Smart Glove for Word-Level Sign Language Recognition Using Flex Sensors, Accelerometer Sensors, and Long-Short Term Memory Model

Minh Chanh Vo[1] and Trong Nhan Le[2]

Ho Chi Minh University of Technology, Ho Chi Minh City, Viet Nam
vmchanh.sdh242@hcmut.edu.vn
trongnhanle@hcmut.edu.vn

Abstract. Sign language is the easiest way to communicate between deaf people. There have been many studies on sign language recognition by computer vision, which, however, is not privacy-friendly and lacks user-friendliness. So, how can deaf individuals actively convey their meaning to people who do not understand sign language in an easy and practical way? In this paper, we propose a new method that can achieve this. The method involves a wearable device equipped with flex sensors and an accelerometer, combined with an LSTM model to recognize the words of people who are unable to speak. The model was trained and tested with 15 sign languages, including static and dynamic gestures. Finally, this project has been successfully implemented with 98% accuracy based on a training dataset recorded by the author within one hour and is capable of translating 15 word-level sign languages into speech in real time.

Keywords: Word-level Sign Language, Flex Sensors, Accelerometer sensor, LSTM model

1 Introduction

In daily life, people communicate with each other through speech to convey opinions. However, for those with hearing and speech impairments, it is very difficult to express themselves vocally. They communicate using hand gestures, facial expressions, and body movements. Therefore, communication between individuals without impairments and those with congenital hearing loss can be challenging. Gestures in sign language follow their own rules and syntax, and learning them requires significant time and effort to understand the intended messages. Furthermore, the same gesture can be interpreted differently depending on the movement of various body parts. There are three main types of variations in sign language [1].

- Non-manual features: Tongue, facial expressions, body poses, and hand gestures – all of them are used to communicate.
- Word-level sign spelling: Each gesture represents a whole word.

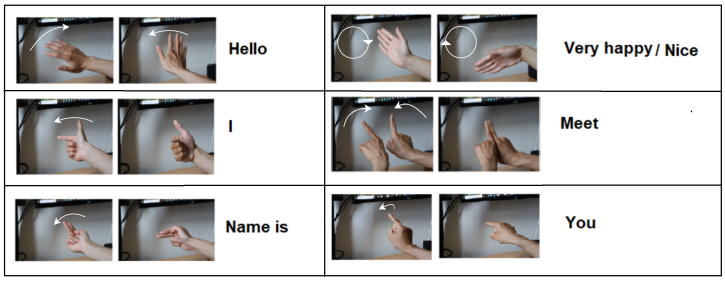

Fig. 1: Example of word-level sign language

– Finger vocabulary: One gesture represents one alphabet/number.

In this system, a flex sensor is placed on each finger to measure the degree of bending, while an accelerometer sensor is mounted on the back of the hand to capture the direction and motion of the hand. Each sensor has its own advantages and methods for capturing different aspects of a gesture. Combining these two measurements enhances the efficiency of gesture recognition. This paper proposes a translation glove system for real-time word-level sign language recognition by integrating data from both the accelerometer and flex sensors. Although smart gloves have been researched previously [2,3,4], to the best of our knowledge, no prior work has combined flex sensors, an accelerometer sensor, and an LSTM model to recognize word-level sign language. In our approach, sign language words are carefully analyzed based on the information received from the sensors. We then use the LSTM machine learning algorithm to classify and translate word-level sign language into speech in real time.

2 Related Work

There have been studies on gesture recognition. Currently, there are two main approaches. One involves using wearable devices, and the other uses 2D and 3D recording devices to detect hand movements.

Camera-based recognition: In the study [5], a depth camera and RNN algorithm were used to recognize Mexican Sign Language through hand gestures, body movements, and facial expressions used to convey messages. Recognition of Bangla Sign Language was also achieved in [6], which used the Esharalipi dataset and a CNN (Convolutional Neural Network) algorithm, achieving 95% accuracy. Similarly, the previously published study [1] used the YOLO v5 algorithm to identify 36 distinct gestures in the American Sign Language system and achieved a 95% accuracy rate on the MU_HandImages_ASL dataset.

In general, camera-based recognition is limited by the range and the manner in which the deaf convey their message. Therefore, in daily life, it is not convenient to set up cameras in desired positions, and privacy concerns cannot be guaranteed.

Device-based recognition: In [7], a wearable system was used to recognize sign language in real time by utilizing IMU and surface EMG sensors to measure electrical activity generated by skeletal muscles and hand movements. Classification algorithms such as Naive Bayes, Nearest Neighbor, Decision Tree, and LibSVM were then applied to identify the gestures. In the same study [3], conductive textile fabric was used to measure finger bending levels, and an IMU was also used to measure the acceleration of the back of the hand. In [8], a system using flex sensors and accelerometers combined with a matching algorithm was used to recognize basic letters of the Vietnamese alphabet.

Overall, sign language relies heavily on the movement and shape of the hands. As a result, when data is extracted from motion and muscle sensors, the recognition accuracy is significantly high. Therefore, if the goal is to transmit messages while maintaining privacy, accuracy, and user-friendliness, this approach is a highly suitable choice for sign language translation.

3 Background

3.1 Word-level Sign Language

In a word-level sign language system, each gesture corresponds to a specific word or concept and is usually expressed through one or two hands combined with movement in space.

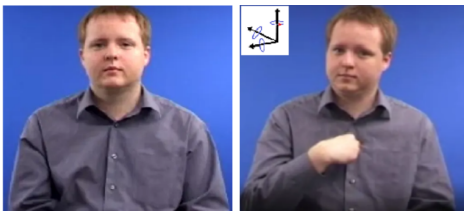

Fig. 2: Representation of "I" in sign language [9]

Examples of "I" in ASL (American Sign Language), a personal pronoun "I" expressed by pointing to the chest. This is a word-level gesture. A "I" gesture language is expressed through each finger as follows: The index finger is kept straight while the other fingers are bent. The axis perpendicular to the back of the hand's plane will rotate around its own vertical axis. This simple example illustrates that each gesture can be broken down into hand and finger movements. But how can these movements be digitized and interpreted by machines? This question leads to the use of hardware such as flex sensors, which can measure the amount of flexion of each finger, and angular accelerometers, which can determine the acceleration of the hand in space.

3.2 Flex Sensor

A flex sensor, also known as a bend sensor, can change its resistance based on the degree of bending. The conductive ink on the sensor functions as a type of resistance. When the sensor is straight, this resistance is approximately 25k Ohm. When the sensor is bent, the conductive ink layer is stretched, leading to a reduced cross-sectional area and an increase in resistance. At a 90° bend, this resistance reaches around 100k Ohm. When the sensor is straightened again, the resistance returns to its original value. By measuring the resistance, we can determine the degree of bending of the sensor. The sensor's output signal is in analog form.

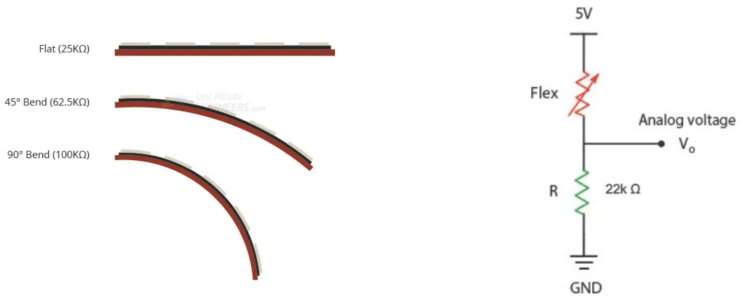

Fig. 3: The resistance varies with the degree of bending (left) and Flex sensor schematic (Right)

Flex sensors commonly come in two standard sizes: 2.2 inches (5.588 cm) and 4.5 inches (11.43 cm). Thus, based on this capability and choosing a sensor size that matches the length of the finger, we can apply this principle to measure the specific degree of bending for each individual finger.The simplest way to read a flex sensor is to pair it with a fixed resistor to form a voltage divider, thereby generating a variable voltage that can be read by the microcontroller's analog-to-digital converter (ADC).

$$\mathtt{VO} = \frac{\mathtt{Vcc} * \mathtt{R}}{\mathtt{R} + \mathtt{Rflex}}$$

In this case, the output voltage decreases as the bending radius increases.

3.3 Accelerometer Sensor

The accelerometer sensor is used to measure changes in the movement direction of the back of the hand. The angular accelerometer is capable of measuring rotational velocity around three axes in the Cartesian coordinate system. This sensor operates based on the gyroscopic principle and can output specific digital

signals for each axis. According to the manufacturer's documentation [10], This signal is digitized using individual on-chip 16-bit analog-to-digital converters to sample each axis via I2C communication.

Fig. 4: Orientation of Axes of Sensitivity and Polarity of Rotation (Left) and Triple-Axis Digital-Output Gyroscope ITG3200 (Right)

3.4 Embedded circuit

To measure the output signals and filter out noise, a dedicated microcontroller is required to read these values. Therefore, we used the Arduino Lilypad due to its similarity to common Arduino boards such as the Arduino Nano and Arduino Mega 2560, while offering a more suitable size and pin layout for this glove design. A real-time noise filtering algorithm, the Kalman filter, is then applied to eliminate unwanted values and extract key data features.

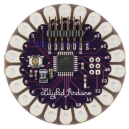

Fig. 5: Arduino lilypad

3.5 Long Short Term Memory model

The LSTM model is a variant of the RNN model used to process sequential data. This model solves the problem of vanishing gradient or exploding gradient during training by storing information through cell state and gates. Thus, the LSTM model over RNNs is its ability to learn and remember long input sequences. The working process of LSTM is described in the following Figure 6. Parameters are entered into the Forget gate to determine the information to be discarded, the Input gate will determine the new information to be added and then update the state of the cell and the Output gate will determine the current output and then pass it to the next cell. Another benefit of the LSTM model is its ability to classify sequences and learn directly from raw time-series data.

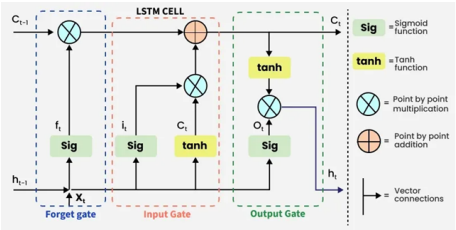

Fig. 6: LSTM model

4 Methods

4.1 Architecture Design

The system is designed with two modules, each serving a specific function:
Gesture Data Collection Module: The input for this module consists of gesture data collected from flex sensors and an accelerometer sensor. Once collected, the data is packaged and sent to the Gesture Recognition Module.
Gesture Recognition Module: The data received from the collection module is processed, classified, and reshaped to be fed into an LSTM model. The output of the LSTM model is the predicted gesture in text form, it can be customized and then converted into speech through a Text-to-Speech (TTS) component.

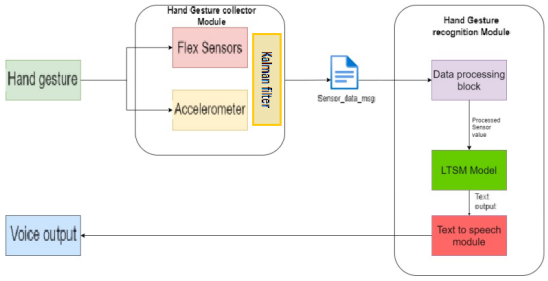

Fig. 7: System Architecture

4.2 Hardware connection

The signals from the flex sensors and gyroscopic sensors are then acquired by the Arduino microcontroller using the following circuit design. This setup also includes a monitor and a bluetooth module, allowing for future development of wireless features and providing a more convenient way to monitor results. Due

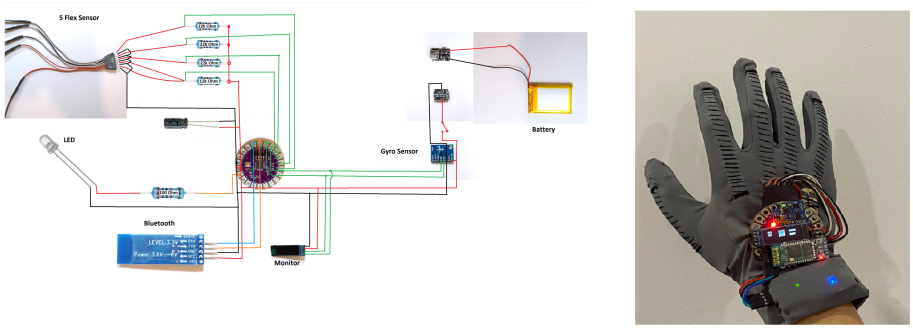

Fig. 8: Schematic and completed glove

to hardware limitations, we implemented 4 outputs from the 5 flex sensors. The thumb and index finger were connected in parallel through a 12k resistor. Within the scope of this project, there are not many actions involving overlapping gestures between these two fingers. The system produces 7 signals in total: 3 signals from the accelerometer and 4 signals from the flex sensors. These signals are then passed through a Kalman filter to remove noise and normalized to a range from 0 to 100.

Figure 9 shows the change in signal when the hand moves. When the fingers are clenched, the hand rotates clockwise, the data read from the X-axis angular acceleration sensor increases, then decreases and returns to the original value, the Y and Z axes are kept the same, the accelerations on these axes do not change too much. In addition, the flex sensors are all bent, so the value will increase to a value above 70 at the outputs flex 0,1; flex 3; flex 4 and above 60 for flex 2

When the fingers are opened, the hand rotates counterclockwise, the angular acceleration on the X axis decreases rapidly and increases again, the Y and Z axes do not have obvious changes, the flex sensor is straightened, so the value is below 70 at the output of flex 0.1; flex 3; flex 4 and below 50 for flex 2

In summary, the behaviors on each finger, from grasping to extending and the movement of the hand have been digitized through the sensors. Finally, the sentence “I’m very happy to meet you” will generate the following signals: .

.

.

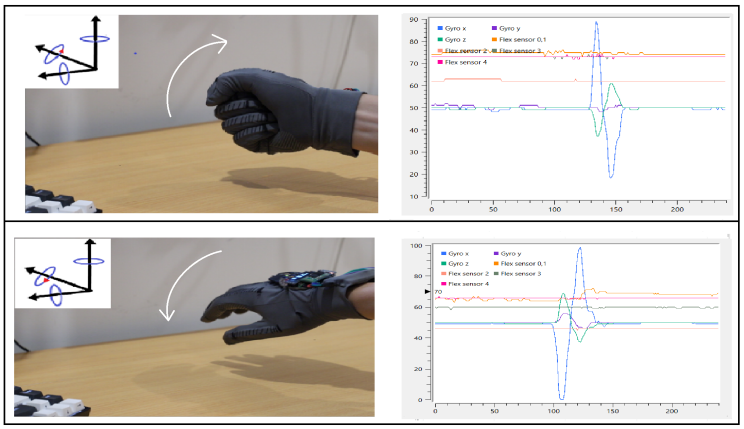

Fig. 9: IMU and Flex sensor signal

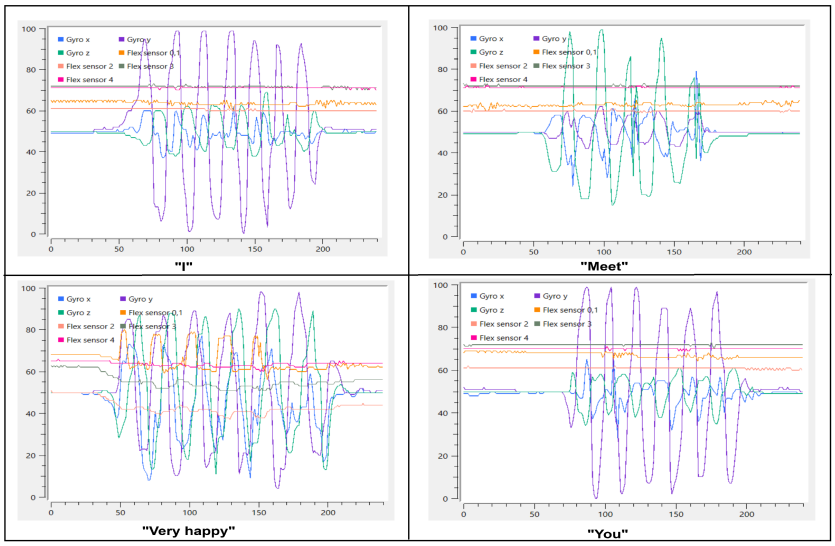

Fig. 10: Word-level Sign Language Signal

.

4.3 Labeling and Training

To recognize these oscillation patterns, we referred to the article [11]. By applying this model, we successfully created a word-level sign language recognition system.

.

Listing 1.1: LSTM Model Layer

```
model.add(LSTM(100, input_shape=(n_timesteps, n_features)))
model.add(Dropout(0.5))
model.add(Dense(100, activation='relu'))
model.add(Dense(n_outputs, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam',
    metrics=['accuracy'])
```

Then, we created a training dataset and a testing dataset by recording the actions described in the table 1.

Label	Word	Description
0	Hold	Fingers clenched into a fist
1	Goodbye	The back of the hand oscillates around the Z-axis at a low frequency
2	Good	The hand forms a shape similar to the "like" icon.
3	I	Pointing to oneself, thumb pointing upward, back of the hand rotates around the Y-axis.
4	Do not know	Rotates around the middle finger axis (X-axis of the accelerometer)
5	"Nothing"	Resting state, hand remains inactive
6	You	Other fingers folded inward, back of the hand oscillates around the Y-axis of the gyroscope
7	Walk	Alternating motion of index and middle fingers, thumb extended
8	Around	Index, middle finger, and thumb extended, back of the hand rotates around the forearm axis
9	Kidding	Simultaneous flexing of index and middle fingers, remaining fingers folded inward
10	Here	Index and middle fingers bent at 90 degrees, thumb extended, back of the hand oscillates around the Y-axis of the sensor
11	Hello	Fingers extended, back of the hand oscillates around the Z-axis at a high frequency.
12	Name is	Index finger and thumb bent at 90 degrees, remaining fingers folded inward, back of the hand oscillates around the X-axis
13	So happy	Fingers extended, back of the hand rotates around the forearm axis
14	Meet	Only the index finger raised, back of the hand oscillates around the Z-axis.

Table 1: Dataset description

To reduce the load on the prediction system while still extracting data features, we designed a recording system with a frequency of 100Hz (one hundred data points per signal recorded each second). We observed that each action, from start to finish, had an average recognition time of 2.4 seconds. Thus, each

data recording includes 100x2.4x7 data points, with 100x2.4 data points for each sensor and 7 output signals for the variables being recorded.

5 Result

The dataset was collected with a size of 100×2.4×7×1020 (frequency*recognition time*output signals*samples) data points per gesture. In total, there are 15300 data samples for the training set (70% ratio) and 4500 samples for the testing dataset. After being fed into the model for training with a 70% ratio for training and 30% for testing. the results are shown as follows.

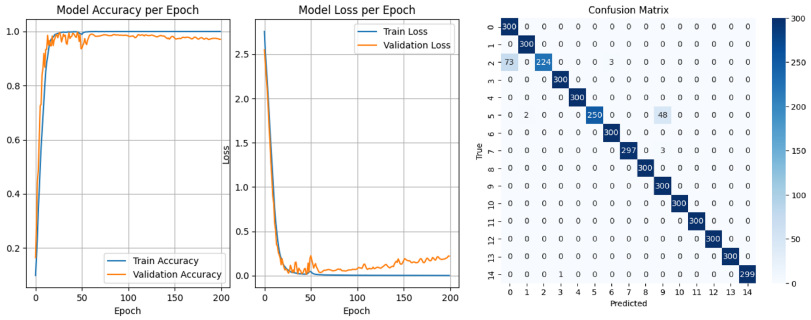

Fig. 11: Evaluation Graph

Class	Precision	Recall	F1-score	Support
Accuracy		0.9820		4500
Macro Avg	0.9831	0.9820	0.9818	4500
Weighted Avg	0.9831	0.9820	0.9818	4500
F1 Score (macro)	0.9818			
Recall (macro)	0.9820			

Table 2: Classification Report

Figure 11 illustrates the training results on the collected dataset. The model converged rapidly within the first 30 epochs and maintained stable performance throughout the subsequent epochs. The average test accuracy reached 98.20%, and the loss gradually decreased towards zero. The model also achieved a macro-average F1-score of 0.9818 and a macro-average recall of 0.9820 in Table 2.
Despite the overall strong performance, the confusion matrix reveals notable misclassifications:
- For class 2, the gesture 'like' (well done) involves all fingers being bent while

only the thumb is extended, whereas in class 0 all fingers, including the thumb, are bent. From the hardware perspective, the thumb's signal is transmitted through a sensor where the resistance is calculated using the parallel formula $R = \frac{R_1 \times R_2}{R_1 + R_2}$ is small (when the sensor is bent), the variation in R1 becomes minimal. As a result, the model fails to extract distinctive features from this gesture, leading to misclassification.

For classes 5 and 9, the misclassification was influenced both by suboptimal data collection and hardware constraints. In class 5, all fingers are relaxed with a slight tendency to bend, while in class 9, the thumb, index, and little fingers remain relaxed in a similar way to class 5, but the middle and ring fingers flex simultaneously. Since each gesture was recorded continuously for two minutes during dataset collection, muscle fatigue gradually reduced this variation, causing signals of class 9 to become more similar to those of class 5.

Nevertheless, the results demonstrate that the model has effectively learned the data features, with no clear signs of overfitting.

Our goal was for the project to operate in real-time. Therefore, we exported the model and performed real-time predictions. By recording the data and feeding it directly into the system, within 2 minutes and 49 seconds, the model, based on 15 pre-trained sign languages, combined with text-to-speech conversion, and then customized for personal communication by configuring text file, it was able to detect two complete sentences: "Hello, I, name is Chanh, very happy to meet you. Goodbye." and "I walk around here. Goodbye."

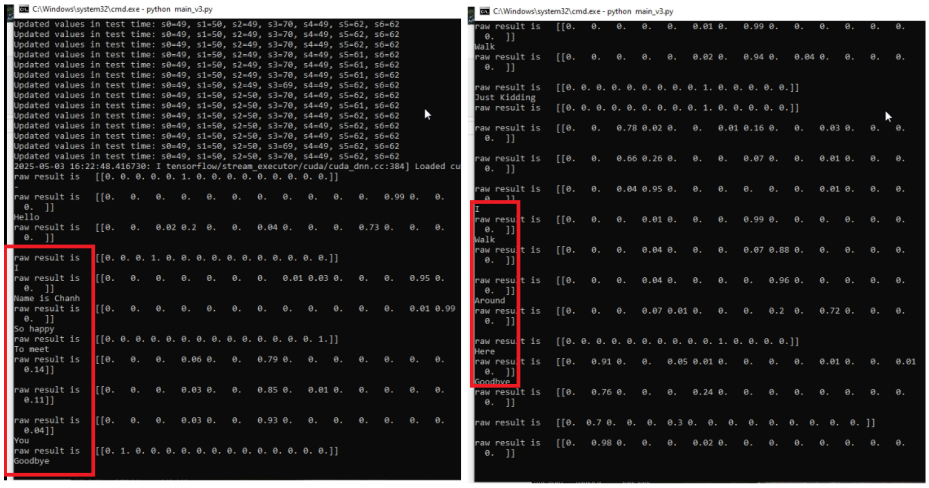

Fig. 12: Result

6 Discussion

This project addresses the limitations of traditional recognition models, which typically rely on computer vision—meaning a device must capture images and send them to a computer for processing. This requirement restricts the ability of speech-impaired individuals to communicate freely. Moreover, the project is able to support both digit and word-level in sign language. Therefore, with a large enough training dataset, the system has the potential to evolve into a new recognition framework—a novel method for language translation.

Limitations and future work :
- The current device accurately captures the accelerated movements of the back of the hand but remains limited in recognizing static postures and spatial orientation. In future work, the system will be enhanced with additional sensors and algorithms to determine the static position of the hand relative to the body, thereby significantly expanding the vocabulary that the model can learn.
- In this study, only four pins were available for five flex sensors, which led to misclassification of certain gestures. In future research, this issue can be easily addressed by employing microcontrollers such as the ESP32 or STM32, which provide a significantly greater number of analog pins compared to the Arduino LilyPad
- According to the authors' investigation, glove-based datasets are neither widely available nor standardized in terms of hardware, making it difficult to employ strategies such as transfer learning to expand the vocabulary. In future work, the authors plan to adopt an incremental learning approach, which allows users to independently train and adapt models to their specific needs. Subsequently, a multi-user database will be established to collect and share datasets across users, thereby enriching and diversifying the overall vocabulary.

Overall, the project focuses on basic movements of the fingers and the back of the hand. However, sign language involves a much wider range of expressions, including full-body gestures, both arms, and facial expressions. Nevertheless, this project lays the groundwork for expanding and enhancing this translation model in the future.

7 Conclusion

The results demonstrate that the model performs well in making predictions. In some real-time prediction scenarios, the model produces highly confident probability outputs, provided that the training dataset is sufficiently comprehensive. Additionally, the prediction speed is remarkably fast. These outcomes highlight the potential of developing sign language translation devices using machine learning. Therefore, this work provides an optimal foundation for building models that apply lightweight AI in sign language recognition systems.

References

1. Dima, T.F., Ahmed, M.E.: Using yolov5 algorithm to detect and recognize american sign language. In: 2021 International Conference on Information Technology (ICIT). pp. 603–607 (2021)
2. N, A., R, V., B, M., S, A.K., S, S.: Flex sensor dataset: Towards enhancing the performance of sign language detection system. In: 2022 International Conference on Computer Communication and Informatics (ICCCI). pp. 01–05 (2022)
3. DelPreto, J., Hughes, J., D'Aria, M., de Fazio, M., Rus, D.: 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. Praveen, N., Karanth, N., Megha, M.S.: Sign language interpreter using a smart glove. In: 2014 International Conference on Advances in Electronics Computers and Communications. pp. 1–5 (2014)
5. Mejía-Peréz, K., Córdova-Esparza, D.M., Terven, J., Herrera-Navarro, A.M., García-Ramírez, T., Ramírez-Pedraza, A.: Automatic recognition of mexican sign language using a depth camera and recurrent neural networks. Applied Sciences 12(11) (2022)
6. Islam, S., Sultana Sharmin Mousumi, S., Rabby, A.S.A., Hossain, S.A., Abujar, S.: A potent model to recognize bangla sign language digits using convolutional neural network. Procedia Computer Science 143, 611–618 (2018), 8th International Conference on Advances in Computing & Communications (ICACC-2018)
7. Wu, J., Sun, L., Jafari, R.: A wearable system for recognizing american sign language in real-time using imu and surface emg sensors. IEEE Journal of Biomedical and Health Informatics 20(5), 1281–1290 (2016)
8. Phi, L.T., Nguyen, H.D., Bui, T.Q., Vu, T.T.: A glove-based gesture recognition system for vietnamese sign language. In: 2015 15th International Conference on Control, Automation and Systems (ICCAS). pp. 1555–1559 (2015)
9. Academy, L.: Swearing in sign language – video image included. https://lead-academy.org/blog/swearing-in-sign-language/, last accessed: 2025-August-20
10. InvenSense: ITG-3200 Product Specification (10 2010), rev. 1.4
11. Brownlee, J.: Lstms for human activity recognition time series classification. https://machinelearningmastery.com/how-to-develop-rnn-models-for-human-activity-recognition-time-series-classification (2022), last accessed: 2025-May-06