter is given by:

$$Xf = \alpha * Xprev + (1 - \alpha) * Xcurr$$

Xf is the filtered acceleration value, Xprev is the previous filtered value, Xcurr is the current sensor reading, and α is the smoothing factor. The smoothing factor is usually between 0.7 and 0.9. To get the best balance between noise suppression and signal responsiveness in this system, α is set to 0.8.

This filtering system gets rid of unwanted high-frequency changes and makes sure that motion detection stays stable while gestures are being made.

2. Gesture detection based on a threshold

After filtering the sensor signals, the processed acceleration values are compared to preset threshold limits to find gesture patterns. The threshold comparison method turns continuous motion signals into separate gesture commands.

The system sees the motion as a gesture in a certain direction when the filtered acceleration value goes above a certain positive threshold. The system also notices movement in the opposite direction if the filtered value goes below a negative threshold. We experimentally found these threshold values to be the best way to show different hand movements.

By comparing hand movements, the system puts them into different gesture groups that match certain communication messages.

C. Communication Layer

The communication layer sends the information about the recognized gesture to other devices using wireless communication. The ESP8266 microcontroller has a full TCP/IP network stack and an IEEE 802.11 b/g/n Wi-Fi transceiver built in. This lets the system talk to cloud platforms and other devices over the internet.

1. Starting Wi-Fi

When the system starts up, the firmware sets up the Wi-Fi module and connects it to a wireless access point. The initialization process turns on the radio frequency module, sets up the MAC layer, and starts the TCP/IP protocol stack. After it is set up, the device connects to the wireless network using information like the SSID, password, IP address, and gateway address.

2. Communication over the network

The communication process goes through the application layer, transport layer, network layer, data link layer, and physical layer of the layered network architecture. These layers use internet communication protocols to handle the sending of gesture data.

The system supports lightweight IoT communication protocols like MQTT, which lets the embedded device and remote monitoring systems send messages to each other in real time. HTTP POST requests can also be used to send gesture data to web servers over HTTP.

D. Output Layer

The output layer lets the user know what the system has done after it recognizes the gesture. A visual interface or an audio signal shows the interpreted message.

A buzzer connected to one of the ESP8266 controller's GPIO pins is used to create an audio feedback system. The buzzer makes a sound whenever it sees a gesture or an emergency message. The operation of the buzzer is achieved through the use of digital signals generated by the microcontroller. When the pin is set to a high level, the buzzer produces a sound, while a low level turns the buzzer off. In order to have a better sound generation, the buzzer can be connected to the microcontroller through a pulse width modulation to have a specific frequency sound.

5. PROPOSED METHODOLOGY

The proposed system performs hand gesture recognition using a webcam and machine learning techniques. The overall workflow begins with image acquisition and proceeds through dataset preparation, preprocessing, feature extraction, model training, and real-time prediction. The methodology pipeline is illustrated conceptually in the following flow fig.1

Fig.1 System Architecture

1. Webcam Input

The system starts with real-time image acquisition using a webcam. The webcam captures continuous frames of the user's hand gestures. These frames serve as the input data for the gesture recognition system.
Each frame is processed individually using computer vision techniques. The captured images contain the hand gesture along with background information such as lighting variations and surrounding objects. Therefore, further processing is required to isolate the relevant hand features.

Real-time input allows the system to recognize gestures dynamically and display the predicted alphabet on the screen.

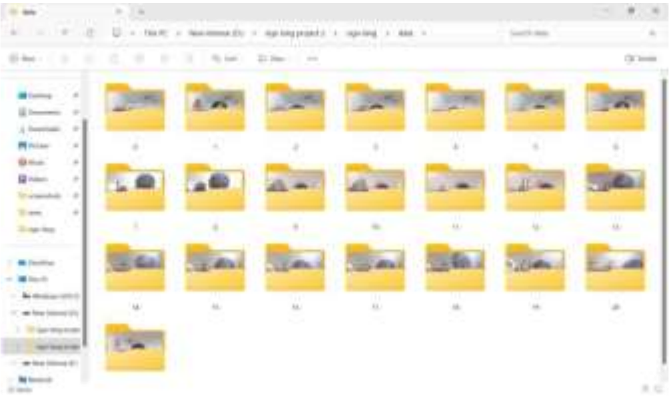


Fig.2

2. Dataset Collection

To train the gesture recognition model, a dataset of labeled hand gesture images was created. The dataset contains multiple gesture classes representing alphabet signs shown in fig.2 .

Dataset Details
* Number of gesture classes : 22
* Images collected per class : 500
* Total dataset size : 11,000 images

Each class corresponds to a specific hand gesture used for sign language recognition. The dataset includes multiple variations of each gesture in order to improve the learning capability of the machine learning model.

Variations include:
* Different hand orientations
* Slight finger movements
* Lighting condition changes
* Background variations

Although sign language contains 26 alphabet gestures, some gestures have very similar visual structures. These similarities can create confusions in static image recognition. Therefore, clearly distinguishable gestures were selected for training. If necessary, such gestures can be incorporated in the future by designing alternative gesture representations or collecting additional training samples.

The dataset is organized into labeled folders, where each folder represents a particular gesture class used for supervised learning

3. Data Preprocessing

The images collected from the webcam or dataset cannot be directly used for machine learning training. Raw images often contain noise, lighting inconsistencies, and unnecessary background information. Therefore, preprocessing is performed to enhance image quality and ensure consistent data representation.

The preprocessing stage includes image resizing, normalization, and noise reduction.

Frame Resizing
Captured images may have different resolutions depending on the camera device. To maintain consistency, all images are resized to a fixed dimension.
Example size used in the system:
224 × 224 pixels

Image Normalization
Image pixel values normally range from 0 to 255. Machine learning models perform better when input values are scaled to a smaller range. Therefore, normalization is applied to convert the pixel values into the range 0 to 1.

Noise reduction
Gaussian filtering is commonly used to smooth the image by computing a weighted average of surrounding pixels,

In this work the system primarily uses MediaPipe hand landmark detection, which internally focuses on the hand region and reduces the influence of background noise.

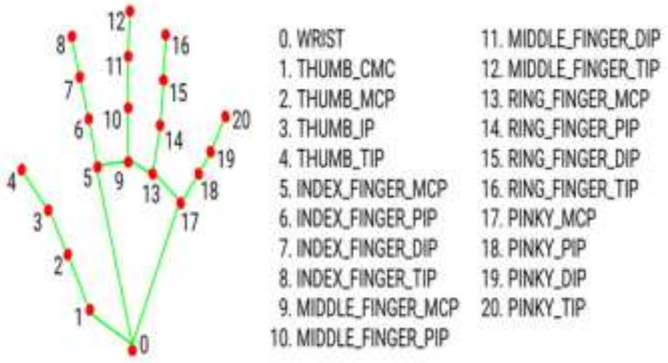


Fig.3(i)

4. Feature Extraction

After detecting the hand landmarks, the coordinate values of each landmark are extracted to form the input features for the machine learning model.Each landmark contains coordinate values:
* x coordinate
* y coordinate
As shown in fig.3

Total features = 21 x 2 = 42
Example representation of a feature vector

[x0, y0,x2, y2, x3, y3 ... x21, y21]

Fig.3(ii)

5. Data Cleaning and Validation

During dataset preparation, some samples may contain incomplete or incorrect landmark information. For example, the hand may not be properly detected or some landmarks may be missing. Such samples are removed during the data cleaning stage. Additional validation checks are performed to ensure consistent feature vector length, remove missing values, and verify correct class labels. This ensures that only valid gesture samples are used for model training.

6. Training Process

After preprocessing and feature extraction, the cleaned dataset is converted into structured numerical data using NumPy arrays. The dataset is then divided into training and testing sets using an 80:20 split. This means that 80% of the dataset is used to train the model, while the remaining 20% is used to evaluate the model performance. A supervised learning approach is used since each gesture sample is associated with a labeled class.

7. Random Forest Classifier

For gesture classification, the Random Forest algorithm is used. Random Forest is an ensemble learning method that combines multiple decision trees to improve prediction accuracy.

During training, multiple decision trees are constructed using different subsets of the training data. Each decision tree learns patterns from the gesture feature vectors.

When a new gesture is provided as input, the feature vector is passed to all decision trees. Each tree independently predicts a gesture class.

The final prediction is determined using majority voting, where the class predicted by most trees becomes the final output.

Example concept:
Tree 1 → Class A
Tree 2 → Class A
Tree 3 → Class B
Tree 4 → Class A
Tree 5 → Class A

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 03 | March - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

8. Model Evaluation

After training, the model performance is evaluated using the testing dataset. The predicted labels are compared with the actual labels to measure classification accuracy.

The accuracy metric is defined as:
A confusion matrix as shown in fig.4 is also used to analyze classification results across different gesture classes. The confusion matrix helps identify which gestures are correctly predicted and which gestures may occasionally be misclassified.

The trained model achieved high classification accuracy as shown in fig.4 on the testing dataset, demonstrating the effectiveness of landmark-based gesture representation.

6. SYSTEM IMPLEMENTATION

The proposed gesture recognition system was implemented in an embedded hardware system with motion sensors, a microcontroller unit, wireless communication modules, and output devices. The physical implementation of the system prototype is depicted in Fig.

In the hardware implementation of the system, the accelerometer sensors are mounted on the gloves to sense the motion and gesture of the hands. The sensors are able to detect the acceleration in the x, y, and z directions and send an electrical signal according to the orientation and motion of the hands. The signal is sent to the processing unit through the sensor interface connection.

The embedded processing unit has been implemented using ESP8266-based microcontroller modules, which have been mounted on the prototype boards. The microcontroller reads the real-time sensor data through the digital communication interface and performs signal processing activities such as filtering and gesture detection. The firmware continuously monitors the motion signals and detects the predefined gesture patterns.

A liquid crystal display (LCD) module has been integrated into the system for providing visual feedback in the form of the interpreted message based on the detected gesture. Additionally, the system has been provided with an audio output device that has been connected to the microcontroller for producing sound alerts for specific gestures. The buzzer has been connected through the GPIO pins of the microcontroller. The system is powered by a regulated power supply that provides the necessary voltage to power the microcontroller. The Wi-Fi module of the ESP8266 chip allows for wireless transmission of gesture data to external devices using internet communication protocols.

The hardware prototype is a proof of concept of the proposed system. The hardware prototype includes various hardware components such as sensing, processing, communication, and output.

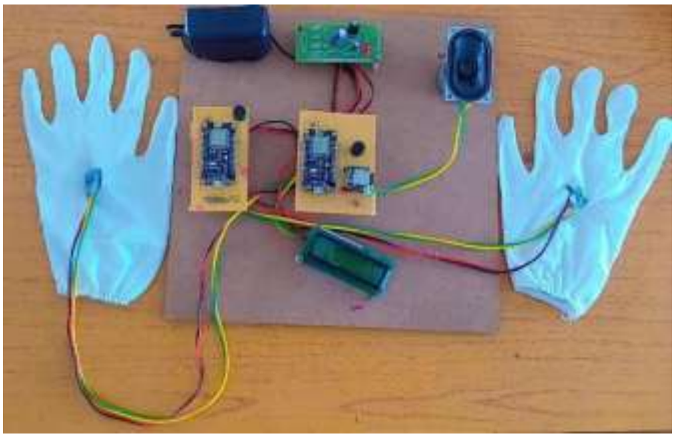


Fig.4- Hardware prototype of the implemented system

A. Real-Time Implementation

In the real-time system, the webcam captures hand gestures continuously.

In our work MediaPipe detects hand landmarks, the feature vector is generated, and the trained Random Forest model predicts the corresponding gesture class. The predicted alphabet is then displayed on the screen as shown in fig.5

During execution ,our system recognizes individual hand gesture alphabets and displays them on the screen. For example, we perform the gestures corresponding to the alphabets ‘A’ and ‘C’, the system forms the word ‘AC’. The recognized text is then converted into speech using a Text-to-Speech module and the voice output pronounces ‘AC’, allowing the gesture to be communicated audibly.

Word: AC

Sentence: AC

and the system speaks "AC" through the voice output.

Webcam Input
↓
Hand Detection (MediaPipe)
↓
Feature Extraction (x, y landmarks)
↓
Random Forest Model
↓
Gesture Prediction (Alphabet)
↓
Text Display
↓
Text-to-Speech Conversion
↓
Voice Output

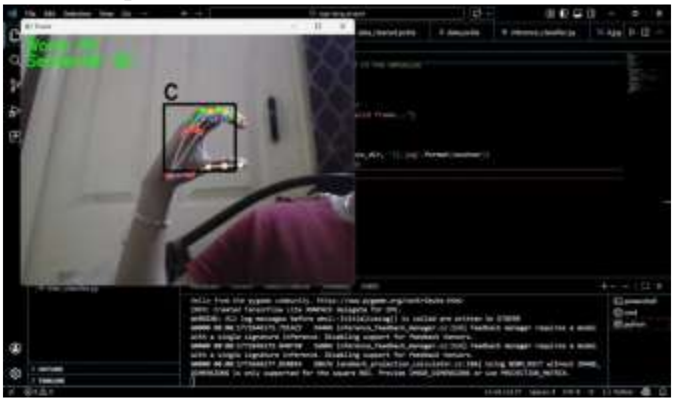


Fig.5

7. CONCLUSION

In this work, two assistive systems based on IoT and Machine Learning technologies are proposed to support individuals with speech and hearing impairments. Communication can be very challenging for such individuals, especially during daily interactions and emergency situations. Therefore, the proposed systems aim to provide a simple and effective solution to improve their communication ability.

The IoT-based system focuses on providing an easy way for users to convey basic needs and emergency messages. By using motion sensors integrated with a wearable glove, specific hand movements can trigger predefined messages that are displayed on an LCD screen and also sent to caregivers through a mobile application using the Blynk platform. In addition, the fall detection feature helps identify abnormal movements and alerts caregivers when the user may be in an emergency situation.

The machine learning–based system focuses on recognizing hand gestures representing sign language alphabets. Using a webcam and MediaPipe hand landmark detection, the system extracts important hand features and uses a Random Forest classifier to recognize the gesture. The predicted alphabet is displayed as text and converted into voice output using a Text-to-Speech module, enabling both visual and audio communication.

Overall, the proposed systems demonstrate that the integration of IoT and machine learning can provide an effective assistive technology for people with communication difficulties. These systems help bridge the communication gap and make interaction easier in real-time situations.

REFERENCES

1]. Suharjito, Anderson R., Wiryana F., Ariesta M. C., and Kusuma G. P., "Sign Language Recognition Application Systems for Deaf-Mute People: A Review Based on Input–Process–Output," Procedia Computer Science, vol. 116, pp. 441–448, 2017.

[2]. Ahmed S., Islam R., Zishan M. S. R., Hasan M. R., and Islam M. N., "Electronic speaking system for speech impaired people: Speak up," International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), pp. 1–4, 2015.

[3]. Harish N. and Poonguzhali S., "Design and development of hand gesture recognition system for speech impaired people," International Conference on Industrial Instrumentation and Control (ICIC), pp. 1129–1133, 2015.

[4]. Kuroki K., Zhou Y., Cheng Z., Lu Z., Zhou Y., and Jing L., "A remote conversation support system for deaf-mute persons based on bimanual gestures recognition using finger-worn devices," IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), pp. 574–578, 2015.

[5]. Mohandes M., Aliyu S., and Deriche M., "Prototype Arabic sign language recognition using

multi-sensor data fusion of two Leap Motion controllers," IEEE 12th International Multi-Conference on Systems, Signals & Devices (SSD), pp. 1–6, 2015.

[6]. Ramakant, Shaik N. E. K., and Veerapalli L., "Sign language recognition through fusion of 5DT data glove and camera-based information," IEEE International Advance Computing Conference (IACC), pp. 639–643, 2015.

[7]. Shukor A. Z., Miskon M. F., Jamaluddin M. H., Ibrahim F. B. A., Asyraf M. F., and Bahar M. B. B., "A new data glove approach for Malaysian sign language detection," Procedia Computer Science, vol. 76, pp. 60–67, 2015.

[8]. Abdulla D., Abdulla S., Manaf R., and Jarndal A. H. J., "Design and implementation of a sign-to-speech/text system for deaf and dumb people," 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA), pp. 1840–1844, 2016.

[9]. O. Koller, H. Ney, and R. Bowden, "Deep Hand: How to Train a CNN on 1 Million Hand Images When Your Data Is Continuous and Weakly Labeled," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3793–3802, 2016.

[10]. S. Mitra and T. Acharya, "Gesture Recognition: A Survey,"IEEE Transactions on Systems, Man, and Cybernetics, vol. 37, no. 3, pp. 311–324, 2007.

[11]. V. Pisharady and M. Saerbeck, "Recent Methods and Databases in Vision-Based Hand Gesture Recognition: A Review," Computer Vision and Image Understanding, vol. 141, pp. 152–165, 2015.

[12]. Z. Cao, G. Hidalgo, T. Simon, S. Wei, and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 1, pp. 172–186, 2021.

[13]. F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1251–1258, 2017.

[14]. C. N. Reddy, A. Razzaque, Harish, K. Arulini, B. Gayathri and M. B. Begum, "AI-Driven Analysis of Consumer Behavior using IoT Systems," 2025 6th International Conference on Data Intelligence and Cognitive Informatics (ICDICI), Tirunelveli, India, 2025, pp. 1463-1469, doi: 10.1109/ICDICI66477.2025.11135197.

[15]. M. Baritha Begum, "Real-time Security in Sensor Networks in Sequential Approach with BWT Compression, Huffman Coding and Reduced Array Encryption," J. Syst. Sci. Syst. Eng., pp. 1–45, May 2025, doi: 10.1007/s11518-025-5661-0.

[16]. A.Yogeshwaran, K. Umadevi, "Optimized neural network-based micro strip patch antenna design for radar application," Intelligent Automation & Soft Computing, vol. 35, no.2, pp. 1491–1503, 2023.

[17]. B Indira Priyadarshini, K Mary Keerthi, V Manimegalai, K Ravikumar, Bhavyarajsinh D Jhala, M Baritha Begum, Leveraging Artificial Intelligence For Strategic Decision-Making In Business Management, Proceedings of International Conference on Computer Science and Communication Engineering (ICCSCE 2025), Atlantis Press, 2025 PP: 132-145, https://doi.org/10.2991/978-94-6463-858-5_13.

[18]. Anushree A Aserkar, Navaneetha Krishnan Rajagopal, P Jayatharani, K Arulini4, Vinod Waikars, M Baritha Begum, Data Driven Predictive Analytics for Strategic Decision-Making and Innovation, Proceedings of International Conference on Computer Science and Communication Engineering (ICCSCE 2025), Atlantis Press, 2025 PP: 326-339, https://doi.org/10.2991/978-94-6463-858-5-29.

[19]. S. Santhosh, R. Srivastava, M. Subbiah, L. C. Maddila, C. Malla and M. B. Begum, "Nanoelectronic Sensors for Real-Time Health and Environmental Monitoring," 2025 5th International Conference on Intelligent Technologies (CONIT), HUBBALI, India, 2025, pp. 1-6, doi: 10.1109/CONIT65521.2025.11167273.

[20]. D. Sivakumar and B. Suganthi, "Performance Comparison and Evaluation of Ad Hoc Routing Protocols," Int. J. Sci. Eng. Res., vol. 5, no. 4, pp. 218–221, Apr. 2014.

****