

Intelligent Sign Multi-Language Real-Time Prediction System with Effective Data Preprocessing

Doaa E. Elmatary, Doaa M. Maher, Areeg Tarek Ibrahim

Department of Communications and Computers, AlSafwa High Institute of Engineering, Qalyubia, Egypt
Email: doaa.elmatary@alsafwa.edu.eg, eng.doaa.m.maher@gmail.com, areegtarek252@gmail.com

How to cite this paper: Elmatary, D.E., Maher, D.M. and Ibrahim, A.T. (2023) Intelligent Sign Multi-Language Real-Time Prediction System with Effective Data Preprocessing. *Journal of Computer and Communications*, 11, 120-134.
<https://doi.org/10.4236/jcc.2023.1110008>

Received: September 10, 2023

Accepted: October 24, 2023

Published: October 27, 2023

Copyright © 2023 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

A multidisciplinary approach for developing an intelligent sign multi-language recognition system to greatly enhance deaf-mute communication will be discussed and implemented. This involves designing a low-cost glove-based sensing system, collecting large and diverse datasets, preprocessing the data, and using efficient machine learning models. Furthermore, the glove is integrated with a user-friendly mobile application called “Lifesign” for this system. The main goal of this work is to minimize the processing time of machine learning classifiers while maintaining higher accuracy performance. This is achieved by using effective preprocessing algorithms to handle noisy and inconsistent data. Testing and iterating approaches have been applied to various classifiers to refine and improve their accuracy in the recognition process. Additionally, the Extra Trees (ET) classifier has been identified as the best algorithm, with results proving successful gesture prediction at an average accuracy of about 99.54%. A smart optimization feature has been implemented to control the size of data transferred via Bluetooth, allowing for fast recognition of consecutive gestures. Real-time performance has been measured through extensive experimental testing on various consecutive gestures, specifically referring to Arabic Sign Language (ArSL). The results have demonstrated that the system guarantees consecutive gesture recognition with a lower delay of 50 milliseconds.

Keywords

Hand Gesture Translator, Sign Multi-Language, Machine Learning Models, Deaf-Mute Community

1. Introduction

The World Health Organization (WHO) has confirmed that the Deaf-Mute

community represents over 5% of the world's population, with approximately 80% of them residing in low- and middle-income countries [1]. Furthermore, statistics from the United Nations reveal that in Egypt, the number of deaf and mute individuals constitute around 10% of the country's population, representing a significant segment of Egyptian society.

Sign language recognition poses a challenging task due to the complexity and variability of hand gestures. Consequently, numerous research and development centers have focused on this subject [2] [3]. Existing techniques for sign language translation can be broadly categorized as either sensor-based [4] [5] or vision-based systems [6] [7]. Sensor-based systems rely on sensor input to recognize gestures, while vision-based systems process images or videos using computers to interpret gestures. Although the latter approach is considered realistic, the complexity and sensitivity to lighting conditions, background clutter, and camera positioning make glove-based systems more suitable [8] [9].

One important aspect of this work is the focus on designing a low-cost and accessible glove-based sensing system that is relatively affordable and comfortable. Additionally, a large and diverse dataset is collected to train the machine learning models, and effective preprocessing algorithms are used to handle noisy and inconsistent data, ensuring the accuracy and reliability of the models in recognizing sign language gestures. Another crucial aspect is the integration of the glove system with a user-friendly mobile application called "LifeSign", providing a practical and accessible solution for easy communication.

Testing and iterating approaches were applied to various classifiers, including SVM, XGB, KNN, and ET, to refine and improve their accuracy in sign language recognition. The classifiers were trained on a large dataset of over 25,000 records for 30 different signs from Ar SL. These records were collected using an embedded sensing system within the glove, and a diverse pool of users contributed to the dataset, enhancing the accuracy and generalizability of the models. Detailed experimental evaluations were conducted to identify the best-performing algorithm. This involved comparing the performance of each classifier on a validation set and selecting the one with the highest accuracy. Once the best algorithm was determined, its weighted parameters were converted into a format suitable for integration into the LifeSign application. Through the smart features of the application, the system is capable of predicting performed gestures with high accuracy and minimal delay.

Furthermore, this work has the potential to be developed into a brand-new commercial product for translating sign language, incorporating several competing design features. The user-friendly nature, affordability, efficiency, and offline functionality of the system give it a significant advantage, appealing to a wide range of users and organizations. Moreover, the system enables bidirectional communication, having a positive impact on the lives of people with hearing impairments by improving their ability to communicate and fully participate in society. Particularly in educational settings for deaf or hard-of-hearing students, it provides them with an excellent opportunity to lead a normal life-

style and enhance their future opportunities.

Finally, the novelty of this system lies in its holistic approach, combining affordability, a robust dataset with effective preprocessing, user-friendly integration, and commercial potential. These unique features position it as a promising advancement in the field of sign language recognition and communication technology.

The remaining sections of this work are divided into four parts. Section 2 presents the technical aspects with a detailed explanation of the methodology, including the hardware system design, the proposed AI recognition model, and the mobile application. Section 3 showcases the experimental results of this work, comparing them with relevant and recent research. Finally, in Section 4, the conclusion of this work is provided, highlighting its significance.

2. Methodology

The proposed sign language recognition system consists of three main parts: a glove-based sensing system, an AI recognition model, and a simple mobile application “LifeSign” as shown in **Figure 1**. Detailed descriptions of these parts will be explained throughout the following sections.

2.1. Glove-Based Sensing System

The proposed system is designed to address the challenges faced by the deaf-mute community and provide a practical solution for communication using sign language. It captures data from the user’s hand movements through a combination of flex and gyroscope sensors. **Figure 2** illustrates the hardware system, which consists of two main parts that are connected and can be easily separated to improve user comfort. The first part is the glove, where each finger incorporates a flex sensor. The flex sensors are arranged in a voltage divider configuration [10]. Two 2-inch (5.5 cm) sensors are used for the thumb and pinky fingers, while three 4.5-inch (11.5 cm) sensors are used for the remaining fingers. The resistance change in the flex sensors is converted into a voltage change that

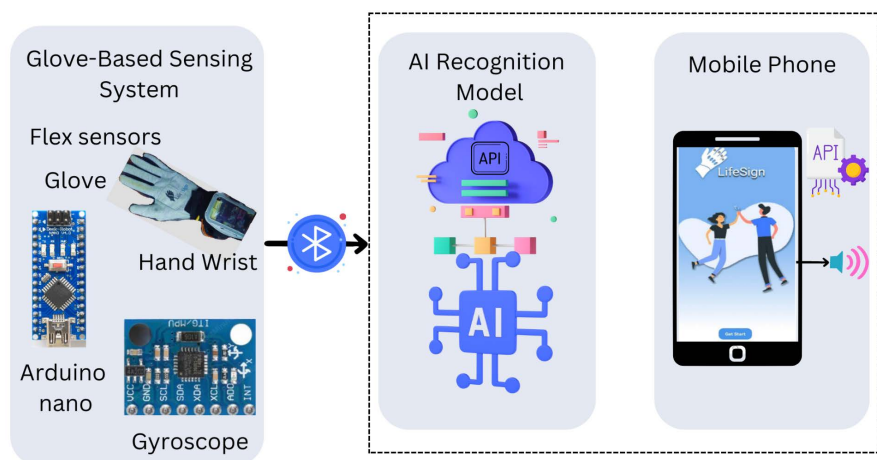


Figure 1. Main parts of the proposed system.

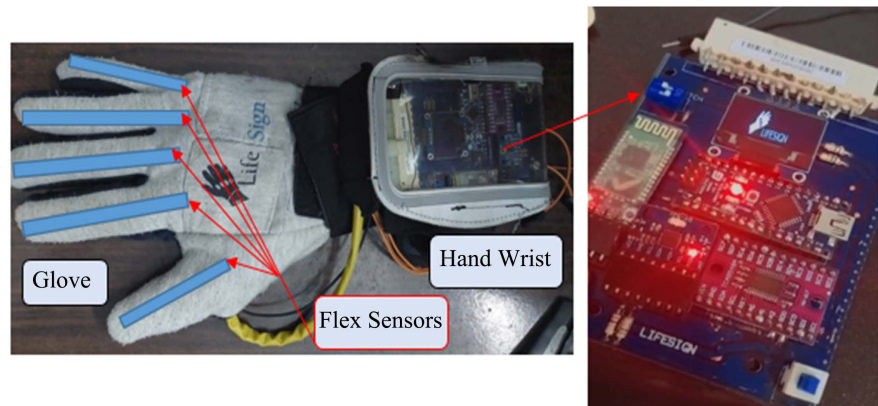


Figure 2. The proposed glove system with its main PCB.

correlates to the sensor bending. The flex sensor outputs are then converted into digital levels by an analog-to-digital converter (ADC) and displayed on the Arduino's serial monitor [11]. The second part of the system is the hand-wrist component, which houses the main printed circuit board (PCB) with all the electronic components. This part includes the Arduino Nano, IMU MPU-6050 module, HC-05 Bluetooth module, OLED display, I2C multiplexer, and two UR18650ZTA lithium-ion battery cells, each providing 3.7 V @ 3000 mAh. These battery cells ensure stable and reliable continuous power for at least 12 hours.

The microcontroller unit, Arduino Nano, is a compact and cost-effective board capable of processing data from flex sensors and the IMU MPU-6050 module. It is based on the ATmega328P microcontroller and offers a range of features and capabilities that make it versatile and flexible [11]. Its small size and low power consumption make it suitable for wearable and battery-powered applications.

The IMU MPU-6050 module is a sensor that combines a 3-axis gyroscope, a 3-axis accelerometer with MEMS technology, and a digital motion processor (DMP) in a single chip. It measures the hand's orientation in space using Euler angles to enhance the accuracy of the sign language recognition system.

The HC-05 Bluetooth module is used to enable the transmission of data from the flex sensors and MPU-6050 sensors to the mobile application for recognition [12].

The OLED Display (128 * 64) is a small monochrome 0.96" display that shows the LifeSign logo during startup. Additionally, the display indicates the battery percentage, which is divided into five levels, providing users with clear and actionable feedback about the battery status.

The TCA9548A I2C multiplexer is utilized to enable independent communication between the gyroscope and OLED devices and the microcontroller over the I2C communication protocol without overlapping addresses [13].

Finally, the glove system is responsible for collecting the data that will be used to train the AI recognition model that will be discussed below.

2.2. The Proposed AI Recognition Model

The proposed recognition system's workflow includes the preprocessing and preparation of the collected data, Cross-Validation, Classification, and Performance Evaluation [14] [15]. Arabic sign language can be recognized using data that has been collected. The experimental study conducted throughout the work will provide insights into the effectiveness of the recognition system. This will be discussed in the following sections.

2.2.1. Sensory Data Acquisition

The hand gesture obtained data during the actual readings via the flex sensors, gyroscope, and accelerometer will be classified and tabulated. The feature variables (input) for classifying hand gestures are the signals from flex sensors as five fingers (Thumb, Fore, Middle, Ring, and Pinky), the gyroscope axes (X, Y, Z), and the accelerometer axes (x_acc, y_acc, z_acc). However, the target variable (output), in this case, is the meaning of the hand gesture (word). Moreover, the datasets used in this study were performed by ten users and consisted of 30 different gestures "classes" with over 25,000 instances for each, and 11 feature variables in each instance. This is a relatively large and complex dataset, which can provide a robust basis for training and testing machine learning models. Once the sensory readings are collected from the serial monitor of Arduino IDE as a text, it is converted into structured data and saved in CSV files.

2.2.2. Data Preprocessing

1) Data Cleaning

According to the hand gesture fabrication, the obtained raw data during the collection process is unfortunately noisy, inconsistent, and incomplete. Data cleaning; by removing any outliers, noise, duplicates, and abnormalities that may negatively impact the efficiency of the recognition model is therefore important for higher-quality results. Missing values are also handled carefully by either dropping or imputing them. **Table 1** shows the dataset records after removing unnecessary, or irrelevant values during the data cleaning process.

2) Correlational Analysis

In addition to data cleaning, correlational analysis is performed by generating a correlation matrix to identify linear relationships between dataset attributes [16] [17]. The plotted correlation matrix for the dataset is illustrated in **Figure 3** using various colors to depict the strength of the correlation between the features. Darker colors indicate a stronger correlation, while lighter ones represent a weaker correlation [18] [19]. The (0.001 to +1.0) correlation range with values closer to zero is strongly confirming the correlation. Positive or negative values indicate a positive or correlation between attributes respectively. A stronger correlation, in the case of the Thumb attribute due to the data cleaning is depicted as shown in the tabulated data of **Figure 3(b)**.

In the context of data cleaning, Andrews plots have been used to evaluate the effectiveness of data cleaning and produce a clear and easily readable presentation of the data [20]. **Figure 4(a)** shows that the curves in the plot before the cleaning

process are randomly scattered and there is no apparent pattern or structure. This indicates that the data is too noisy or there are not enough relevant features to train a classifier. Contrary, **Figure 4(b)** shows a significant improvement in the structure and quality of the data which reflects the success of the cleaning process in removing any outliers or irrelevant data points. As a result, the cleaned data is now suitable for further analysis with no need for further data cleaning and feature engineering.

Table 1. The dataset records during the data cleaning process.

Applied process	Original dataset	After removing nulls	After removing duplicates
No. of dataset	1,029,801	1,029,659	680,198

	Thumb	Fore	Middle	Ring	Pinky	X	Y	Z	x_acc	y_acc	z_acc
Thumb	1.000000	0.031837	0.030505	0.034306	0.055000	0.003003	0.001177	0.002241	0.073216	-0.049237	-0.016502
Fore	0.031837	1.000000	0.463619	0.334233	0.129586	0.000556	-0.001103	0.005521	-0.057652	-0.173818	0.197106
Middle	0.030505	0.463619	1.000000	0.753060	0.397298	0.010547	-0.006137	0.004169	-0.051495	-0.138073	0.128546
Ring	0.034306	0.334233	0.753060	1.000000	0.567083	0.012924	-0.004929	0.007085	-0.072064	-0.183624	0.135378
Pinky	0.055000	0.129586	0.397298	0.567083	1.000000	0.003897	-0.003937	-0.000052	-0.125437	-0.055357	0.126048
X	0.003003	0.000556	0.010547	0.012924	0.003897	1.000000	-0.024347	0.231218	0.006664	-0.004749	-0.032054
Y	0.001177	-0.001103	-0.006137	-0.004929	-0.003937	-0.024347	1.000000	0.020935	0.005516	0.006363	-0.014599
Z	0.002241	0.005521	0.004169	0.007085	-0.000052	0.231218	0.020935	1.000000	0.014891	0.074653	0.005096
x_acc	0.073216	-0.057652	-0.051495	-0.072064	-0.125437	0.006664	0.005516	0.014891	1.000000	-0.264941	-0.643580
y_acc	-0.049237	-0.173818	-0.138073	-0.183624	-0.055357	-0.004749	0.006363	0.074653	-0.264941	1.000000	-0.123310
z_acc	-0.016502	0.197106	0.128546	0.135378	0.126048	-0.032054	-0.014599	0.005096	-0.643580	-0.123310	1.000000

(a)

	Thumb	Fore	Middle	Ring	Pinky	X	Y	Z	x_acc	y_acc	z_acc
Thumb	1.000000	0.114612	0.120738	0.150589	0.237021	-0.003495	0.000152	0.000173	0.344896	-0.202380	-0.097279
Fore	0.114612	1.000000	0.464011	0.334691	0.128629	0.000541	-0.001107	0.005445	-0.057417	-0.174001	0.196942
Middle	0.120738	0.464011	1.000000	0.753002	0.395956	0.010963	-0.006048	0.004279	-0.052357	-0.137918	0.128673
Ring	0.150589	0.334691	0.753002	1.000000	0.566588	0.013272	-0.004852	0.007142	-0.072762	-0.183486	0.135659
Pinky	0.237021	0.128629	0.395956	0.566588	1.000000	0.004340	-0.003877	0.000061	-0.126446	-0.054839	0.126308
X	-0.003495	0.000541	0.010963	0.013272	0.004340	1.000000	-0.024281	0.229960	0.006873	-0.004826	-0.031151
Y	0.000152	-0.001107	-0.006048	-0.004852	-0.003877	-0.024281	1.000000	0.020702	0.005532	0.006017	-0.014298
Z	0.000173	0.005445	0.004279	0.007142	0.000061	0.229960	0.020702	1.000000	0.014867	0.073853	0.005996
x_acc	0.344896	-0.057417	-0.052357	-0.072762	-0.126446	0.006873	0.005532	0.014867	1.000000	-0.265699	-0.643669
y_acc	-0.202380	-0.174001	-0.137918	-0.183486	-0.054839	-0.004826	0.006017	0.073853	-0.265699	1.000000	-0.122035
z_acc	-0.097279	0.196942	0.128673	0.135659	0.126308	-0.031151	-0.014298	0.005996	-0.643669	-0.122035	1.000000

(b)

Figure 3. Pair-wise correlations of the data frame (a) before data cleaning and (b) after data cleaning.

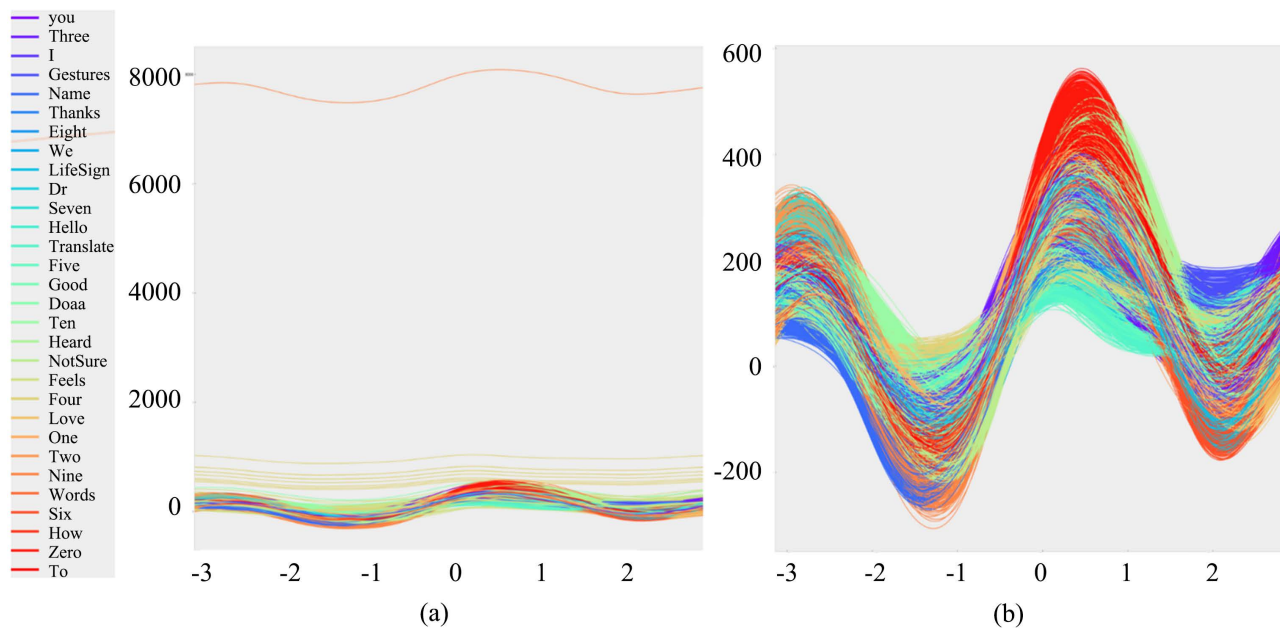


Figure 4. Andrews plot (a) before the cleaning process and (b) after the cleaning process.

2.2.3. Data Preparation

The data preparation includes data scaling, splitting, and cross-validation. The data is first normalized to prevent bias in the classification process and to ensure that features are on the same scale [20]. A log transformation has been used to transform skewed and Kurtosis features into a more normal distribution. The normalized data was then divided into two groups: 80% for training and 20% for testing. This split was both random and representative of the overall dataset to avoid overfitting. In addition, K-fold Cross-validation with $k = 10$ was employed to further evaluate the performance of the model [21]. This means that the data was divided into 10 equal parts, and the model was trained and tested 10 times, with each fold being used as the test set exactly once. This can reduce the variance in the evaluation metrics across different splits of the data. The results from each fold were then averaged to provide a more reliable estimate of the model's performance.

2.2.4. Classification Algorithms

Four supervised classification algorithms are presented in this study (SVM, XGB, KNN, and ET), with detailed explanations of each algorithm's operation on the datasets [22]. These four algorithms were selected due to their better suitability for the gathered dataset and their ability to achieve high-performance metrics like accuracy, precision, recall, and F-score. These metrics can provide insights into the effectiveness of each algorithm in correctly classifying different sign language gestures and identifying true positives and false positives.

2.2.5. Performance Evaluation

The confusion matrix evaluation and the estimated parameters below could be used to estimate the performance of the classifier.

Accuracy: the ability of the classifier to accurately predict how each instance will be labeled according to its class [23]. Equation (1) can be used to compute it.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (1)$$

where TP and TN represent true positive and true negative respectively and FP and FN demonstrate false positive and false negative.

Precision: is the relationship between the total number of positive prediction class values and the number of positive predictions [23]. It can gauge how accurate the classifier is, as “Equation (2)” demonstrates.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2)$$

Recall: is the ratio of the number of positive predictions to the number of positive class values in test data. It can be calculated using “Equation (3)” and refers to the completeness of the classifiers [23].

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3)$$

F-Measure: shows the harmonic mean of recall and precision, which demonstrates their balance [23]. It is computed using “Equation (4)”, as it is referred to

$$\text{F-Measure} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \quad (4)$$

Each algorithm is tested using the “testing set” after having been trained using a portion of the data known as the “training set”, which is withheld as “invisible data” from the model evaluation.

Table 2 presents the performance metrics of all classification algorithms using the dataset obtained from the suggested glove system. The performance of the algorithms varied based on their parameters and operation principles. Based on the results provided, it appears that KNN and Extra Trees (ET) classifiers achieved the most promising results among the classifiers evaluated, while SVM and XGB performed relatively poorly and had longer testing times. Moreover, ET has a faster training and testing time than KNN. This is because KNN does not have a traditional training period, as the training data is simply stored in memory. However, the classification time can be slow for large datasets or when the number of stored training data points is large. In contrast, Extra Trees has a training period where the model is constructed by building multiple decision trees. The training time for Extra Trees depends on the number of trees to be built, the number of features, and the size of the training data.

For testing, KNN requires calculating the distance between each test point and all the training points, which can result in slow testing times and be computationally expensive. In contrast, Extra Trees involves traversing a decision tree to classify each test point, resulting in faster testing times. However, based on the typical performance characteristics of KNN and Extra Trees, Extra Trees is generally faster to train and test than KNN for larger datasets.

Table 2. Parameters and performance metrics of all classification models.

Classifier	Parameters	Training time (Sec./Fold 1)	Testing time (Sec./Sample)	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)
SVC	Kernel = Radial Basis Function C = 11 Degree = 3	1 m 20 s	5 m	90.17	93	90	89
XGB	Learning_rate = 0.01 No. of trees = 25 Max_depth = 15	17 m 32 s	10 m 18 s	91.5	92	92	91
KNN	N_neighbors = 1 Algorithm = "Auto" Leaf_size = 30 Weights = "distance"	2 m 4 s	6 s	98.68	99	99	99
Extra Trees	No. of trees = 100 Criterion = gini Max_depth = 2 Min_samples_split = 2	1 m 6 s	3 s	99.54	100	100	100

Furthermore, Extra Trees achieved a higher accuracy score of 99.54% and perfect precision, recall, and F-score of 100%, while KNN achieved slightly lower performance overall with an accuracy score of 98.68%. So, both KNN and Extra Trees appear to be highly effective classifiers for the given dataset, and hence for this type of problem. However, it should require further evaluation analysis to determine the best classifier for the proposed work.

Figure 5 presents the testing accuracy results for each of the 30 classes using both KNN and ET classifiers. The results suggest that the Extra Trees (ET) classifier generally outperforms K-Nearest Neighbors (KNN) in discriminating between the 30 classes for the given dataset. In addition, the performance metrics for most of the classes have been improved. As observed, the minimum precision scores in the KNN model were 93% for both classes, "are" and "to", but improved in the ET model to 96% and 98%, respectively.

In addition to performance analysis, the computational requirements should be discussed to ensure that the classifier is practical and feasible for the given task. The time complexity for testing Extra Trees (ET) is $O(\log(N) * M)$, where N is the number of training points and M is the number of test points. This is because the ET algorithm traverses a decision tree to classify each test point, and the depth of the tree is typically logarithmic in the number of training points [23].

In contrast, the time complexity for testing K-Nearest Neighbors (KNN) is $O(N * M)$. This is because KNN requires calculating the distance between each test point and all the training points, which can be computationally expensive for large datasets or when the number of stored training data points is large [23]. Therefore, in general, the time complexity of testing ET is much lower than KNN for large datasets, which can result in faster testing times and make ET a more suitable classifier for some applications.

Accuracy of K-NeighborsClassifier: 98.68637902619656					Accuracy of Extra Trees: 99.37594384184142				
	precision	recall	f1-score	support		precision	recall	f1-score	support
Doaa	0.98	1.00	0.99	5343	Doaa	0.99	1.00	1.00	5343
Dr	1.00	0.98	0.99	3146	Dr	1.00	0.97	0.98	3146
Eight	0.99	1.00	0.99	5761	Eight	1.00	1.00	1.00	5761
Feels	0.93	0.98	0.96	4238	Feels	0.93	1.00	0.96	4238
Five	1.00	1.00	1.00	5713	Five	1.00	1.00	1.00	5713
Four	0.94	1.00	0.97	2689	Four	0.99	1.00	0.99	2689
Gestures	1.00	1.00	1.00	8095	Gestures	1.00	1.00	1.00	8095
Good	1.00	1.00	1.00	4321	Good	1.00	1.00	1.00	4321
Heard	0.97	0.98	0.97	4174	Heard	1.00	1.00	1.00	4174
Hello	0.97	0.97	0.97	4579	Hello	0.98	0.98	0.98	4579
How	1.00	0.98	0.99	5419	How	1.00	1.00	1.00	5419
I	0.99	0.99	0.99	4734	I	0.99	1.00	1.00	4734
LifeSign	1.00	1.00	1.00	4326	LifeSign	1.00	1.00	1.00	4326
Love	1.00	1.00	1.00	3427	Love	1.00	1.00	1.00	3427
Name	1.00	0.99	0.99	5561	Name	1.00	0.99	1.00	5561
Nine	0.99	0.98	0.99	5361	Nine	1.00	1.00	1.00	5361
NotSure	1.00	1.00	1.00	5585	NotSure	1.00	1.00	1.00	5585
One	1.00	1.00	1.00	4969	One	1.00	1.00	1.00	4969
Seven	1.00	0.97	0.98	6610	Seven	1.00	1.00	1.00	6610
Six	1.00	0.99	1.00	4238	Six	1.00	1.00	1.00	4238
Ten	1.00	1.00	1.00	4990	Ten	1.00	1.00	1.00	4990
Thanks	0.98	0.90	0.94	5450	Thanks	0.98	0.96	0.97	5450
Three	0.99	0.99	0.99	4677	Three	0.98	1.00	0.99	4677
To	0.93	1.00	0.96	4118	To	0.98	1.00	0.99	4118
Translate	0.98	0.99	0.99	4024	Translate	1.00	0.95	0.97	4024
Two	0.96	1.00	0.98	5054	Two	1.00	1.00	1.00	5054
We	1.00	0.99	0.99	4121	We	1.00	0.99	0.99	4121
Words	1.00	0.99	0.99	4472	Words	1.00	0.99	1.00	4472
You	1.00	0.99	0.99	5224	You	1.00	0.99	0.99	5224
Zero	0.99	0.95	0.97	4600	Zero	1.00	0.99	1.00	4600
accuracy			0.99	145019	accuracy			0.99	145019
macro avg	0.99	0.99	0.99	145019	macro avg	0.99	0.99	0.99	145019
weighted avg	0.99	0.99	0.99	145019	weighted avg	0.99	0.99	0.99	145019

Figure 5. Performance analysis of both KNN and ET models for all classes.

After the training process, the efficient weights of the pre-trained ET classifier will be converted into a suitable and lightweight format (JSON, and Bin files) to be transmitted to the mobile application to be able to achieve the prediction process instead of the glove system. Hence, the data from the sensors-based glove is sent wirelessly via Bluetooth module to the mobile application “Life-Sign” linked with the classifier that predicts the sign and displays it on the phone.

2.3. Mobile Application “LifeSign”

The mobile application provides a user-friendly interface for users to interact with the sign language recognition system. It was installed on the user’s smartphone and includes the following features.

Bidirectional communication feature: provides an effective and practical solution for enabling communication between signers and non-signers through both modes, speaking and listening as depicted in Figure 6(a). The speaking mode

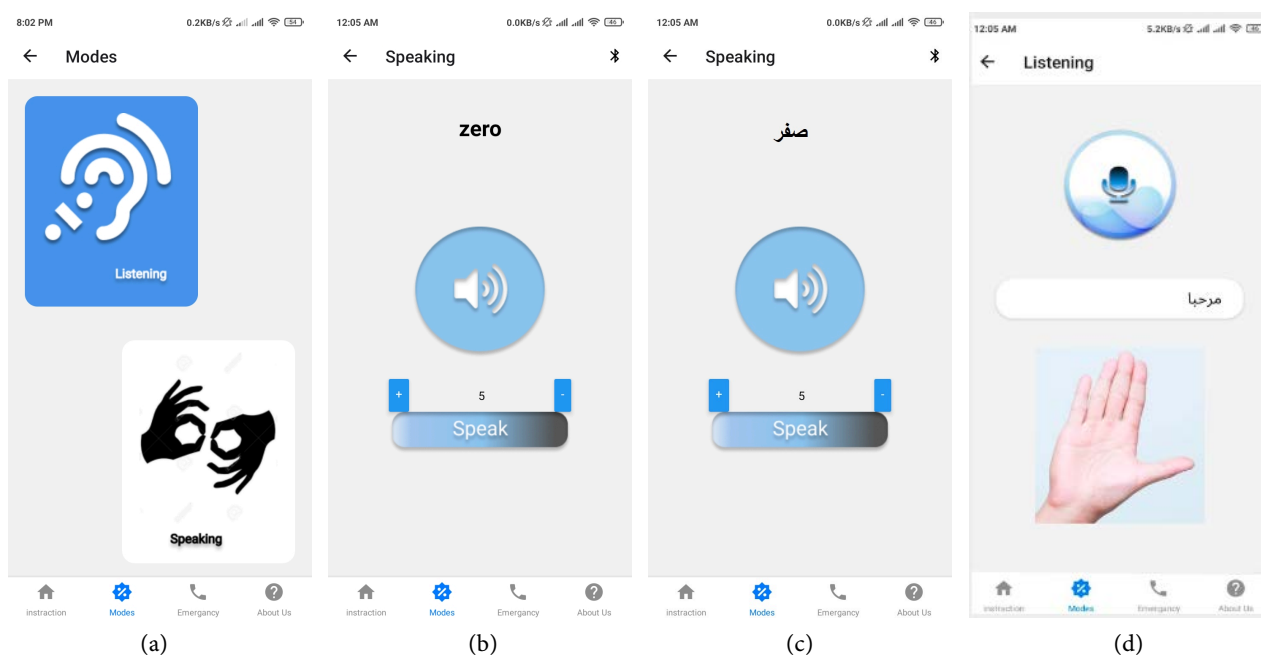


Figure 6. The mobile application “LifeSign” with its two modes. (a) LifeSign Application. (b) Speaking Mode “Arabic-added language”. (c) Listening Mode.

converts the sign language gestures into both voice and text format as in **Figure 6(b)**. While in the listening mode, the user speaks into the mobile application, and the system converts the spoken language input into both text and picture format as in **Figure 6(c)**.

The use of advanced machine learning algorithms and speech synthesis technology in the speaking mode ensures accurate and reliable recognition of sign language gestures, while the use of speech-to-text conversion technology and picture format display in the listening mode improves the comprehension of spoken words for less educated users. Hence, it is accessible to a wide range of users, regardless of their level of experience with sign language or technology.

Smart optimization feature: enhance the system’s performance by controlling the size of data transferred via Bluetooth and improving the speed of gesture recognition. This feature involves a simple controlling algorithm for the size of the transferred data via Bluetooth, which results in faster recognition of consecutive gestures. By reducing the size of data transmitted via Bluetooth, the system can achieve faster recognition of consecutive gestures, which is crucial for effective communication.

Localization feature “Arabic-added language”: provides better accessibility and convenience for users with diverse backgrounds and needs that help to overcome language and cultural barriers. The system provides with ability to install the language packs on the phone that allow users to select the desired language which appears to others on the user interface as shown in **Figure 6(b)**.

Online or offline working feature: users can switch between online and offline modes as needed, depending on their available resources, internet connectivity,

and other factors. Online mode allows the system to access cloud-based databases, enabling it to allow the system to perform real-time updates, ensuring that the system remains up-to-date with the latest sign language gestures and other data. Offline mode, on the other hand, allows the system to operate independently of external resources, making it more convenient and accessible for users who may not have access to the internet or who prefer not to rely on cloud-based services. In offline mode, the system stores and processes data locally on the user's mobile device, providing a more secure and private solution for sign language recognition.

3. Experimental Results

The study reported the proposed glove-based system as a commercial product for the Arab deaf-mute community, using the Egyptian case study as a basis. It was important to assess how well the suggested algorithms would work in predicting and recognizing a series of gestures with high accuracy and minimal delay. Additionally, the primary goal of this study is to propose a real-time Egyptian sign language recognition system for a sample of real-time data. Therefore, extensive testing was conducted to ensure the system's capability to recognize different consecutive gestures. There are 11 numbers from zero to ten and 10 different expressions of 19 words taken from Egyptian sign language are used in examining the proposed system.

The proposed sign language recognition system underwent evaluation using a dataset consisting of ten users executing various gestures and expressions. Each gesture was performed ten times, resulting in a total of 100 instances for each gesture. The system's performance was assessed based on accuracy and the delay observed between consecutive gestures, considering different input data sizes.

The proposed sign language recognition system can employ different reading techniques based on the input data size, presenting a trade-off between accuracy and delay between consecutive gestures. The choice of reading technique depends on the specific use case and user preferences. The system's performance has been evaluated with input data sizes ranging from three rows to eight rows. Three rows provided an acceptable reading accuracy with a delay of 50 ms, while eight rows yielded higher accuracy but with a delay of 82 ms, as indicated in **Table 3**.

The delay between consecutive gestures may concern some users, as it can impact real-time communication between signers and non-signers. The reading technique employed by the proposed sign language recognition system results in an increasing delay as the number of reading rows increases. This is due to the need to process a larger amount of data to accurately recognize the sign language gesture, which takes more time. However, advancements in hardware platforms for future generations can mitigate these delay issues by offering faster processing and response times.

Finally, a performance comparison of the accuracy of the proposed models with those of existing systems is presented in **Table 4**. It is noteworthy that the

Table 3. Experimental real-time consecutive data accuracy with the corresponding delay.

Input data size	Accuracy %	Delay in msec
Three rows	90	50
Five rows	94	67
Eight rows	96	82

Table 4. Comparison of the proposed work with others.

Classifier/Method	Interface	Dataset type/size	Accuracy %	
Proposed Work	Sensor-based glove	SVM	Arabic-added language	90.17
		XGB	30 different static and dynamic gestures	91.5
		KNN	Original dataset size 1,029,801	98.68
		ET	10 performers – 11 features	99.54
	[4] ANN	Sensor-based glove	Malaysian sign language	99
	RF		21 static gestures	99
	KNN		4 performers – 64 features	99
	DT		data size 2100	97
	SVM			89
Others	Wearable IMU system	[5] CNN	Indian Sign Language	87.99
		CapsNet 3 rout.	20 sentences with each consisting	94
		CapsNet 5 rout.	of 2 - 4 signs- 10 performers	92.5
[8] Multi-label Classification	Wearable sensors	LP-MC	Indian sign language	No acc.
			100 isolated signs	Categ. error of
			10 performers – 20,000 samples	2.73%

ET classifier, despite not being previously utilized, demonstrates superior performance compared to other models, except for the model mentioned in [17]. However, the accuracy of the model in [17] is considered unreliable due to the small size of their dataset. Therefore, the ET classifier can be deemed as the optimal choice for the suggested system.

4. Conclusion

A proposed lower-cost sign language recognition based-glove system is designed for commercial use by the deaf-mute community. An innovative real-time sign multi-language recognition of integrates advanced sensing, as well as faster performance, has been designed and practically implemented. Four classifiers were examined, the Extra Trees (ET) classifier is the best algorithm, and the gestures successfully prediction is about 99.54% average accuracy. The LifeSign user-friendly interface smart features gave the system the potential for flexibly extendable communication for the deaf-mute community. This allows them to connect with other nations and participate fully in society. Moreover, it offers some salient features such as; affordability, robustness, privacy, immunity to en-

vironmental factors, precise capturing of movements, portability and convenience, and cost-effectiveness. Fast system consecutive gestures successfully recognition of 50 ms has been experimentally realized. Additionally, an offline mode of operation adds value for faster performance and system portability. Finally, despite the features of this work, it is constrained by its concentration on a particular subset of sign language gestures; therefore, future research should aim to broaden the range of recognized gestures.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Alsaadi, Z., Alshamani, E., *et al.* (2022) A Real Time Arabic Sign Language Alphabets (ArSLA) Recognition Model Using Deep Learning Architecture. *Computers*, **11**, Article No. 78. <https://doi.org/10.3390/computers11050078>
- [2] Mustafa, M. (2020) A Study on Arabic Sign Language Recognition for Differently Abled Using Advanced Machine Learning Classifiers. *Journal of Ambient Intelligence and Humanized Computing*, **12**, 4101-4115. <https://doi.org/10.1007/s12652-020-01790-w>
- [3] Suharjito, M.C.A., Wiryana, F. and Kusuma, G.P. (2018) A Survey of Hand Gesture Recognition Methods in Sign Language Recognition. *Pertanika Journal of Science and Technology*, **26**, 1659-1675.
- [4] Alrubayi, A.H., Ahmed, M.A., Zaidan, A.A., Albahri, A.S., *et al.* (2021) A Pattern Recognition Model for Static Gestures in Malaysian Sign Language Based on Machine Learning Techniques. *Computers and Electrical Engineering*, **95**, Article ID: 107383. <https://doi.org/10.1016/j.compeleceng.2021.107383>
- [5] Suri, K. and Gupta, R. (2019) Continuous Sign Language Recognition from Wearable IMUs Using Deep Capsule Networks and Game Theory. *Computers and Electrical Engineering*, **78**, 493-503. <https://doi.org/10.1016/j.compeleceng.2019.08.006>
- [6] Deriche, M., Aliyu, S.O. and Mohandes, M. (2019) An Intelligent Arabic Sign Language Recognition System Using a Pair of LMCs with GMM Based Classification. *IEEE Sensors Journal*, **19**, 8067-8078. <https://doi.org/10.1109/JSEN.2019.2917525>
- [7] Abdul, W., Alsulaiman, M., Amin, S.U., Faisal, M., Ghaleb, H., *et al.* (2021) Intelligent Real-Time Arabic Sign Language Classification Using Attention-Based Inception and BiLSTM. *Computers and Electrical Engineering*, **95**, Article ID: 107395. <https://doi.org/10.1016/j.compeleceng.2021.107395>
- [8] Gupta, R. and Kumar, A. (2020) Indian Sign Language Recognition Using Wearable Sensors and Multi-Label Classification. *Computers and Electrical Engineering*, **90**, Article ID: 106898. <https://doi.org/10.1016/j.compeleceng.2020.106898>
- [9] Hassan, M., Assaleh, K. and Shanableh, T. (2019) Multiple Proposals for Continuous Arabic Sign Language Recognition. *Sensing and Imaging*, **20**, Article No. 4. <https://doi.org/10.1007/s11220-019-0225-3>
- [10] Rizwan, S.B., Khan, M.S.Z. and Imran, M. (2019) American Sign Language Translation via Smart Wearable Glove Technology. *International Symposium on Recent Advances in Electrical Engineering (RAEE)*, Islamabad, 28-29 August 2019, 1-6. <https://doi.org/10.1109/RAEE.2019.8886931>

- [11] Lee, B.G. and Lee, S.M. (2018) Smart Wearable Hand Device for Sign Language Interpretation System with Sensors Fusion. *IEEE Sensors Journal*, **18**, 1224-1232. <https://doi.org/10.1109/JSEN.2017.2779466>
- [12] Saleh, N., Farghaly, M. and Elshaaer, E. (2020) Smart Glove-Based Gestures Recognition System for Arabic Sign Language. *International Conference on Innovative Trends in Communication and Computer Engineering (ITCE)*, Aswan, 8-9 February 2020, 303-307. <https://doi.org/10.1109/ITCE48509.2020.9047820>
- [13] Mummadi, C.K., Leo, F.P., *et al.* (2018) Real-Time and Embedded Detection of Hand Gestures with an IMU-Based Glove. *Informatics*, **5**, Article No. 28. <https://doi.org/10.3390/informatics5020028>
- [14] Panda, A.K., Chakravarty, R. and Moulik, S. (2021) Hand Gesture Recognition Using Flex Sensor and Machine Learning Algorithms. *Conference on Biomedical Engineering and Sciences (IECBES)*, Langkawi Island, 1-3 March 2021, 449-453. <https://doi.org/10.1109/IECBES48179.2021.9398789>
- [15] Montalvo, P.D.R., Godoy-Trujillo, P., *et al.* (2018) Sign Language Recognition Based on Intelligent Glove Using Machine Learning Techniques. *IEEE 3rd Ecuador Technical Chapters Meeting (ETCM)*, Cuenca, 15-19 October 2018, 1-5.
- [16] Tharwat, G., Ahmed, A.M. and Bouallegue, B. (2021) Arabic Sign Language Recognition System for Alphabets Using Machine Learning Techniques. *Journal of Electrical and Computer Engineering*, **2021**, Article ID: 2995851. <https://doi.org/10.1155/2021/2995851>
- [17] Amin, M.S., Latif, M.Y., *et al.* (2021) Alphabetical Gesture Recognition of American Sign Language Using e-Voice Smart Glove. *IEEE 23rd International Multitopic Conference (INMIC)*, Bahawalpur, 5-7 November 2020, 1-6. <https://doi.org/10.1109/INMIC50486.2020.9318185>
- [18] Ibrahim, N.B., Selim, M.M. and Zayed, H.H. (2018) An Automatic Arabic Sign Language Recognition System. *Journal of King Saud University—Computer and Information Sciences*, **30**, 470-477. <https://doi.org/10.1016/j.jksuci.2017.09.007>
- [19] Ahmed, A.M., Abo Alez, R., Tharwat, G., Taha, M., Belgacem, B. and Al Moustafa, A.M.J. (2020) Arabic Sign Language Intelligent Translator. *4e Imaging Science Journal*, **68**, 11-23. <https://doi.org/10.1080/13682199.2020.1724438>
- [20] Elakkiya, R. and Selvamani, K. (2019) Subunit Sign Modeling Framework for Continuous Sign Language Recognition. *Computers and Electrical Engineering*, **74**, 379-390. <https://doi.org/10.1016/j.compeleceng.2019.02.012>
- [21] Ding, I., Lin, R.-Z. and Lin, Z.-Y. (2018) Service Robot System with Integration of Wearable Myo Armband for Specialized Hand Gesture Human-Computer Interfaces for People with Disabilities with Mobility Problems. *Computers and Electrical Engineering*, **69**, 815-827. <https://doi.org/10.1016/j.compeleceng.2018.02.041>
- [22] Chuang, W.-C., Hwang, W.-J., *et al.* (2019) Continuous Finger Gesture Recognition Based on Flex Sensors. *Sensors*, **19**, Article No. 3986. <https://doi.org/10.3390/s19183986>
- [23] Bonaccorso, G. (2017) *Machine Learning Algorithms*. Packt Publishing, Birmingham.