Urav Dalal[1*], Aasmi Thadhani[2], Mahek Upadhye[3], Shreya Shah[4], Dr. Meera Narvekar[5], Dr. Nilesh Patil[6]

Deep Learning-Enabled Smart Glove for Real-Time Sign Language Translation

Journal of Electrical Systems

***Abstract*—**
Individuals with speech impairments face unique challenges in communication. They find it difficult to communicate with people conveniently. American Sign Language is a widely recognized sign language that employs hand gestures. It is important for society to promote inclusivity and better understand the different communication needs of these individuals. This research aims at developing an improvised sign language to text translator using deep learning and sensor fusion. The glove consists of five flex sensors and an MPU-6050 sensor to capture hand gestures and movements. Data collected from the sensors is processed and transmitted to a deep learning model trained on a specially created set of various sign language actions. The integration of flex sensors enables the detection of finger movements, while the MPU-6050 sensor provides information about hand orientation and motion. By combining data from these sensors, the glove is effective at accurately recognising gestures.

Index Terms—Sign language, gestures, flex sensor, MPU-6050 sensor, deep learning, Bi-directional LSTM.

I. INTRODUCTION

In today's interconnected world, communication has become more efficient, yet there remains a significant segment of the population that encounters barriers in expressing themselves. Among these individuals are those who have severe speech impairments. Despite advancements in technology, there is a notable absence of widely-used applications designed to cater to their unique communication needs. Addressing this gap has become a focal point for technologists worldwide, who are exploring innovative solutions to facilitate better communication for these individuals. There are two primary approaches that are currently being developed to address this communication gap:

sign language detection using computer vision and gesture vocalizers.

[1] This method uses computer vision to detect gestures, converting them into sign language. It employs a camera to continuously capture gestures, which are predicted using a mix of recurrent and convolutional neural networks. While promising, this approach has drawbacks, primarily related to its lack of portability and convenience. The necessity of a camera to constantly monitor gestures makes the system less practical for everyday use, particularly in dynamic and mobile environments.

Another approach involves the use of gesture vocalizers [2], that makes use of a glove with 4 flex sensors for 4 fingers except the thumb. The finger bend is identified by sensors and mapped to gestures using a database of predefined gestures. Such devices are typically more compact and portable, making them more convenient for users. By incorporating sensors that detect the movement and positioning of the hand, gesture vocalizers can interpret a wide range of gestures with high accuracy.

The proposed solution is a wearable device that leverages gesture recognition technology allowing people with speech impairments to interact with greater ease. The device consists of a glove embedded with five flex sensors, each capable of accurately detecting the bending of individual fingers through changes in electrical resistance. Additionally, the glove incorporates an MPU-6050 module, which includes an accelerometer and a gyroscope. The accelerometer measures the hand's movement, while the gyroscope detects its orientation and position.

[1*]*Department of Computer Engineering, Dwarkadas J. Sanghvi College of Engineering,* Mumbai, India. uravdalal01@gmail.com
[2]*Department of Computer Engineering, Dwarkadas J. Sanghvi College of Engineering,* Mumbai, India. aasmithadhani@gmail.com
[3]*Department of Computer Engineering, Dwarkadas J. Sanghvi College of Engineering,* Mumbai, India.mahekupadhye123@gmail.com
[4]*Department of Computer Engineering, Dwarkadas J. Sanghvi College of Engineering,* Mumbai, India. shreyashah100803@gmail.com
[5]*Department of Computer Engineering, Dwarkadas J. Sanghvi College of Engineering,* Mumbai, India. meera.narvekar@djsce.ac.in
[6]*Department of Computer Engineering, Dwarkadas J. Sanghvi College of Engineering,* Mumbai, India. nilesh.p@djsce.ac.in

This setup allows the device to differentiate between two categories of gestures: dynamic as well as static. Static gestures do not involve movement and can be captured by the flex sensors and accelerometer, while dynamic gestures involve movement that can be accurately captured by the gyroscope in the MPU-6050.The integration of flex sensors and the MPU-6050 module allows the device to gather comprehensive data related to hand movements and gestures. Flex sensors offer insights into the bending angles of the fingers, while the accelerometer and gyroscope offer insights into the hand's overall motion and orientation. This combination of sensors ensures that the device can accurately interpret a wide variety of gestures. The sensor data from the glove is used to generate a comprehensive American Sign Language (ASL) dataset consisting of 34 classes which includes the 26 letters and common words used while conversing. A Python script is developed to collect gesture data, recording both static and dynamic gestures for exactly three seconds, ensuring that all data collected during this time period is labelled as the same gesture. The data is subsequently employed to train the model.

The system utilizes a Bi-directional Long Short-Term Memory (LSTM) model, an instance of Recurrent Neural Network (RNN) that is capable of handling data which is sequential in both forward as well as backward directions. This model is a suitable choice for processing sequential data. In sign language, the context of a gesture is essential for accurate interpretation. For example, the meaning of a single gesture can vary based on what occurs before or after it. Bi-directional LSTMs perform well in capturing this context by examining the complete series of movements. The interpreted gestures are then sent to a user-friendly client interface. This interface displays the translated gestures as text, enabling seamless communication. The design prioritizes accessibility, allowing individuals with speech impairments to use the device in various social and professional settings.

II. LITERATURE REVIEW

As observed in [3], a knitted glove is utilized due to its high sensitivity to strain along the knit, enabled by embedded silver threads. Stripped wire woven over the knit forms a strain sensor loop. The STM32H7 microcontroller processes data from the glove and a 3-axis accelerometer using an LSTM recurrent neural network is trained on a custom dataset of 24 ASL letters and words.

The study [4] maps American Sign Language gestures using a set of preset gestures by using MEMS accelerometers in place of flex sensors on a glove. MEMS accelerometers are inexpensive and provide accurate movement tracking, however they are not capable of measuring finger bending directly, unlike flex sensors. Because of this constraint, additional computational methods could be needed to deduce the bending angles, as accelerometers by themselves are insufficient to detect hand movements.

This paper [5] suggests a radar sensor-based approach for hand gesture recognition. This method uses sensors like radar and depth cameras to detect hand motions, as opposed to glove-based systems. Although this approach was not created with sign language translation in mind, it finds utility in fields where gesture detection might improve user involvement, such as video games.

The aim of this research [6] is to predict Indian Sign Language gestures using a proprietary dataset of over 1100 video samples and 11 sentences. It evaluates six combinations of GRU and LSTM models. The gating mechanisms of these two allow them to recall past inputs and select appropriate words based on past activations and current input properties. However, a limitation is the compulsory need for a camera to detect the gestures which does not make it portable.

The aim of this research [7] is to develop an interpretive system for American Sign Language (ASL) using sensor fusion technology. Self-developed IMU sensors, positioned on five fingers and one on the hand, record the complex movements involved in ASL. These lightweight sensors interface with a Teensy microcontroller known for its small size and effective

processing capabilities. To create a diverse set of ASL gestures, samples from individuals with disabilities were collected, resulting in a dataset comprising 27 different classes which is used to train a RNN model.

This study [8] focuses on developing a smart glove which is able to detect ASL gestures. Five flex sensors along with a MEMS accelerometer is used to measure finger bends to capture hand orientation. These sensors are interfaced with an Arduino Nano board, and data is transmitted wirelessly using a Bluetooth HC-05 module. Flex sensors give the finger positions, while the accelerometer tracks hand orientation. Processing occurs on the Arduino Nano followed by mapping sensor inputs to ASL gestures based on predefined conditions. However, the system sometimes misclassifies gestures that look similar.

Research [9] that explores the use of machine learning techniques to map sign language gestures. Their system involves a glove which uses 5 flex sensors for the fingers and a 3-axis accelerometer, interfaced with the Arduino mega 2560. Using a serial communication, the sensor data is sent to a laptop. On processing the sensor data, Random Forest Classifier is used to predict the required output. Lack of a gyroscope does not capture movement of the hand, which makes predicting dynamic gestures difficult.

From the observations made above, it became clear that a method which utilises both, a smart glove and deep learning techniques to map the gestures needs to be explored. This research thus develops a wearable device with five flex sensors, one for each finger to capture finger movements, along with an accelerometer to track movement and a gyroscope to monitor orientation of the hand. This helps to train the neural network on a variety of data and include dynamic gestures as well. Due to sensor readings not being extremely consistent, the use of deep learning can help map the gestures better rather than simply mapping them to predefined conditions, as the model can handle certain level of inconsistency in the data.

III. METHODOLOGY

A. Hardware Components

1. Arduino Mega Microcontroller

The Arduino mega, based on the AT mega 2560 8-bit microcontroller, is chosen as the computation powerhouse of the glove. It provides more memory space and has higher processing power which makes working with multiple sensors together smoother. It has 16 analog pins for input and output along with 54 digital pins, in contrast to the Arduino Uno's 6 analog pins and 16 digital pins. To utilize I2C modules like the MPU-6050, the Mega has dedicated SDA (serial data) as well as SCL (serial clock) pins that makes it possible to use flex and MPU-6050 sensors simultaneously, unlike the Arduino Uno where the analog and I2C pins overlap. With the extra pins, it becomes easier to interface more modules in the future such as Bluetooth and Wi-Fi.

2. Flex Sensors

Flex sensors operate by detecting changes in resistance when the strip is bent. It is a variable resistor which measures the amount of deflection it experiences when bent. Thus, flex sensors are a suitable choice for measuring finger bend angles, making it possible to determine sign language actions with accuracy. These sensors are interfaced using analog pins of the Arduino board.

3. MPU-6050

The MPU-6050 incorporates both a gyroscope and an accelerometer, enabling the analysis of complex movements across all six axes. Placed on the wrist, it helps to know the orientation of the hand. This information is critical for mapping gestures which have similar flex bend angles with difference in orientation of the hand. Even dynamic gestures can be

mapped due to the 3-axis accelerometer and 3-axis gyroscope which measure linear and angular movements of the hand respectively. The sensor uses the I2C communication protocol. I2C, short for Inter-Integrated Circuit, is a protocol that serves as bus interface connection that facilitates serial communication among devices. I2C utilizes SDA (Serial Data) for data transfer and SCL (Serial Clock) for signal timing.

The glove and its components are designed to be user-friendly and durable as seen in Fig 1. An adjustable Velcro patch on the arm allows the user to easily slide their hand into the glove without disrupting the connection. To avoid a lot of components to be placed around the wrist, the Arduino Mega is positioned along the arm. This ensures that all gestures can be made with ease without any component or wire interrupting the gesture. All of the flex sensors and resistors are soldered on a Zero PCB Board. The MPU-6050 is also mounted on the Zero PCB located at the wrist. The connections on the glove are as follows: the flex sensors have two pins, one of which is shorted with the VCC of all other flex sensors, and the other is used for both grounding and connecting to the analog pin. This pin is grounded once it is connected to a resistor. Flex sensors are connected to the board's analog pins. The Arduino Mega interfaces with the MPU-6050 via the SDA and SCL pins.

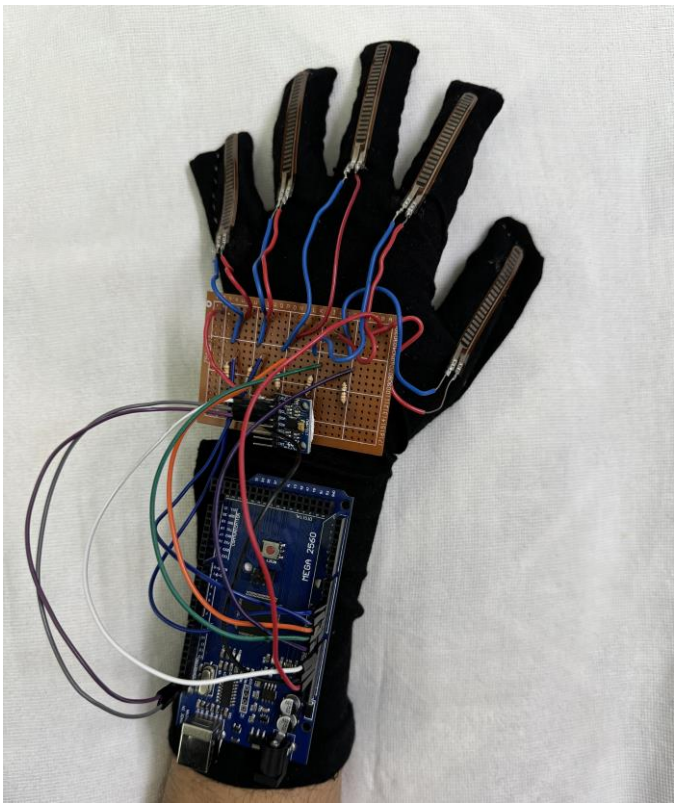

Fig. 1. Design of the smart glove

B. Software Implementation

The flowchart in Fig 2. clearly outlines the software implementation process, providing a detailed visual representation of the process.

1. Dataset Generation

Due to the lack of available and suitable datasets, a custom dataset was created. The authors wore the glove and made different gestures of the ASL. The readings from the MPU-6050 and flex sensors were recorded in a

csv (comma-separated values) file. The process was automated using a Python script that establishes a connection with the Arduino mega board and records the readings of the five flex sensors and the MPU-6050 sensor into a csv file, taking

the readings for 3 seconds per gesture, along with the target gesture. This helps to record dynamic gestures which would take some time to make. Thus, the dataset consists of 11 features (5 from the flex sensors and 6 from the MPU-6050) with 34 classes consisting of letters and words of the ASL. This ensured a diverse range of readings, which helped train the model better. Challenges were faced when there were occasional sensor malfunctions, which led to blank readings being recorded, necessitating re-recording. Fig 3. and Fig 4. depict the static and dynamic gestures which have been included in the dataset respectively, which make up a comprehensive dataset.

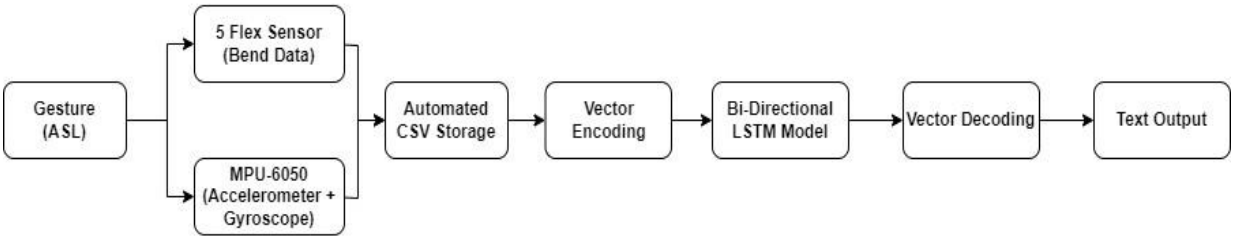

Fig. 2. Software implementation flowchart

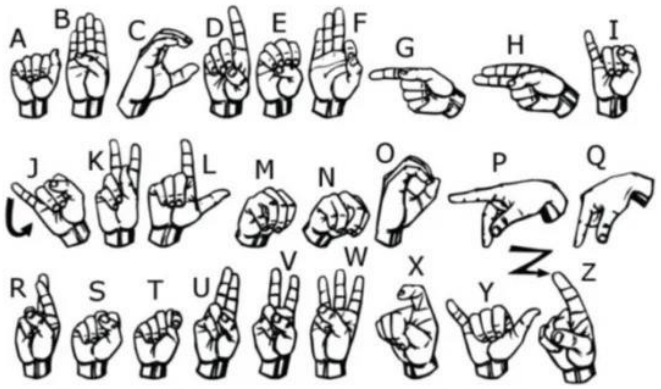

Fig. 3. Static Gestures

Fig. 4. Dynamic Gestures

2. Data Pre-processing

Certain pre-processing steps were performed in the Arduino code itself. Data manipulation was possible due to the Arduino Mega's powerful processing capacity. The raw readings from the flex sensors were scaled within an interval by

using appropriate resistors. The MPU-6050 is very sensitive, with readings that varied due to even air movements. This type of variation increased noise in the data and reduced model accuracy due to the unpredictable nature of the data. It led to inconsistencies in the dataset, making it difficult to scale down, and such outliers resulted in biased predictions. To stabilize the readings from the MPU-6050 sensor, bit-shifting techniques were applied. This process helped reduce sensitivity to minor variations, such as those caused by air movement, ensuring more consistent and reliable data for the deep learning model.

Once the server received the data, standardization was applied, which helped algorithms like gradient descent in the Bidirectional LSTM model to converge faster. This ensures a uniform scale among the features and no one feature dominating the model due to its numeric value.

3. Bi-directional LSTM Model Development

Bi-directional Long Short-term Memory networks are a type of Recurrent Neural Network that improves learning abilities by processing data in both forward and reverse directions. This enables BiLSTMs to capture context in both directions, backwards and forwards. An additional dense layer is incorporated to improve the performance of the model. Softmax is used as an activation function as it is ideal for multi class classification tasks. It converts the raw outputs we get from the dense layer into a probability distribution over all possible classes. This is required for using the sparse categorical cross-entropy loss function as it expects the output to be a probability distribution. Softmax also helps in interpreting the model's predictions. The equation for the activation function is as follows:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

where:

- σ: Softmax function. It converts raw scores into a probability distribution that spans multiple classes.
- z: Input vector. It is the input to the Softmax function.
- e^{zi}: Exponential function on the i-th element of the input vector z.
- K: Number of classes. It represents the total number of classes in the classification problem.
- e^{zj}: Exponential function on the j-th element of the output vector.

After training the model with various combinations, the optimal model configuration was found to be 10 epochs using a batch size of 32. This helps with the learning of patterns in the underlying gesture data by the model while preventing overfitting. The model achieves a training loss of 0.8112 and an accuracy of 84.59%.

4. Testing on New Data

The unseen sensor data from test files are loaded for model evaluation. The test data is pre-processed by scaling and reshaping to ensure consistency with the training data, followed by using the trained model to predict the class labels for the test data. To ensure mapping of dynamic gestures the data for each gesture is recorded for 3 seconds. This results in 10 tuples being recorded in the csv file. These tuples are passed as a sequence to the model which outputs the predicted class. This is done to ensure that the model captures the changes in the gyroscope readings for dynamic gestures. In case of static gestures, no significant change is recorded in the readings.

As seen in Table I, the data from the MPU-6050 and the flex sensors is arranged in the csv file. The features f0 till f4 indicate the readings of the flex sensors placed on the fingers from the thumb to the tiny finger respectively. The values for these features lie in the range of 8-10 when the finger is not bent, and as seen in f4, a lower value indicates bending of the finger. ax, ay, and az represent the readings from the 3-axis accelerometer along the X, Y, and Z axes respectively,

which measures the linear acceleration. gx, gy and gz are measurements of the gyroscope which gives the angular acceleration in the X, Y and Z axes respectively. Since 'W' is a static gesture, the gyroscope readings do not vary.

TABLE I - SENSOR DATA FOR GESTURE W

f0	f1	f2	f3	f4	ax	ay	az	gx	gy	gz
8	7	8	10	4	16	124	16	1	-10	0
8	7	8	10	4	12	123	18	2	-8	0
7	8	7	10	4	14	123	16	1	-11	0
7	7	7	11	4	12	122	16	2	-9	-1
8	7	8	10	4	13	123	16	0	-8	1
8	8	7	11	4	13	122	18	3	-7	-1
7	7	7	10	4	18	124	18	3	-11	1
7	7	7	10	4	15	121	15	2	-11	-1
8	7	7	10	4	15	123	17	2	-11	0
8	7	7	10	4	15	123	17	2	-11	0

IV. RESULTS

After testing the model on unseen samples, the accuracy is 82%. Appropriate evaluation metrics for multi class classification such as accuracy, confusion matrix, ROC scores are employed to test the model. The model has micro-average and macro-average ROC AUC scores of 0.90, indicating high overall effectiveness and consistent performance across all classes. The micro-average score demonstrates the model's strong ability to distinguish instances uniformly, whereas the macro-average score emphasizes its balanced performance across all class sizes. This alignment demonstrates the model's reliability and fairness.

Figure 5 shows the confusion matrix, and Figure 6 depicts the Receiver Operating Characteristic (ROC) curve.

It is noted that the model performs very well for static gestures. The model also predicts dynamic gestures well.

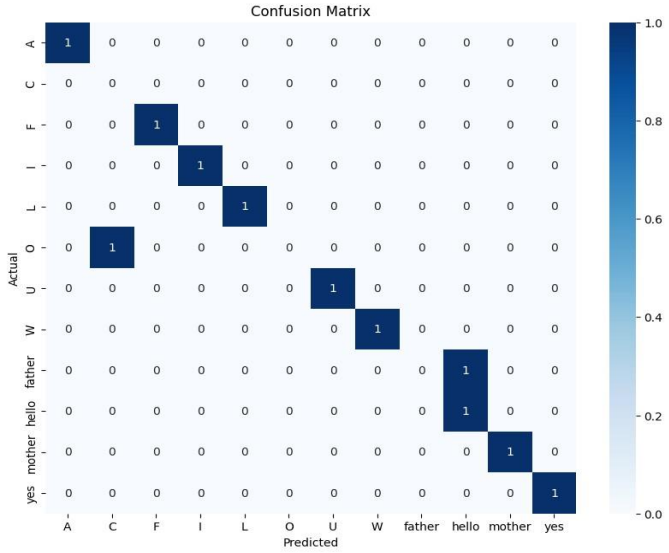

Fig. 5. Confusion Matrix

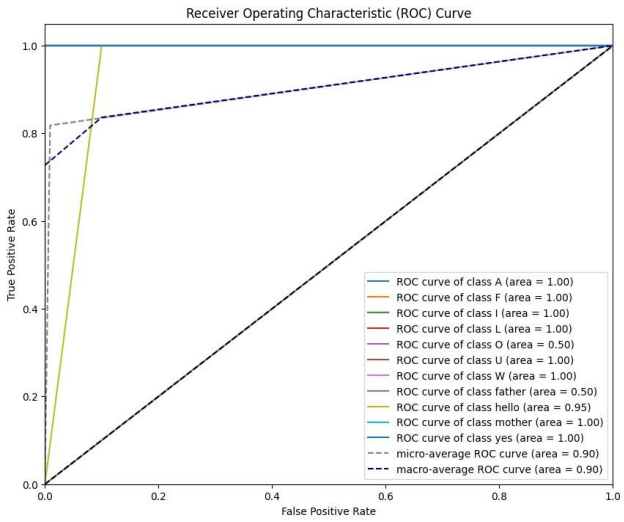

Fig. 6. ROC Curve

V. CONCLUSION

In conclusion, this study advances the field of sign language recognition by developing a smart glove and creating a custom dataset to train a Bi-directional LSTM model. This improvised approach integrates 5 flex sensors and an MPU-6050 sensor with an Arduino Mega 2560, which enables precise capture of finger bends and hand orientation. The hardware setup, coupled with the breathable, lightweight fabric of the glove makes it easy to wear and presentable in everyday use. They created a custom dataset that encompasses a vast range of gestures, including variations in hand orientations, to train the Bi-directional LSTM model. By learning from both forward and backward sequences, the model is capable of enhancing the recognition accuracy by understanding the full context of each gesture. Although the mapping of static gestures is quite accurate, the mapping of dynamic gestures can be improved in the future.

These efforts are driven by the goal of empowering individuals with speech impairments by providing them with a tool for seamless communication. The smart glove system advances gesture recognition technology and underscores a commitment to social inclusion. By bridging the communication gap for sign language users, they aim to foster greater understanding and interaction within diverse communities. In using technology for social good, this study emphasizes the possibility of helping improve the quality of life for people who rely on sign language for communication.

VI.FUTURE SCOPE

Currently the glove is connected to the server via a USB cable, so it can send sensor data directly to the server. However, by integrating either a Wi-Fi module such as the ESP8266 or a Bluetooth module like the HC-05, data can be sent wirelessly to a central server. Making use of a rechargeable battery eliminates the need for a continuous wired connection. Wi-Fi connectivity would allow it to establish a secure online storage connection for sensor data, removing the need for a centralised server. For sign language users, this integration of ubiquitous computing would create lot of opportunities and facilitate easier and more convenient communication.

The glove is currently trained to translate American Sign Language gestures. However, by including diverse sign languages will broaden the vocabulary and foster inclusivity. Additionally, the dataset can be augmented with Indian Sign Language (ISL) gestures, which primarily require two-hand movements. By incorporating ISL gestures and implementing a dual-glove system, these two-hand gestures can be captured as well, improving accuracy and enabling users to convey a broader range of words and expressions. This approach would cater to a wider audience and ensure that users from various linguistic backgrounds can benefit from the device.

Moreover, integrating a speaker with text-to-speech capabilities or a dedicated vocalizer into the system would enhance the system by providing real-time vocal communication. This would allow the glove to convert translated gestures into audible speech. Such a feature would make interactions more effective, especially in environments where visual communication may be challenging.

Accurate and full data collection can be difficult due to occasional sensor malfunctions. Thus, LLMs (Large Language Models) can be used to piece together coherent phrases from a fragmented string of words. These models ensure successful communication even in situations where the input is insufficient by producing meaningful phrases by capturing the context and the relationships between words.

REFERENCES

[1] Anjali Kanvinde, Abhishek Revadekar, Mahesh Tamse, Dhananjay R. Kalbande, and Nida Bakereywala. Bidirectional sign language translation. In *2021 International Conference on Communication information and Computing Technology (ICCICT)*, pages 1–5, 2021.

[2] Faisal Qayoom, N Balaji, S Gurukiran, and SN Sourabh. Hand gesture vocaliser for deaf.

[3] Joseph DelPreto, Josie Hughes, Matteo D'Aria, Marco de Fazio, and Daniela Rus. A wearable smart glove and its application of pose and gesture detection to sign language classification. *IEEE Robotics and Automation Letters*, 7(4):10589–10596, 2022.

[4] Swayam Sa, M Rishitha Chowdary, M Satvika, Kumuda Kalidindi, Sandesh Bj, and P Kokila. Gesture recognition glove for american sign language using accelerometers. In *2023 International Conference on Advancement in Computation Computer Technologies (InCACCT)*, pages 784–789, 2023.

[5] Shahzad Ahmed, Karam Dad Kallu, Sarfaraz Ahmed, and Sung Ho Cho. Hand gestures recognition using radar sensors for human-computerinteraction: A review. *Remote Sensing*, 13(3), 2021.

[6] Deep Kothadiya, Chintan Bhatt, Krenil Sapariya, Kevin Patel, Ana-Belen´ Gil-Gonzalez, and Juan M. Corchado. Deepsign: Sign language detection´ and recognition using deep learning. *Electronics*, 11(11), 2022.

[7] Boon Giin Lee, Teak-Wei Chong, and Wan-Young Chung. Sensor fusion of motion-based sign language interpretation with deep learning. *Sensors*, 20(21), 2020.

[8] Ahmed Abougarair and Walaa Arebi. Smart glove for sign language translation. *International Journal of Robotics and Automation*, 8:109– 117, 12 2022.

[9] Sanish Manandhar, Sushana Bajracharya, Sanjeev Karki, and Ashish Kumar Jha. Hand gesture vocalizer for dumb and deaf people. *SCITECH Nepal*, 14(1):22–29, 2019.