

Article

Not peer-reviewed version

---

# Novel Wearable System to Recognize Sign Language in Real Time

---

[Ilhan Umut](#)\* and [Ümit Can Kundereli](#)

Posted Date: 30 May 2024

doi: 10.20944/preprints202405.2038.v1

Keywords: Artificial intelligence; Computer software; Human-computer interaction; Inertial measurement unit; Sign language recognition; Surface electromyography



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

# Novel Wearable System to Recognize Sign Language in Real Time

İlhan Umut <sup>1,\*</sup> and Ümit Can Kundereli <sup>2</sup>

<sup>1</sup> Department of Electronics and Automation, Corlu Vocational School, Tekirdag Namik Kemal University, 59850, Tekirdag, Turkey; iumut@nku.edu.tr

<sup>2</sup> Department of Computer Engineering, Faculty of Engineering, Trakya University, 22030, Edirne, Turkey; ucankundereli@trakya.edu.tr

\* Correspondence: iumut@nku.edu.tr

**Abstract:** The aim of this study is to develop a software solution for real-time recognition of sign language words using two arms. This will enable communication between hearing-impaired individuals and those who can hear. Several sign language recognition systems have been developed using different technologies, including cameras, armbands, and gloves. The system developed in this study utilizes surface electromyography (muscle activity) and inertial measurement unit (motion dynamics) data from both arms. Other methods often have drawbacks, such as high costs, low accuracy due to ambient light and obstacles, and complex hardware requirements, which have prevented their practical application. A software has been developed that can run on different operating systems using digital signal processing and machine learning methods specific to the study. For the test, we created a dataset of 80 words based on their frequency of use in daily life and performed a thorough feature extraction process. We tested the recognition performance using various classifiers and parameters and compared the results. The Random Forest algorithm was found to have the highest success rate with 99.875% accuracy, while the Naive Bayes algorithm had the lowest success rate with 87.625% accuracy. Feedback from a test group of 10 people indicated that the system is user-friendly, aesthetically appealing, and practically useful. The new system enables smoother communication for people with hearing disabilities and promises seamless integration into daily life without compromising user comfort or lifestyle quality.

**Keywords:** Artificial intelligence; Computer software; Human-computer interaction; Inertial measurement unit; Sign language recognition; Surface electromyography

## 1. Introduction

The World Health Organization reports that around 466 million individuals have hearing impairments, with this number expected to rise to 700 million by 2050 [1]. Sign language uses hands, facial movements, and body posture to express thoughts, feelings, and information instead of verbal communication. There are several types of sign languages with alphabets and signs used worldwide. Turkish Sign Language, like other sign languages, has a different word order and grammar than Turkish. Although hearing-impaired individuals use sign language to communicate with each other, they may face difficulties when interacting with others. As a result, the prevalence of psychological problems among them is high [2]. Writing and reading are common methods of communication between hearing people and the deaf. Writing is the most effective method of communicating with a deaf person, particularly when accuracy is crucial. However, in some situations, writing may not be feasible. Speechreading is another method of communication, but many spoken Turkish sounds do not appear on the lips. Therefore, a real-time sign language recognition (SLR) system is necessary to translate signs into sound or text.

Although the focus of our paper is on SLR, most of the work in this area consists of hand gesture recognition. Hand gestures are used to express only letters and numbers. However, communicating in this way can be slow and difficult. It is much faster and easier to sign words using hand, arm, and

facial expressions instead of individual letters [3]. Studies have been conducted on the subject and solutions have been developed. The majority of these studies have focused on specific sign languages, particularly American Sign Language (ASL) and Chinese Sign Languages (CSL). Various technologies have been employed, including Microsoft Kinect [4,5], leap motion [6–11], data gloves [12,13], cameras [14–25], surface electromyography (sEMG), and inertial measurement unit (IMU) [26–36]. Although some systems have achieved high levels of accuracy, none are likely to be suitable for everyday real-life situations. Refer to Table 1 [37] for a comparison of current technology-based systems.

**Table 1.** Comparison of existing it based solutions.

<b>Device/ Technology</b>	<b>Accuracy (around)</b>	<b>Mobility</b>	<b>User Convenience</b>
Kinect	90%	Not a mobile solution	User must stand in front of the sensor
Data Glove	80%	Not a mobile solution	User must wear device
Leap Motion	95%-98%	Not a mobile solution	User must stand in front of the sensor
Image Processing	90%	Not a mobile solution	User must stand in front of the camera

Motion data in image-oriented techniques is acquired through a camera, while solutions using data gloves acquire position, direction, and velocity information of movement through sensors. Although most developers prefer image-based techniques for gesture recognition, these techniques have several disadvantages. For instance, gestures are highly dependent on perspective, which can cause different gestures to appear the same to a poor-quality passive camera. Furthermore, image-based methods necessitate consistent lighting and fixed camera placements, while also requiring high processing power, memory, and energy. Although Leap Motion, a different visual-based method, has advantages over cameras, it remains theoretical since it must be in a fixed position.

Methods such as depth-based method, skeleton-based method and RGB can be given as examples of camera-based solutions. Depth-Based Method uses depth-sensing cameras, such as Microsoft Kinect, to capture the three-dimensional position of the hands and body. Depth cameras provide data that helps in distinguishing the hands and fingers from the rest of the body and the background by measuring the distance of objects from the camera. This spatial data is crucial for accurately recognizing sign language gestures that involve complex hand shapes and motions. Skeletal-based systems create a virtual skeleton of the signer's body, particularly the hands and arms, from video data. These systems often use machine learning algorithms to interpret the movements of these skeletal points over time. By analyzing the positions, movements, and angles of joints, these systems can recognize specific signs based on predefined gesture patterns associated with sign language. RGB-based recognition uses standard RGB (red, green, blue) color video data to detect and analyze sign language. This method processes the visual data from conventional video cameras without the need for specialized depth sensors. The challenge with RGB-based systems is accurately isolating the hands and fingers from the rest of the scene, especially in complex backgrounds. Advances in image processing and artificial intelligence are improving the efficacy of RGB-based sign language recognition. Each method has its strengths and limitations. Depth-based and skeletal-based methods are generally more robust against complex backgrounds but require specialized hardware. In contrast, RGB-based methods are more accessible as they can use standard camera equipment but may struggle with accuracy in diverse settings.

In solutions where sensors are used, the user wears gloves or armbands containing various sensors to detect movements instead of relying on images. Fels et al. [38] are one of the pioneering teams in glove-based SLR methods. They achieved good classification success (94%) by using artificial neural networks to classify 203 ASL signals. However, wearing gloves is uncomfortable under normal conditions, and technical obstacles limit the use of this method to laboratory environments. When comparing armbands and gloves, it is important to note that armbands can help reduce sweating and provide a more aesthetically pleasing appearance. They also require fewer sensors and can communicate wirelessly. Furthermore, armband systems that use sEMG and IMU

technology are currently popular due to their affordability and portability. IMU sensors are specifically used to capture arm movements. sEMG sensors, on the other hand, are suitable for acquiring movement information of hands and fingers [39]. For instance, although some signs in Turkish Sign Language (TSL) share similarities in hand and arm movements, the movements of the fingers differ. This makes it challenging to differentiate between IMU data alone and these movements. However, with the aid of sEMG data, finger movements can also be distinguished, enabling the classification of signs. In a study on CSL recognition, accelerometers and sEMG sensors were employed together. The study employed a hierarchical decision tree as the classification algorithm and achieved 95.3% accuracy [40]. The study most similar to ours was conducted by Wu et al. [41]. They developed a system that utilized a wrist worn IMU and four sEMG sensors to recognize 80 different ASL signs. The study achieved an average accuracy of 96.16% using a support vector machine classifier. However, their hardware was a prototype and they did not have software like ours, which can obtain and classify data simultaneously. Therefore, their study remained theoretical.

sEMG and IMU sensors can be produced as wristbands [39] or armbands [42] as they are compact, inexpensive, light and consume less energy.

The use of wearable technologies, particularly smartwatches, is increasingly common. These devices have various applications, including motion recognition [39], SLR [40], and prosthetic control [43]. The purpose of this study is to propose software for real-time SLR using two Myo Armbands. In our research, we did not find a comparable system appropriate for daily use.

The proposed system offers key contributions in terms of practicality, affordability, and mobility. Unlike many current technologies that are limited to controlled environments, this system is designed to be used in everyday real-life situations. The system utilizes Myo armbands, which are compact, light, and less costly compared to other technologies, enhancing mobility and reducing user inconvenience associated with more cumbersome devices. The software achieves a high accuracy rate of 99.875% with the Random Forest algorithm, thanks to its utilization of advanced digital signal processing and machine learning techniques. User feedback confirms that the system is both aesthetically appealing and user-friendly. Furthermore, the software is compatible with multiple operating systems and languages. The software is compatible with a range of operating systems, including Windows and Android, and will soon expand to Linux and iOS. The software is compatible with a range of operating systems, including Windows and Android, and will soon expand to Linux and iOS. This approach promotes inclusivity and accommodates various user preferences. The software is compatible with a range of operating systems, including Windows and Android, and will soon expand to Linux and iOS. Users can also create their own dictionary within the application, adding a personalized aspect and allowing for a tailored user experience based on individual needs and preferences.

## 2. Materials and Methods

### 2.1. System Hardware

Data from the arm was collected using two Myo Armbands from Thalmic Labs, each consisting of an IMU and sEMG sensors [44]. The device is depicted in Figure 1.



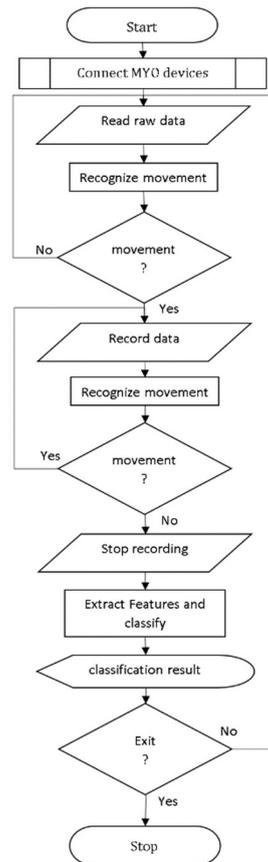
**Figure 1.** Myo Armband.

Myo is a commercial motion controller that contains an ARM Cortex-M4 120 MHz microprocessor, eight dry sEMGs with a sampling rate of 200 Hz, and a nine-axis IMU with a sampling rate of 50 Hz. The IMU provides ten different types of data. The accelerometers in the device measure acceleration in terms of the gravitational constant (G), while the gyroscope measures rotation in terms of rad/s. The device includes magnetometers to obtain orientation and positioning measurements, which are used to determine the movement of the arm. Additionally, the device has eight dry sEMG sensors that allow for the detection of finger movements.

The Myo armband was the ideal choice for the study outlined in the manuscript due to its exceptional features that make it particularly suited for real-time sign language recognition (SLR). The Myo armband was selected for its integrated sensors, including both inertial measurement unit (IMU) sensors and surface electromyography (sEMG) sensors. The IMU sensors capture arm movements, while the sEMG sensors detect muscle activity related to hand and finger movements. This combination provides comprehensive data necessary for accurate gesture recognition. The Myo armbands are wireless and highly portable, allowing for practical, everyday use outside of laboratory environments. Additionally, the armbands enable real-time processing of the captured data. The Myo armband processes data in real-time, making it ideal for translating sign language into text or speech with minimal delays. Its sensors are sensitive and precise, ensuring high accuracy in gesture recognition. The study's results demonstrate that the Random Forest algorithm achieved an accuracy rate of 99.875%. The Myo armband design is both user-friendly and comfortable to wear, which is crucial for ensuring user acceptance and continuous use. It can be effortlessly put on and taken off and does not necessitate the user to handle complex setups. In comparison to other motion capture devices such as data gloves and camera-based systems, Myo armbands are a cost-effective option. This technology is highly accessible and practical for large-scale deployment, thanks to its enhanced accessibility. The Myo armband boasts a robust Application Programming Interface (API) that empowers the creation of tailored applications, such as the SLR system mentioned in the manuscript. This support allows developers to personalize applications to suit the unique demands of their projects. The Myo armband is an excellent choice for developing a sign language recognition system that is effective and practical for everyday use. It addresses many of the limitations found in previous technologies.

## *2.2. Developed Software*

The application was developed using the Delphi programming language and Embarcadero RAD Studio 11 Integrated Development Environment. Two versions were created for Windows and Android platforms, with the possibility of creating additional versions for Linux and IOS platforms using the same code. SQLite was used as the database. Figure 2 presents a flowchart outlining the sequential steps of the proposed methodology for recognizing sign language using the Myo armband.

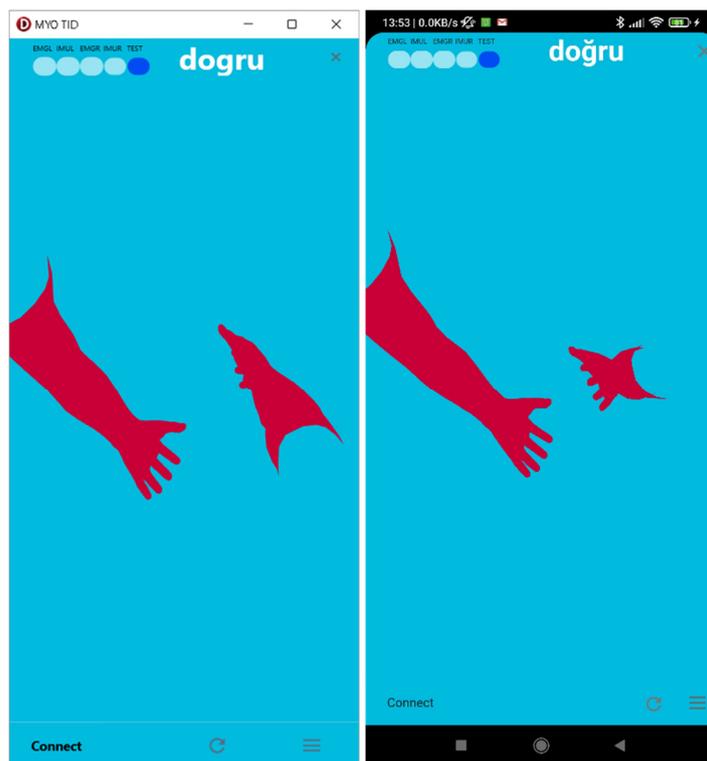


**Figure 2.** Flow chart of the software.

The program has 5 different user interfaces.

### 2.2.1. Main User Interface

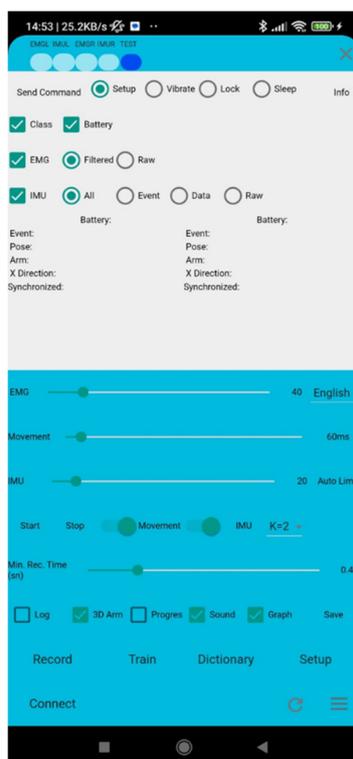
Upon running the program for the first time, the user interface depicted in Figure 3 will appear. The interface is designed with simplicity in mind. Pressing the Connect button triggers the program to check for location and Bluetooth permissions. If permission is not granted, the program will request it from the user. If permission is granted but Bluetooth connection or location access is turned off, the program will turn them on. Finally, the program will establish a connection to both Myo Armbands using the Bluetooth Low Energy (BLE) protocol. After establishing the connection, the label and oval-shaped markers on the left of the screen appear if the EMG and IMU services are successfully subscribed. These markers turn blue when there is a change above the threshold value in the data. The blue oval indicator under the TEST label indicates that the program can make predictions from instantaneous movements. The guessed word is displayed in written form in the upper middle of the screen. The color transition in the three-dimensional arms can be used to observe changes in the EMG data of each arm. The battery status of each armband is displayed as a number ranging from 0 (empty) to 100 (fully charged) in the upper left and right corners of the interface. To return the three-dimensional arms to their reference position, click the refresh icon in the lower right corner. Access the Settings interface by clicking on the three-line icon located in the lower right corner.



**Figure 3.** An example of the main graphical user interface. Windows on the left, Android on the right.

### 2.2.2. Settings User Interface

The interface (Figure 4) allows access and modification of all device and software settings. It is possible to adjust the device's data, vibration, lock, sleep, and wake functions, as well as change the language. Currently, Turkish and English are available, with more languages to be added in the future.



**Figure 4.** Example of a graphical user interface for settings.

Precise data segmentation settings can be adjusted individually. For segmentation, each EMG and IMU data is stored in a global variable. Motion is detected by calculating the difference between the current and previous measurements of 200 Hz (EMG) and 50 Hz (IMU) data. The user can adjust the limit values for this in the settings interface to suit their needs.

The maximum value of the EMG Trackbar is 256, which corresponds to the range of EMG data from the device (-127 to 128). In the example interface, this value is set to 40. Therefore, any absolute change of 40 in the values of the eight EMG sensors indicates motion.

The pause between two signals can be adjusted between 20 ms and 1 s (1000ms). If you wish to speak quickly, you can select the minimum value of 20 ms. In the example interface, this value is set to 60 ms. If the limit value specified in the EMG or IMU is not exceeded for 60ms (12 measurements, each taking 5 ms since the device operates at 200 Hz), the sign is considered finished. Following the end of the sign, the data properties are calculated and sent to the artificial intelligence algorithm for classification.

The IMU orientation data determines the position of the device in the x, y, and z planes and detects motion. The IMU Trackbar maximum value is set to 360, as this is the maximum value of the angles being measured. In the provided interface, the value is set to 20. If the absolute value of the mean of the changes in the Euler angle labels ( $\alpha$ ,  $\beta$ ,  $\gamma$ ) exceeds 20 degrees, it indicates movement. The Euler angles are obtained from the IMU sensor using the Quaternion method. To convert the given Euler angle labels to quadrilateral labels, use the following conversion. The values of roll, pitch, and yaw angles from the Euler angle notation are represented by  $\alpha$ ,  $\beta$ , and  $\gamma$  respectively. The assumed rotation order is from pitch to roll from deflection. The corresponding quarter q is defined as:

$$q = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} \cos \frac{\alpha}{2} \cos \frac{\beta}{2} \cos \frac{\gamma}{2} + \sin \frac{\alpha}{2} \sin \frac{\beta}{2} \sin \frac{\gamma}{2} \\ \sin \frac{\alpha}{2} \cos \frac{\beta}{2} \cos \frac{\gamma}{2} - \cos \frac{\alpha}{2} \sin \frac{\beta}{2} \sin \frac{\gamma}{2} \\ \cos \frac{\alpha}{2} \sin \frac{\beta}{2} \cos \frac{\gamma}{2} + \sin \frac{\alpha}{2} \cos \frac{\beta}{2} \sin \frac{\gamma}{2} \\ \cos \frac{\alpha}{2} \cos \frac{\beta}{2} \sin \frac{\gamma}{2} - \sin \frac{\alpha}{2} \sin \frac{\beta}{2} \cos \frac{\gamma}{2} \end{bmatrix} \quad (1)$$

The minimum duration of the movement is set with the minimum record time, which is 0.4 seconds in the example interface. To calculate certain features, a certain amount of data is required. An option has been added to prevent the software from giving errors.

Additionally, detailed information can be displayed or hidden for simplicity and performance improvement using checkboxes for Log, 3D Arm, Progress, Sound, and Graph.

The Log Checkbox allows for instant viewing of all data and software/device changes.

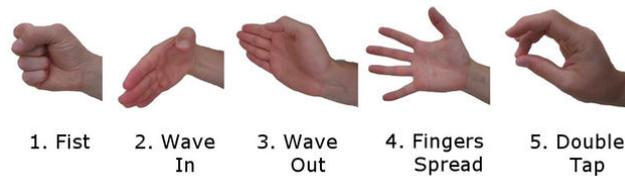
The 3D Arm Checkbox displays arm movements on the screen in real-time, with the option to hide the feature to reduce performance and battery consumption.

The Progress Checkbox displays recording and training times as a progress bar.

If the Sound Checkbox is selected, the signs are read aloud using the operating system's text-to-speech feature. This software feature enables the user's phone to convert sign language movements into sound.

The Graph Checkbox allows for visualization of device data in chart form.

Additionally, the MYO Armband's classification features can be used to control the application (e.g., putting the device to sleep, waking it up, starting and stopping). Figure 5 shows that the device classifies five different hand movements as hardware and sends them over the BLE service. The application can also utilize these features to generate a total of ten different commands from two different devices. For instance, when the user makes a 'Fist' movement with their right arm, the 3D arms move to the reference position. To display the logs, click on the icon of three lines in a row located in the lower right corner of the main graphical interface. The logs will appear when the 'Wave In' movement is made with the right arm and disappear when the same movement is made again.



**Figure 5.** The five gestures recognized by the Myo Armband [45].

When the 'Fingers Spread' movement is made with the right arm, the BLE characteristics of the devices are subscribed, meaning that the data begins to be received. In other words, the program starts. When the movement is repeated, the program stops. Many commands like this can be performed with wristbands without the need to touch the phone, even when it is in our pocket.

### 2.2.3. Records User Interface

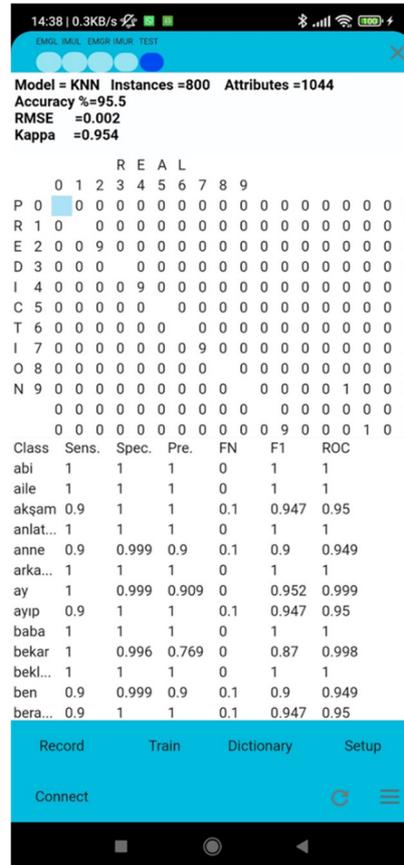
This interface was designed to collect data for training artificial intelligence. Figure 6 displays a list box on the left showing words and the number of records associated with each word. To create a new record, select the word and click the 'New Rec' button. Recording begins when the motion starts and continues until it ends. When the recording is complete, the word count increases by one. If the Auto Checkbox is selected, the list box will automatically switch to the next word. When the movement starts, the data for the next word is saved. If it reaches the last word in the list, it will move on to the first word and continue until the Stop Rec Button is clicked. This interface allows for graphical display of data from the device during recording. Additionally, if a gif file with the same name as the word exists, the movement video will be displayed. The list box at the top displays the record numbers of the selected word in the database. By selecting the record number, the data of that record is graphically displayed, allowing for the identification of any incorrect records. To remove unwanted records, double-click (double Tab on the phone) on the record number and select 'yes' in the warning message. The interface displays the graph of record number 244 for the word 'correct'.



**Figure 6.** An example records user graphical interface.

#### 2.2.4. Train User Interface

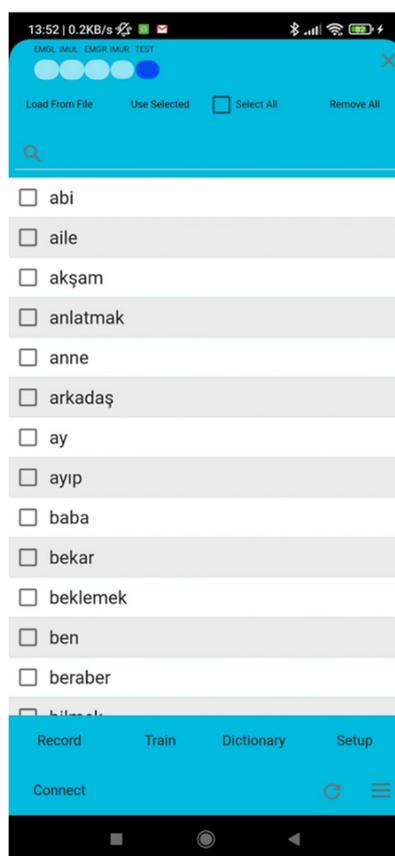
Training involves using the data obtained after extracting features from the IMU and EMG raw data. The algorithm and parameters selected in this interface are used to train existing data. The training is then tested using the 10-fold cross-validation method. The performance values of the end-of-training algorithm are displayed in Figure 7. The sample interface uses the K-nearest neighbor algorithm with a K=3 parameter.



**Figure 7.** An example train user graphical interface.

#### 2.2.5. Dictionary User Interface

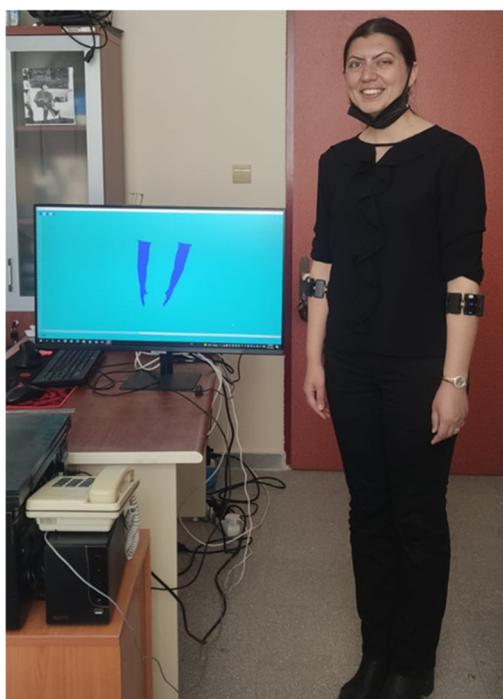
This interface allows for the addition and removal of words or sentences from the database (refer to Figure 8). Additionally, words can be uploaded from a txt file for convenience. However, please note that loading from the file will result in the deletion of all existing records. If accepted, a new dictionary will be added. Another noteworthy feature of the application is its ability to support multiple languages. As the application is personalized, it can create a unique experience by allowing users to choose their own words, create custom records, and train it using artificial intelligence.



**Figure 8.** An example dictionary user graphical interface.

### 2.3. Data Collecting

Determining the signs of Turkish Sign Language is a crucial aspect of this study, which will be tested by the system. To achieve this, we received support from Hülya AYKUTLU, a licensed sign language trainer and translator with years of experience in the Special Education Department. The testing location is shown in Figure 9.



**Figure 9.** Testing the system by a sign language instructor.

The system was designed to predict more than 80 words, but to test the performance of the system, 80 words, which is the maximum number of words used in similar studies, was chosen. While selecting these 80 words, requests from sign language instructors and users were evaluated.

To test the system, 80 frequently used words in sign language and daily speech were selected. The system also supports additional words.

If we need to categorize the selected words.

- Greeting and introduction words (hello, I'm glad, meet, etc.), Turkish (merhaba, sevindim, görüşürüz vb.);
- Family Words (mother, father, brother, etc.), Turkish (anne, baba, abi vb.);
- Pronouns and Person Signs (I, you, they etc.), Turkish (ben, sen, onlar vb.);
- Common Verbs (come, go, take, give, etc.), Turkish (gel, git, al, ver, vb.);
- Question Words (what, why), Turkish (ne, neden);
- Other Daily Life Words (home, name, good, warm, easy, married, year, etc.), Turkish (ev, isim, iyi, sıcak, kolay, evli, yıl vb.).

The 80-word dictionary was repeated 10 times, with each word being recorded by the IMU sensors of the Myo Armband device 50 times per second. The data collected includes 10 measurements, consisting of gyroscope accelerometer (x, y and z) and orientation (x, y, z and w). Additionally, data from eight sEMG sensors of the device are measured 200 times per second and stored in memory during recording. At the end of the recording, 1044 features are extracted from the stored data, including both raw and feature-extracted data, which are then stored in the database. If the sign recording lasted for 1 second, a total of 4200 raw data and 1044 feature data would be stored. The data was initially segmented using either a fixed time or a fixed reference arm position. The slow and tiring nature of the application testing was not well received by users. To ensure fast and effective communication during data segmentation, a motion detection system is employed instead of a fixed time. The sensitivity setting can be adjusted in the settings section to determine the most suitable option. Another important aspect of this feature is that hearing-impaired individuals may produce signs at varying speeds depending on their level of excitement and emotional state. As a result, the duration of the same sign may differ. To account for this, ten different recordings were taken for each received sign.

#### 2.4. Feature Extraction

In machine learning, it is crucial to use distinguishable features as they enhance the success and performance of the system. The accuracy and performance of the developed SLR system were improved by using feature extraction methods from the EMG data in Table 2 [46]. Some of the features listed in Table 2 were also utilized for the IMU data.

**Table 2.** Features extracted from sEMG signals.

Time domain	Frequency domain	Entropy-based
Mean value	Peak frequency	Shannon
Integrated EMG	Median frequency	Spectral
Higuchi Fractal Dimension	Modified Median Frequency	SVD
Petrosian Fractal Dimension	Modified Mean Frequency	Fisher
Detrended Fluctuation Anal.	Intensity weighted mean frequency	
Nonlinear energy	Intensity weighted bandwidth	
Slope	Total spectrum	
Line length	Mean power spectrum	
Willison Amplitude	Wavelet energy	
Standard deviation	AR coefficient 1	
Min value	AR modelling error 1	
Max value	AR coefficient 2	

Hurst exponent	AR modelling error 2
Minima	AR coefficient 4
Maxima	AR modelling error4
Skewness	AR coefficient 8
Kurtosis	AR modelling error 8
Zero crossings	
Zero crossings of 1 derivative	
Zero crossings of 2 derivative	
RMS amplitude	
Inactive samples	
Mobility	
Activity	
Complexity	

The Fourier transform is a technique for extracting frequency information from a signal. In this study, we collected discrete samples, so we used the Discrete Fourier Transform (DFT). Additionally, we obtained the wavelet energy property using the wavelet transform. According to information theory, entropy is a measure that describes the uncertainty in a variable. In this application, we extracted features based on entropy using four different methods. The methods used converted all IMU and EMG data from two devices into 1044 pieces of data, resulting in a high level of classification performance. However, training with raw data was not successful.

### 2.5. Classification

In this study, we used various classification methods to address research objectives. The Weka Deep Learning (WDL) algorithm was employed to harness the power of deep learning for extracting features and classifying data. The k-Nearest Neighbor (KNN) method, a non-parametric algorithm, is utilized for pattern recognition based on data point proximity in the feature space. We also employ the Multi-Layer Perceptron (MLP), a type of artificial neural network, for its ability to model complex relationships within the data through its layered structure. Naïve Bayes (NB), a probabilistic classifier, was chosen for its simplicity and efficiency in managing datasets. The Random Forest (RF) method, an ensemble learning technique, was applied to combine the predictions from numerous decision trees, improving the classification performance. Support Vector Machines (SVM), known for their effectiveness in high-dimensional spaces, were employed to determine the optimal hyperplane to separate data. Each of these classification methods has been selected to make use of its specific strengths and capabilities in tackling the complexities of the research problem, enabling a comparative study for a thorough analysis.

The study compared the classification performance of features obtained through feature extraction using various classification algorithms and parameters. All algorithms were tested using the 10-Fold cross-validation method. The Weka Application Programming Interface (API), developed specifically for the Windows platform, allows for the use of all algorithms available in Weka by converting the data in the database into ARFF file format [47]. Trainings are saved as a model file containing the algorithm name and parameters. Therefore, a previously performed training can be predicted using the model file without the need for retraining. On the Android platform, only the KNN algorithm is used due to its classification performance and fast operation [48]. The aim is to incorporate additional algorithms in the future, including the Weka API on the Android platform.

To add Weka algorithms to the program, edit the 'Data\weka\algorithms.txt' file located in the program's installation folder. Here is an example of the file's contents:

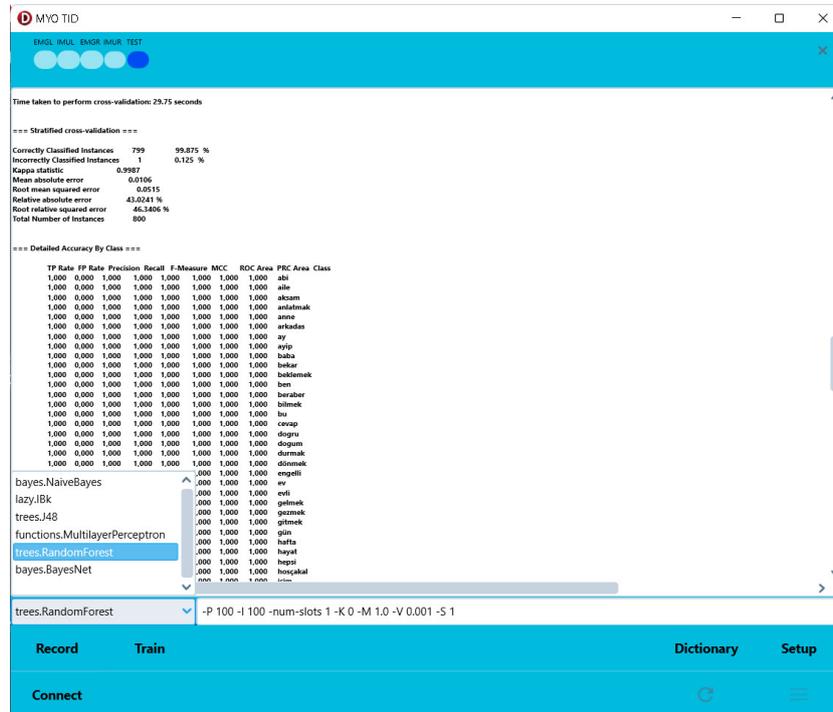
```

bayes.NaiveBayes
lazy.IBk -K 1 -W 0 -A
trees.J48 -C 0.25 -M 2
functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 50 -V 0 -S 0 -E 20 -H 50
trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

```

*bayes.BayesNet -D -Q*

When the program is executed, each added line is displayed as a combo box, as shown in Figure 10 when the training interface is accessed.



**Figure 10.** An example Weka training user graphical interface (only Windows).

In the measurements taken for a total of 800 data points, consisting of 80 signs and 10 repetitions, the average time taken for feature extraction from the raw data and classification after the signal ended was 21.2ms. This demonstrates the system's ability to perform in real-time, as the time was imperceptible to the users testing the system.

The training results of six different algorithms, selected based on their popularity and classification success, were compared. Table 3 presents the results of the training conducted using a total of 800 data points, with 80 signs and 10 records for each sign. The 10-fold cross-validation method was used for testing. This method uses all data for both testing and training. The default parameters of the algorithms in Weka were used for this comparison, as their performance was quite high. The default parameters of the algorithms in Weka were used for this comparison, as their performance was quite high. No alternative parameters were tested.

**Table 3.** 10-fold cross validation classification performances of different algorithms.

Algorithm	Accuracy (%)	Kappa statistic	Root mean squared error
Weka Deep Learning (WDL)	99.8750	0.9987	0.0053
k-Nearest Neighbor (KNN)	95.5000	0.9542	0.0020
Multi-Layer Perceptron (MLP)	98.0000	0.9797	0.0201
Naïve Bayes (NB)	87.6250	0.8747	0.0556
Random Forest (RF)	99.8750	0.9987	0.037
Support Vector Machines (SVM)	97.6250	0.9759	0.11

In another application, the training was conducted by splitting the data at different rates instead of using 10-fold cross-validation. Some of the randomly selected records from a total of 10 records for each sign were used for training, while the remaining records were used for testing. The results of these classifications, made using the same algorithm and parameters, are also shown in Table 4.

**Table 4.** Classification performances of different algorithms with splitted data.

Number of Records Used		Algorithm Accuracy (%)					
Training	Test	MLP	WDL	RF	KNN	SVN	NB
1	1	73.75	62.5	30	71.25	62.5	16.25
2	2	90.625	73.75	89.375	78.125	81.25	26.875
3	3	91.25	99.5833	99.1667	89.166	90	45.833
4	4	92.1875	99.375	98.75	91.25	90	61.875
5	5	95	99.75	99.25	92.25	94	67.75
6	4	95	91.5625	99.0625	95	95.312	75.312
9	1	97.5	96.25	100	97.5	98.75	85

The classification results obtained from different algorithms indicate that the Random Forest algorithm outperformed the Naïve Bayes algorithm. It is important to note that training using a single record resulted in very low success rates. Therefore, it is recommended to repeat each sign at least three times to create a record. Increasing the number of repetitions is directly proportional to the increase in performance. Despite the variations in recording speeds, the classification performance remains consistently high.

The variation in performance among different algorithms in a 10-fold cross-validation classification task can be attributed to several factors, such as the nature of the algorithms, their handling of data complexity, and their sensitivity to the specifics of the dataset used. In this section, we will evaluate the performance of the listed algorithms based on three key metrics: accuracy, kappa statistic, and root mean squared error (RMSE).

**WDL and RF Performance:** Both WDL and RF have demonstrated exceptional accuracy of 99.875%, with identical kappa statistics of 0.9987, indicating almost perfect classification capabilities compared to a random classifier. However, it is worth noting that WDL outperforms RF in terms of RMSE, with an impressively low value of 0.0053, compared to RF's RMSE of 0.037. The analysis shows that WDL is more consistent in its predictions across the dataset, possibly due to better handling of outlier data or noise within the dataset.

KNN performs moderately well, with an accuracy of 95.5% and a kappa statistic of 0.9542. It has the lowest RMSE among all algorithms at 0.0020, indicating very tight clustering around the true values. KNN is a strong model, despite its lower accuracy compared to WDL and RF. It is important to note that KNN may require parameter tuning, such as the choice of 'k', and may be sensitive to noisy data.

MLP exhibits a strong performance with an accuracy of 98% and a kappa statistic of 0.9797, despite its relatively higher RMSE of 0.0201. a kappa statistic of 0.9797, despite its relatively higher RMSE of 0.0201. The higher RMSE in comparison to its accuracy and kappa indicates variations in prediction errors, possibly due to the complexity of the model and the need for careful tuning of its layers and neurons.

**NB:** In contrast, NB demonstrates the lowest performance among all evaluated models, with an accuracy of 87.625%, a kappa statistic of 0.8747, and a relatively high RMSE of 0.0556. While NB may encounter difficulties when dealing with datasets where features are not independent, which is a core assumption of the algorithm, SVM is able to handle such datasets with ease.

Although SVM has the highest RMSE of 0.11 among the algorithms, its superior performance in other areas makes it the recommended choice. The analysis clearly demonstrates that the SVM algorithm outperforms the NB algorithm in terms of accuracy and kappa statistic. Although SVM has the highest RMSE of 0.11 among the algorithms, its superior performance in other areas makes it the recommended choice. Although SVM has the highest RMSE of 0.11 among the algorithms, its superior performance in other areas makes it the recommended choice. The high RMSE, despite good accuracy and kappa statistic, suggests that SVM's decision boundary may be less stable or more sensitive to individual data points, possibly due to the choice of kernel or regularization parameters.

WDL and RF outperform the other models in terms of accuracy and kappa statistics, likely due to their robustness to data imperfections and their ability to model complex patterns. WDL is superior in handling outliers or noise compared to RF, as evidenced by its lower RMSE. The other models' performance is dependent on their intrinsic assumptions and sensitivity to data characteristics. It is imperative to select the appropriate model based on the specific requirements and nature of the dataset.

### 3. Results and Discussion

In their review article, Kudrinko et al. examined all systematic literature review (SLR) studies to date and identified the essential features that should be present in SLR systems [49].

- Real-life usability: The sign language technology should be applicable in real-life situations outside of the laboratory.
- Accuracy and minimal delay: The technology must accurately convert sequences of movements into text and speech with minimal delay.
- Aesthetics and comfort: The design of the technology should prioritize aesthetics and comfort for the user.
- Wireless and rechargeable: To be useful, it must be wireless and easily rechargeable.
- User-centric design: When designing systems.

The existing literature focuses mainly on the recognition of sign language movements, but fails to meet these essential criteria, thereby limiting its practical applicability. To address this gap, our study presents a mobile application developed in collaboration with 10 individuals with disabilities who provided valuable feedback confirming its utility and effectiveness in real-life scenarios. In particular, our approach emphasizes personalized application development, recognizing the inherent variability of sEMG data between users. By tailoring the application to individual physiological characteristics, we mitigate the challenges posed by inter-user variability, thereby enhancing performance and usability. In addition, the inclusion of multiple users in the training sets in previous studies may be the reason why classification performance decreases as the number of signs increases, highlighting the importance of personalized approaches to account for individual differences and optimize system performance. Through these efforts, we aim not only to advance the theoretical understanding of sign language recognition, but also to provide practical solutions that address the diverse needs of people with hearing impairments, ultimately facilitating seamless communication and integration into society.

The KNN algorithm and its corresponding model were chosen for the mobile application due to their exceptional performance and accuracy. It is worth noting that this algorithm has a single effective parameter, namely the K parameter.

The system can be used not only for Turkish Sign Language but for all sign languages. Because the application has been developed specifically for the user. The user can add any number of words (regardless of language) to the dictionary. Since biological data such as EMG may vary from person to person, the application is for the use of a single person. Since different people's data are tried to be trained on the same model in different studies, success rates are low. The user creates his own model by creating his own training data by performing the desired movements. He can adjust the desired delay and sensitivity settings. Since the Text to Speech(TTS) feature in the application uses the TTS system in the operating system, voice-over can be made in all languages.. The application allows users to create their own dictionary, store data, and train and use it with the existing artificial intelligence algorithm. In addition, the application allows users to control the device using sign language, including putting it to sleep, waking it up, and starting and stopping it. All signals are easily recognizable when using the Myo wristbands. The application is now being actively used by disabled people who have taken part in the trials.

In the future, we will redesign the software to use the device we have developed. Although we encountered a challenge during the study with the discontinuation of the Myo Armband, we overcame this obstacle by embarking on a project to develop our own device. Our goal is to create a novel system that comprehensively addresses the needs of people with disabilities. We envision this

system not only as a replacement for the Myo Armband, but also as an innovative solution that surpasses its capabilities. Through this initiative, we aim to democratize access to assistive technology and ensure that people with disabilities around the world can benefit from our system. Our goal is to make this device readily available and accessible to anyone in need, thereby promoting inclusivity and empowerment within the disability community.

**Author Contributions:** Conceptualization, Ü.K.; Data curation, Ü.K.; Formal analysis, Ü.K.; Funding acquisition, Ü.K.; Investigation, Ü.K.; Methodology, İ.U. and Ü.K.; Project administration, İ.U.; Resources, Ü.K.; Software, İ.U.; Supervision, İ.U.; Validation, İ.U.; Visualization, İ.U.; Writing – original draft, İ.U.; Writing – review & editing, Ü.K.

**Funding:** No funding was received for conducting this study.

**Data Availability Statement:** Access to this data is restricted, but interested researchers may contact Assoc. Prof. İlhan UMUT (iumut@nku.edu.tr) for inquiries regarding data access.

**Acknowledgments:** The authors acknowledge the support given by Hülya AYKUTLU in providing them with the knowledge.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. W. H. Organization. "Deafness and hearing loss." <https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss> (accessed 06.04.2023).
2. G. E. Crealey and C. O'Neill, "Hearing loss, mental well-being and healthcare use: results from the Health Survey for England (HSE)," *Journal of Public Health*, vol. 42, no. 1, pp. 77-89, 2018, doi: 10.1093/pubmed/fdy209.
3. M. J. Cheok, Z. Omar, and M. H. Jaward, "A review of hand gesture and sign language recognition techniques," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 1, pp. 131-153, 2019/01/01 2019, doi: 10.1007/s13042-017-0705-5.
4. H.-D. Yang, "Sign Language Recognition with the Kinect Sensor Based on Conditional Random Fields," *Sensors*, vol. 15, no. 1, pp. 135-147, 2015. [Online]. Available: <https://www.mdpi.com/1424-8220/15/1/135>.
5. Z. Zafrulla, H. Sahni, A. Bedri, P. Thukral, and T. Starner, "Hand detection in American Sign Language depth data using domain-driven random forest regression," in 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), 4-8 May 2015, 2015, vol. 1, pp. 1-7, doi: 10.1109/FG.2015.7163135.
6. S. B. Abdullahi and K. Chamnongthai, "American Sign Language Words Recognition of Skeletal Videos Using Processed Video Driven Multi-Stacked Deep LSTM," *Sensors*, vol. 22, no. 4, p. 1406, 2022. [Online]. Available: <https://www.mdpi.com/1424-8220/22/4/1406>.
7. S. B. Abdullahi and K. Chamnongthai, "American Sign Language Words Recognition Using Spatio-Temporal Prosodic and Angle Features: A Sequential Learning Approach," *IEEE Access*, vol. 10, pp. 15911-15923, 2022, doi: 10.1109/ACCESS.2022.3148132.
8. S. B. Abdullahi and K. Chamnongthai, "IDF-Sign: Addressing Inconsistent Depth Features for Dynamic Sign Word Recognition," *IEEE Access*, vol. 11, pp. 88511-88526, 2023, doi: 10.1109/ACCESS.2023.3305255.
9. S. B. Abdullahi, K. Chamnongthai, V. Bolon-Canedo, and B. Cancela, "Spatial-temporal feature-based End-to-end Fourier network for 3D sign language recognition," *Expert Systems with Applications*, vol. 248, p. 123258, 2024/08/15/ 2024, doi: <https://doi.org/10.1016/j.eswa.2024.123258>.
10. Z. Katılmış and C. Karakuzu, "ELM based two-handed dynamic Turkish Sign Language (TSL) word recognition," *Expert Systems with Applications*, vol. 182, p. 115213, 2021/11/15/ 2021, doi: <https://doi.org/10.1016/j.eswa.2021.115213>.
11. C. H. Chuan, E. Regina, and C. Guardino, "American Sign Language Recognition Using Leap Motion Sensor," in 2014 13th International Conference on Machine Learning and Applications, 3-6 Dec. 2014, 2014, pp. 541-544, doi: 10.1109/ICMLA.2014.110.
12. H. Wang, M. Leu, and C. Oz, "American Sign Language Recognition Using Multidimensional Hidden Markov Models," *Journal of Information Science and Engineering - JISE*, vol. 22, pp. 1109-1123, 09/01 2006.
13. N. Tubaiz, T. Shanableh, and K. Assaleh, "Glove-Based Continuous Arabic Sign Language Recognition in User-Dependent Mode," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 4, pp. 526-533, 2015, doi: 10.1109/THMS.2015.2406692.
14. A. Garg, "Converting American sign language to voice using RBFNN," *Sciences*, 2012.

15. T. Starner, J. Weaver, and A. Pentland, "Real-time american sign language recognition using desk and wearable computer-based video," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 20, no. 12, pp. 1371-1375, 1998.
16. S. D. Thepade, G. Kulkarni, A. Narkhede, P. Kelvekar, and S. Tathe, "Sign language recognition using color means of gradient slope magnitude edge images," in *2013 International Conference on Intelligent Systems and Signal Processing (ISSP)*, 2013: IEEE, pp. 216-220.
17. T. Kim, "American Sign Language fingerspelling recognition from video: Methods for unrestricted recognition and signer-independence," *arXiv preprint arXiv:1608.08339*, 2016.
18. K. Sadeddine, F. Z. Chelali, and R. Djeradi, "Sign language recognition using PCA, wavelet and neural network," in *2015 3rd International Conference on Control, Engineering & Information Technology (CEIT)*, 25-27 May 2015, 2015, pp. 1-6, doi: 10.1109/CEIT.2015.7233117.
19. R. Yang, S. Sarkar, and B. Loeding, "Handling Movement Epenthesis and Hand Segmentation Ambiguities in Continuous Sign Language Recognition Using Nested Dynamic Programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 3, pp. 462-477, 2010, doi: 10.1109/TPAMI.2009.26.
20. Y. C. Bilge, R. G. Cinbis, and N. Ikizler-Cinbis, "Towards Zero-Shot Sign Language Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 1217-1232, 2023, doi: 10.1109/TPAMI.2022.3143074.
21. Z. Liu, L. Pang, and X. Qi, "MEN: Mutual Enhancement Networks for Sign Language Recognition and Education," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1-15, 2022, doi: 10.1109/TNNLS.2022.3174031.
22. M. Mohandes, M. Deriche, and J. Liu, "Image-Based and Sensor-Based Approaches to Arabic Sign Language Recognition," *IEEE Transactions on Human-Machine Systems*, vol. 44, no. 4, pp. 551-557, 2014, doi: 10.1109/THMS.2014.2318280.
23. X. Han, F. Lu, J. Yin, G. Tian, and J. Liu, "Sign Language Recognition Based on R (2+1) D With Spatial-Temporal-Channel Attention," *IEEE Transactions on Human-Machine Systems*, vol. 52, no. 4, pp. 687-698, 2022, doi: 10.1109/THMS.2022.3144000.
24. E. Rajalakshmi et al., "Multi-Semantic Discriminative Feature Learning for Sign Gesture Recognition Using Hybrid Deep Neural Architecture," *IEEE Access*, vol. 11, pp. 2226-2238, 2023, doi: 10.1109/ACCESS.2022.3233671.
25. E. Rajalakshmi et al., "Static and Dynamic Isolated Indian and Russian Sign Language Recognition with Spatial and Temporal Feature Detection Using Hybrid Neural Network," *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. 22, no. 1, p. Article 26, 2022, doi: 10.1145/3530989.
26. P. Paudyal, A. Banerjee, and S. K. S. Gupta, "SCEPTRE: A Pervasive, Non-Invasive, and Programmable Gesture Recognition Technology," presented at the *Proceedings of the 21st International Conference on Intelligent User Interfaces*, Sonoma, California, USA, 2016. [Online]. Available: <https://doi.org/10.1145/2856767.2856794>.
27. R. Fatmi, S. Rashad, R. Integlia, and G. Hutchison, "American Sign Language Recognition using Hidden Markov Models and Wearable Motion Sensors," 03/18 2017.
28. C. Savur and F. Sahin, "American Sign Language Recognition system by using surface EMG signal," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 9-12 Oct. 2016, 2016, pp. 002872-002877, doi: 10.1109/SMC.2016.7844675.
29. M. Seddiqi, H. Kivrak, and H. Kose, "Recognition of Turkish Sign Language (TID) Using sEMG Sensor," in *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 15-17 Oct. 2020 2020, pp. 1-6, doi: 10.1109/ASYU50717.2020.9259859.
30. Z. Zhang, K. Yang, J. Qian, and L. Zhang, "Real-Time Surface EMG Pattern Recognition for Hand Gestures Based on an Artificial Neural Network," (in eng), *Sensors (Basel)*, vol. 19, no. 14, Jul 18, 2019, doi: 10.3390/s19143170.
31. Z. Zhang, Z. Su, and G. Yang, "Real-Time Chinese Sign Language Recognition Based on Artificial Neural Networks," in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 6-8 Dec. 2019, pp. 1413-1417, doi: 10.1109/ROBIO49542.2019.8961641.
32. S. P. Y. Jane and S. Sasidhar, "Sign Language Interpreter: Classification of Forearm EMG and IMU Signals for Signing Exact English," in *2018 IEEE 14th International Conference on Control and Automation (ICCA)*, 12-15 June 2018, 2018, pp. 947-952, doi: 10.1109/ICCA.2018.8444266.
33. X. Yang, X. Chen, X. Cao, S. Wei, and X. Zhang, "Chinese Sign Language Recognition Based on an Optimized Tree-Structure Framework," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 4, pp. 994-1004, 2017, doi: 10.1109/JBHI.2016.2560907.
34. B. G. Lee and S. M. Lee, "Smart Wearable Hand Device for Sign Language Interpretation System with Sensors Fusion," *IEEE Sensors Journal*, vol. 18, no. 3, pp. 1224-1232, 2018, doi: 10.1109/JSEN.2017.2779466.
35. R. Fatmi, S. Rashad, and R. Integlia, "Comparing ANN, SVM, and HMM based Machine Learning Methods for American Sign Language Recognition using Wearable Motion Sensors," in *2019 IEEE 9th Annual*

- Computing and Communication Workshop and Conference (CCWC), 7-9 Jan. 2019, 2019, pp. 0290-0297, doi: 10.1109/CCWC.2019.8666491.
36. Z. Zheng, Q. Wang, D. Yang, Q. Wang, W. Huang, and Y. Xu, "L-Sign: Large-Vocabulary Sign Gestures Recognition System," *IEEE Transactions on Human-Machine Systems*, vol. 52, no. 2, pp. 290-301, 2022, doi: 10.1109/THMS.2022.3146787.
  37. A. L. P. Madushanka, R. G. D. C. Senevirathne, L. M. H. Wijesekara, S. M. K. D. Arunatilake, and K. D. Sandaruwan, "Framework for Sinhala Sign Language recognition and translation using a wearable armband," in *2016 Sixteenth International Conference on Advances in ICT for Emerging Regions (ICTer)*, 1-3 Sept. 2016, 2016, pp. 49-57, doi: 10.1109/ICTER.2016.7829898.
  38. S. S. Fels and G. E. Hinton, "Glove-Talk: a neural network interface between a data-glove and a speech synthesizer," *IEEE Transactions on Neural Networks*, vol. 4, no. 1, pp. 2-8, 1993, doi: 10.1109/72.182690.
  39. S. Jiang et al., "Feasibility of Wrist-Worn, Real-Time Hand, and Surface Gesture Recognition via sEMG and IMU Sensing," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3376-3385, 2018, doi: 10.1109/TII.2017.2779814.
  40. X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, and J. Yang, "A Framework for Hand Gesture Recognition Based on Accelerometer and EMG Sensors," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 41, no. 6, pp. 1064-1076, 2011, doi: 10.1109/TSMCA.2011.2116004.
  41. J. Wu, L. Sun, and R. Jafari, "A Wearable System for Recognizing American Sign Language in Real-Time Using IMU and Surface EMG Sensors," *IEEE Journal of Biomedical and Health Informatics*, Article vol. 20, no. 5, pp. 1281-1290, 2016, Art no. 7552525, doi: 10.1109/JBHI.2016.2598302.
  42. C. Savur and F. Sahin, "Real-Time American Sign Language Recognition System Using Surface EMG Signal," in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, 9-11 Dec. 2015, 2015, pp. 497-502, doi: 10.1109/ICMLA.2015.212.
  43. M. Atzori et al., "Characterization of a Benchmark Database for Myoelectric Movement Classification," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 1, pp. 73-83, 2015, doi: 10.1109/TNSRE.2014.2328495.
  44. P. Visconti, F. Gaetani, G. A. Zappatore, and P. Primiceri, "Technical Features and Functionalities of Myo Armband: An Overview on Related Literature and Advanced Applications of Myoelectric Armbands Mainly Focused on Arm Prostheses," *International Journal on Smart Sensing and Intelligent Systems*, vol. 11, no. 1, pp. 1-25, 2018, doi: doi:10.21307/ijssis-2018-005.
  45. M. Cognolato et al., *Hand Gesture Classification in Transradial Amputees Using the Myo Armband Classifier\** This work was partially supported by the Swiss National Science Foundation Sinergia project # 410160837 MeganePro. 2018, pp. 156-161.
  46. İ. Umut, "PSGMiner: A modular software for polysomnographic analysis," *Computers in Biology and Medicine*, vol. 73, pp. 1-9, 2016/06/01/ 2016, doi: <https://doi.org/10.1016/j.compbimed.2016.03.023>.
  47. E. F. M. A. H. I. H. Witten, *The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques, Fourth Edition ed. Morgan Kaufmann, 2016.*
  48. T. L. Daniel and D. L. Chantal, "k Nearest Neighbor Algorithm," in *Discovering Knowledge in Data: An Introduction to Data Mining: Wiley, 2014*, pp. 149-164.
  49. K. Kudrinko, E. Flavin, X. Zhu, and Q. Li, "Wearable Sensor-Based Sign Language Recognition: A Comprehensive Review," *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 82-97, 2021, doi: 10.1109/RBME.2020.3019769.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.