مجلة كلية الآداب بقنا (دورية أكاديمية علمية محكمة)

Alphabet Recognition in Arabic Sign Language: A Machine Learning Perspective

Dr. Mahmoud M. Khattab

Prof. Akram M. Zeki

Dr. Safaa S. Matter

Prof. Mohamed A. Abdella

Dr. Rada A. E. Atiia

Dr. Amr Mohmed Soliman

DOI: 10.21608/qarts.2024.267418.1882

مجلة كلية الآداب بقنا – جامعة جنوب الوادي – المجلد (33) العدد (62) يناير 2024

الترقيم الدولي الموحد للنسخة المطبوعة ISSN: 1110-614X

الترقيم الدولي الموحد للنسخة الإلكترونية ISSN: 1110-709X

موقع المجلة الإلكتروني: https://qarts.journals.ekb.eg

Alphabet Recognition in Arabic Sign Language: A Machine Learning Perspective

Mahmoud M. Khattab[1], Akram M. Zeki[1], Safaa S. Matter[2], Mohamed A. Abdella[3], Rada A. E. Atiia[4], Amr Mohmed Soliman[4]

[1]Faculty of Information and Communication Technology, International Islamic University Malaysia, Kuala Lumpur, Malaysia
[2]Department of Computer Science, Applied College, King Khalid University, Abha, Saudi Arabia
[3]Faculty of Social Sciences, Imam Muhammad Ibn Saud Islamic University, Saudi Arabia
[4]College of Education, King Khalid University, Abha, Saudi Arabia

Abstract:

Pattern recognition in human-computer interaction systems has gained significant attention in recent years, particularly in computer vision and machine learning applications. One prominent application is the recognition of hand gestures used in communication with deaf individuals, specifically in identifying the dashed letters within Quranic surahs. This paper proposes a new alphabet-based Arabic sign language recognition model, which employs a vision-based approach. The system comprises four stages: data acquisition, data preprocessing, feature extraction, and classification. The proposed model accommodates three types of datasets: bare hands against a dark background, bare hands against a light background, and hands wearing dark-colored gloves. The process begins with capturing an image of the alphabet gesture, followed by hand separation and background isolation. Hand features are then extracted based on the chosen method. In terms of classification, supervised learning techniques are employed to classify the 28-letter Arabic alphabet using 9,240 images. The focus is on classifying the 14 alphabetic letters representing the initial Quranic surahs in the Quranic sign language. The experimental results demonstrate that the new proposed model has achieved an impressive accuracy of 99.5% using the k nearest neighbor classifier.

Keywords: ArSL; feature extraction; gestures; machine learning; classification.

1. Introduction

Sign language (SL) evolves organically within the deaf community, with each SL having its own rules. However, there is a lack of understanding of SL outside the deaf community, leading to significant communication difficulties between deaf and hearing individuals. This issue is particularly challenging when deaf children are born into hearing families, creating a language gap within the family. Furthermore, the absence of standardized SLs adds to the difficulty of teaching SL to deaf individuals [1]. Deaf people also face challenges in accessing religious teachings, such as the inability of deaf Muslims to learn and understand the Holy Quran [1].

SL encompasses descriptive and nondescriptive signs, as well as fingerspelling using the alphabet. However, Arabic sign language (ArSL) lacks standardized coordination, posing a significant obstacle to learning and translation efforts within Arab deaf communities [2]. SLs vary across regions and dialects, including Egyptian, Jordanian, Tunisian, and Gulf SLs. Although these languages may share some signs, the lack of education and communication between deaf and hearing individuals creates gaps in understanding [2].

These challenges in ArSL, along with the differences between SL and spoken language, highlight the need for machine translation between ArSL and spoken Arabic. Additionally, the development of ArSL recognition systems can facilitate deaf individuals' integration into education and enable them to access scientific knowledge using their native language [3]. Generally, ArSL involves three levels: recognizing gestures of the Arabic alphabet, isolated gestures at the word level, and continuous gestures at the sentence level [4].

This paper focuses on alphabetic ArSL and their potential to assist the deaf community in overcoming communication challenges and learning Arabic alphabets, particularly those used in the 29 Quranic surahs with dashed letters, which are 14 letters: " ح ط ص س ك ر ق ا ه ن م ل ع ي". Previous research indicates that automatic ArSL recognition systems have limited performance accuracy and face significant limitations. The paper presents a technique for identifying the Arabic alphabet, with key contributions including the formation and design of datasets consisting of 8,400 True (RGB) images for supervised learning, the development of an effective system for recognizing fixed signs in the ArSL alphabet, and the integration of features from different signers to enhance recognition accuracy.

The paper is structured as follows: a literature review in Section 2, a discussion of hearing-impaired communication methods and the similarities between finger spelling and the Arabic alphabet in Section 3, a description of the research methodology and system design in Section 4, the experimental results in Section 5, and the concluding results in Section 6.

2. Literature Reviews

The research landscape in SL recognition systems, both globally and within the Arab world, has witnessed significant development in recent years, often focusing on either vision-based or sensor glove-based approaches. This study specifically delves into vision-based systems, particularly those aimed at identifying ArSL alphabets. A comprehensive review of pertinent literature from the last decade reveals notable contributions in this domain.

Zabulisy et al. [5] proposed a vision-based hand gesture recognition system for Human-Computer Interaction, while Mohandes [6-9] introduced a prototype system employing support vector machine

(SVM) for ArSL recognition and an automatic translation system from Arabic text to SL. AlJarrah and Halawani [10] developed a neuro-fuzzy system achieving a recognition rate of 93.55%, whereas Al-Rousan and Hussain [11] introduced an adaptive neuro-fuzzy interference system with a recognition accuracy of 95.5%. Assaleh and Al-Rousan [12] utilized polynomial classifiers for alphabet recognition, attaining error rates of 1.6% and 6.59% for training and testing data, respectively. Maraqa and Abu-Zaiter [13] employed Elman and fully recurrent networks, achieving accuracy rates of 89.66% and 95.11%, respectively. El-Bendary et al. [14] developed an Arabic alphabet signs translator with up to 91.3% accuracy using multilayer perceptron (MLP) neural networks and minimum distance classifiers (MDC). Hemayed and Hassanien [15] introduced a recognition system converting signs to speech, incorporating principal component analysis (PCA) and k nearest neighbor (KNN) classification. Ahmed et al. [16] proposed an automatic translation system for ArSL, emphasizing manual detection techniques and statistical classifiers.

The field witnesses ongoing advancements with research focusing on efficient interface systems, leveraging emerging techniques such as graph convolutional neural networks to enhance accuracy and functionality in real-world applications, particularly for deaf and dumb learning initiatives [17].

3. Communication Methods for the Hearing-Impaired

SLs are the primary means of communication for deaf communities, evolving naturally within these communities. ArSL, American SL, and French SL are examples of distinct SLs used in corresponding regions. While SLs differ from spoken languages, the main distinction lies in how communication elements are produced and perceived. In addition to SLs, deaf communities use other communication methods such as fingerspelling and cued speech [18]. Fingerspelling involves visually representing the alphabet using one or both hands. It serves various purposes in ArSL, such as expressing words without equivalent signs, spelling names, and supporting reading and writing skills in Arabic for deaf individuals. Table 1 illustrates the similarities between ArSL alphabet symbols and the corresponding Arabic letters [19].

Cued speech is a communication method based on phonetics and the characteristics of spoken languages. It utilizes lip movements and hand gestures to represent phonemes, assisting deaf individuals in understanding spoken languages and overcoming challenges in lip reading. By replacing the invisible vocal components involved in sound production (vocal cords, tongue, and jaw) with hand gestures while keeping the visible elements (lips), cued speech enables the recognition of phonemes with similar lip shapes or movements [20]. This method exchanges audio information through lip and hand configurations.

Table 1: The similarities between ArSL alphabet symbols (S) and the corresponding Arabic letters

S	Similarity	S	Similarity
ا	Matches letter "أ" with thumbs up	ب	Index finger indicates a single point under the shape of "ب"
ت	Index and middle fingers indicate two points above the shape of "ت"	ث	Index, middle, and ring fingers indicate to three points above "ث"
ج	Thumb position indicates a point inside the letter "ج"	ح	Thumb at same level as other finger as there is no point to the letter "ح"

S	Similarity	S	Similarity
خ	Thumb height above other fingers indicating a point above the letter "خ"	د	Matches the shape of a letter "د"with a sharp angle between the thumb and index finger on the left
ذ	Sharp angle among thumb and index finger on left, and middle above index indicates a point on the letter " ذ"	ر	Matches to alphabet shape, thumb with curvature of index finger and other fingers closed, represents the letter "ر"
ز	Matches the shape of the alphabet, where the thumbs with the index finger curve and the middle above it, and the rest of the fingers closed, the shape of the letter "ز"	س	Matches the alphabet "س" with the numerator and affinity of the fingers
ش	No similar form, but disperse fingers indicating dots above "ش"	ص	Near at the format with the letter "ص" with a note placed thumb
ض	Close when the thumb raised indication of a point above letter "ض"	ط	Close match with the letter "ط" and raise the index finger to the top
ظ	Close match with thumbs above the rest of fingers to indicate a point above letter "ظ" and index finger up	ع	Match with alphabet letter "ع"
غ	Match with the shape of the alphabet and raise thumb above palm of the hand to indicate point above letter "غ"	ف	Match the shape of the alphabet "ف" and the index finger position over the thumb to indicate a point above the letter

S	Similarity	S	Similarity
ق	Match the shape of alphabet "ق" with index and middle fingers above the thumb to indicate 2 points above letter	ك	Matches and put thumb at middle of palm of the hand to indicate Hamza "ء" for a character "ك"
ل	Match with alphabet letter "ل"	م	No match with alphabet letter "م"
ن	Matches alphabet letter "ن" with a point above the letter	ه	match with alphabet letter "هـ"
و	Match with alphabet letter "و"	ى	Match with alphabet letter "ي"

4. Research Methodology and System Design

This section presents an overview of the ArSL recognition system based on alphabet letters. The system consists of four main phases: (1) Images or video acquisition, (2) Images or video preprocessing, (3) Features extraction, and (4) Classification and recognition of alphabet letters, as shown in Figure 1. Detailed explanations of each stage will follow.

4.1. Images or Video Acquisition

The first phase involves capturing video using a webcam. Different alphabets from 10 individuals are considered, resulting in 28 variations. To ensure high-accuracy gesture recognition, 100 images are captured for each alphabet and included in the system's databases. The captured images underwent modifications to enhance clarity. The vision-based approach is used, where signers performed gestures while wearing dark-colored gloves in various lighting conditions against a light background, or without gloves against a dark background. The output of this phase is a collection of color pictures (RGB) representing the hand gestures for each letter of ArSL [21-24]. Table 2 provides an overview of the datasets created for training and testing purposes.

Integrated webcams and a smart camera for a Huawei mobile device are utilized to capture consecutive pictures. Four comprehensive datasets specifically designed for ArSL images are developed for the deaf community. These datasets provide researchers and interested individuals with opportunities to explore and develop automated systems for the deaf and hard of hearing, employing machine learning techniques, human-computer interaction, computer vision, and deep learning algorithms. The datasets, consisting of 9240 images of ArSL gestures, are collected from 10 locations, encompassing different age groups and hand sizes. The images exhibit various dimensions, angles, and complex backgrounds. Digital image preprocessing techniques, such as noise removal, centering, resizing, and enhancement, can be applied to these images during the preprocessing stage. Additionally, these datasets facilitate the identification of Quranic surahs that begin with dashed alphabetic characters, aiding deaf individuals in recognizing the Quranic alphabet. The datasets will be publicly accessible to researchers.

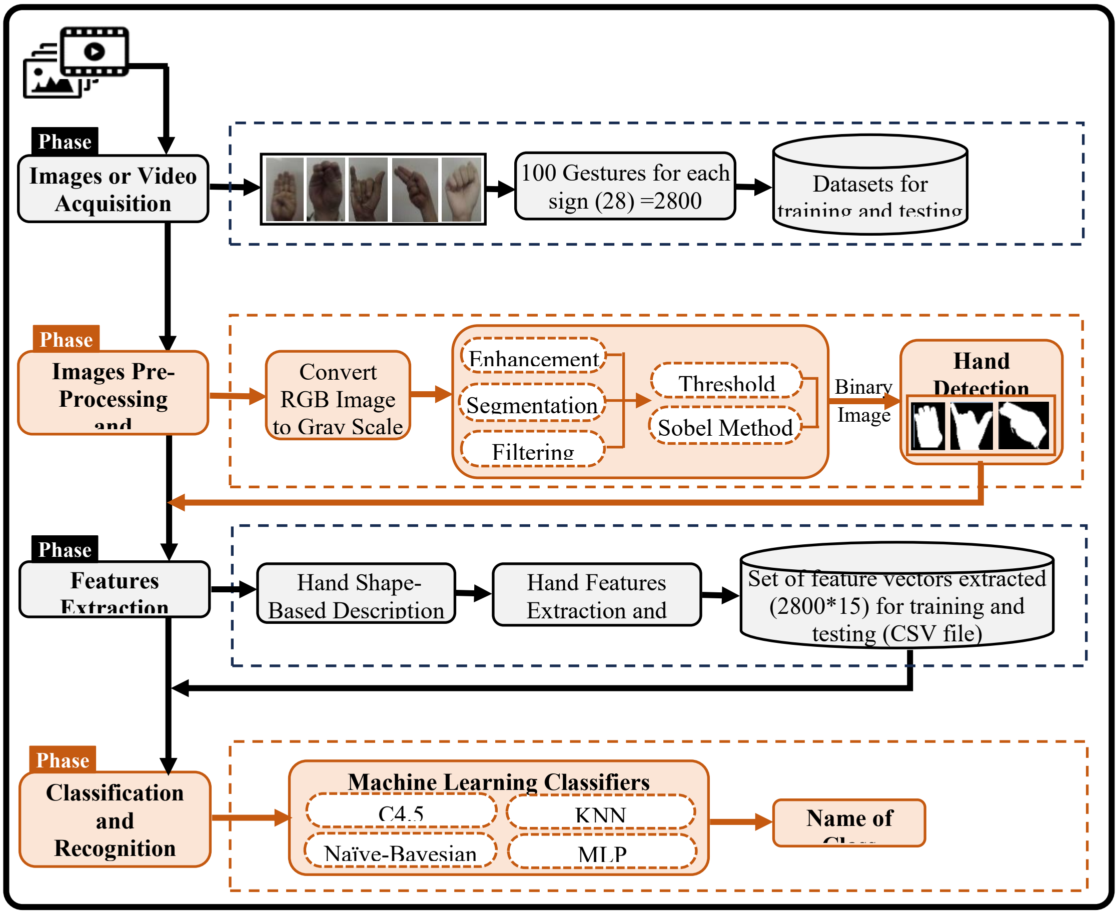


Figure 1: Major phases of the alphabet based ArSL recognition system

4.2. Image Preprocessing and Hand Detection

The second stage of our system involves preprocessing the data to prepare it for further processing [25-32]. To ensure robustness, separate images are taken of multiple signers with different hand sizes and complex backgrounds. These images contain round samples and variable angles between 60 and -60 degrees. Color images are converted to grayscale with 256 density levels and resized to 640x480 pixels. Filtering methods can be applied to remove noise. Therefore, the preprocessing stage aims to convert the data into a more easily processable format. It involves optimization, image enhancement, segmentation, and morphological filtering. The preprocessing operations are crucial for hand detection and extracting the best features to achieve high accuracy [33].

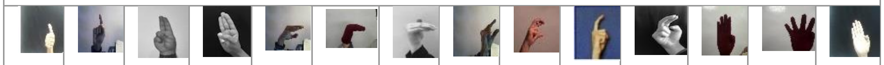
In this phase, we follow several steps to preprocess the gesture alphabet images of ArSL, as follows:

1) *Converting RGB Image to Grayscale (First Step)*

The conversion of an image or video frame from the RGB color space to grayscale constitutes a crucial preprocessing phase in the realm of computer vision. This transformative process serves as a foundational step, enabling the execution of various operations such as image enhancement, segmentation, and morphology. Its primary objective is the mitigation of noise and the facilitation of the identification of significant regions, particularly in the context of hand shape detection within the image. The adoption of grayscale images is pivotal in this context due to the prevalent use of grayscale-focused techniques in image processing. Grayscale representations offer superior clarity and present a more amenable canvas for manipulation, aligning seamlessly with the methodologies employed in diverse image processing applications. The utilization of grayscale as the intermediary state in this workflow ensures not only the compatibility with prevalent techniques but also optimizes the interpretability and malleability of the visual data under examination.

Table خطأ! لا يوجد نص من النمط المعين في المستند.2: ArSL datasets of alphabet letters

Datasets	Description and samples													
Data set 1 (Training Set)	2800 samples of Arabic alphabet gestures for the deaf language, captured with a bare hand against a light background and varying lighting conditions. Each letter of the Arabic alphabet is represented by 100 samples.													
														
	أ	ب	ت	ث	ج	ح	خ	د	ذ	ر	ز	س	ش	ص
														
	ض	ط	ظ	ع	غ	ف	ق	ك	ل	م	ن	ه	و	ي
Data set 2 (Training Set)	2800 samples of Arabic alphabet gestures for the deaf language, captured with a bare hand against a dark background and varying lighting conditions. Each letter of the Arabic alphabet is represented by 100 samples.													
														
	أ	ب	ت	ث	ج	ح	خ	د	ذ	ر	ز	س	ش	ص
														
	ض	ط	ظ	ع	غ	ف	ق	ك	ل	م	ن	ه	و	ي
Data set 3 (Training Set)	2800 samples of Arabic alphabet gestures for the deaf language, captured with a right-hand wearing glove against a white background and varying lighting conditions. Each letter of the Arabic alphabet is represented by 50 samples.													
														
	أ	ب	ت	ث	ج	ح	خ	د	ذ	ر	ز	س	ش	ص
														
	ض	ط	ظ	ع	غ	ف	ق	ك	ل	م	ن	ه	و	ي

Datasets	Description and samples
Data set 4 (Testing Set)	840 samples of Arabic alphabet gestures for the deaf language, captured with two bare hands and gloves against different backgrounds, angles, and lighting conditions. Each letter of the Arabic alphabet is represented by 50 samples.  أ ب ت ث ج ح خ د ذ ر ز س ش ص  ض ط ظ ع غ ف ق ك ل م ن هـ و ي

2) *Image Enhancement, Segmentation, and Morphological Filtering (Second Step)*

In the second step, we perform image enhancement, segmentation, and morphological filtering to convert the grayscale alphabet images to binary. This process adjusts the contrast, removes noise, and detects the edges of the hand in the grayscale image. Two different methods are employed to achieve these goals, and the resulting binary images are obtained by applying various filtering techniques, as follows:

- ***Threshold Method***

  In this method, the image is segmented using a fixed color threshold. Pixels with a density greater than the threshold are assigned a value of 255 (white), representing the background, while pixels with a density less than or equal to the threshold are assigned a value of 0 (black), representing the hand shape. The threshold value can be adjusted based on the color of the gloves used.

- ***Sobel Filter Method***

In this method, the image is segmented using edge detection technology with the Sobel filter. The Sobel filter detects strong edges of the hand based on high-density derivatives. It preserves the shape of the hand and the contours of the fingers. Median filtering is used to adjust image contrast and remove noise, and 2D convolution is applied to enhance the edges.

3) Hand Edge Detection (Third Step)

Following the initial step of segmenting the hand area utilizing one of the two prescribed methods, the subsequent stage involves the extraction of the hand itself. This is achieved through the process of labeling the identified elements and creating a matrix that accurately represents the shape of the hand. From these labeled elements, the area with the most significant extent, which corresponds to the hand, is selectively chosen for further analysis and processing. To refine the hand's boundaries and establish coherence within the extracted region, morphological operations are employed. Specifically, two commonly utilized operations, namely dilation and erosion, are applied.

Dilation, a fundamental morphological operation, serves to expand the boundaries of objects within an image. By convolving a pre-defined structuring element with the image, each pixel location is examined. If any portion of the structuring element overlaps with the object, the corresponding pixel in the resulting image is assigned a value indicating the presence of the object. When applied to the hand region, dilation effectively increases the thickness of the hand's edges, promoting a more distinct and well-defined boundary.

Conversely, erosion, another widely employed morphological operation, effectively contracts the boundaries of objects in an image. Similar to dilation, erosion involves convolving a structuring element with the image, inspecting each pixel location, and verifying if all the pixels within the element overlap with the object. In cases where this condition is met, the corresponding output pixel is assigned the value denoting the object; otherwise, it is assigned the background value. Employing erosion on the hand region results in the thinning or erosion of the edges, aiding in the elimination of noise or minor irregularities within the boundary.

By strategically incorporating these morphological operations, the hand region's edges are refined, enhancing their quality, and subsequent connectivity is established. This process contributes to the overall accuracy and reliability of the hand extraction procedure, facilitating accurate subsequent analysis and interpretation of the hand gestures or features.

4.3. Feature Extraction

After the hand is detected and extracted using the previously described methods, the next phase involves identifying the best features that distinguish each sign gesture from others. These features are crucial for training and testing the dataset. The proposed system focuses on shape-based descriptions, using contour-based or region-based methods. A set of features is selected to describe the hand gestures, as outlined in Table 3. The feature extraction stage is implemented on the training datasets, resulting in numeric data stored in vectors. Each vector represents a sample with 15 feature values, forming a matrix with 2800 rows (representing the examples of alphabet gesture samples) and 15 columns (representing the features of each alphabet sample). The datasets are stored as Excel files with the extension (.csv).

Table 3: The hand features used in our model

Features	Description	Equation
F1	The ratio of hand center of mass (x-axis) to the width of the hand bounding box	F1 = CX / x.width
F2	The ratio of hand center of mass (y-axis) to the length of the hand bounding box	F2 = CY / y.length
F3	The ratio of width of the hand bounding box to the length of the hand bounding box	F3 = x.width / y.length
F4	The ratio of hand area to the area of the hand bounding box	F4 = area (object) / area of bounding box
F5	The ratio of hand perimeter to hand area	F5 = perimeter (object) / area (object)
F6	The ratio of object major axis length to area of bounding box	F6 = object major axis length / area of bounding box
F7	The ratio of object minor axis length to area of bounding box	F7 = object minor axis length / area of bounding box
F8-F15	Extrema (endpoints of hand shape)	F8 = X top-left F9 = X top-right F10 = X right-top F11 = Y right-top F12 = X right-bottom F13 = Y right-bottom F14 = Y left-bottom F15 = Y left-top

4.4. Classification and Recognition

In this stage of the research, the focus shifts towards employing statistical classification and neural network methodologies for the purpose of classifying and identifying alphabet letters based on hand gestures. A classifier is meticulously crafted and subsequently trained utilizing the dataset procured in the preceding phase. The overarching aim is to establish a robust mapping between hand gestures and their corresponding alphabet letters within the context of the Arabic language, as cataloged within the previously generated sign database.

The utilization of supervised learning algorithms becomes pivotal in the classification of novel alphabet gestures. Notably, the implementation encompasses a spectrum of statistical classification algorithms such as C4.5, Naïve-Bayesian, and K-Nearest Neighbors (KNN), alongside the Multilayer Perceptron (MLP) network algorithm. This amalgamation of methodologies facilitates a comprehensive assessment wherein the attained results are meticulously juxtaposed in terms of accuracy and computational speed, thereby facilitating the discernment of the most optimal algorithmic approach for the given task.

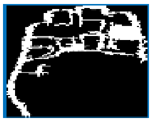
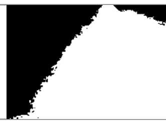
5. Experimental Results and Analysis

In this section, we present the results of our proposed system for recognizing the alphabet in ArSL. The system is capable of translating and recognizing gestures performed using one or both hands, without the need for glove-based sensors or additional devices. We also compare the classification results obtained through the WEKA software tool with those of our proposed system on the three datasets.

5.1. Results of Preprocessing and Hand Detection Phase

In this subsection, we showcase the outcomes of the hand detection process and subsequent extraction from alphabet gestures within the three datasets obtained during the image acquisition phase. We utilize two distinct methods for hand detection, as delineated in Table 4, which displays hand shapes after extraction from background noise. Figure 2 provides a comparative analysis between the two detection methods across the datasets.

Table 4: Samples of hand detection by two methods

Hand Detection Case	Sobel Method		Threshold Method	
Hand Detection				
Hand Detection with noise				
No Hand Detection				

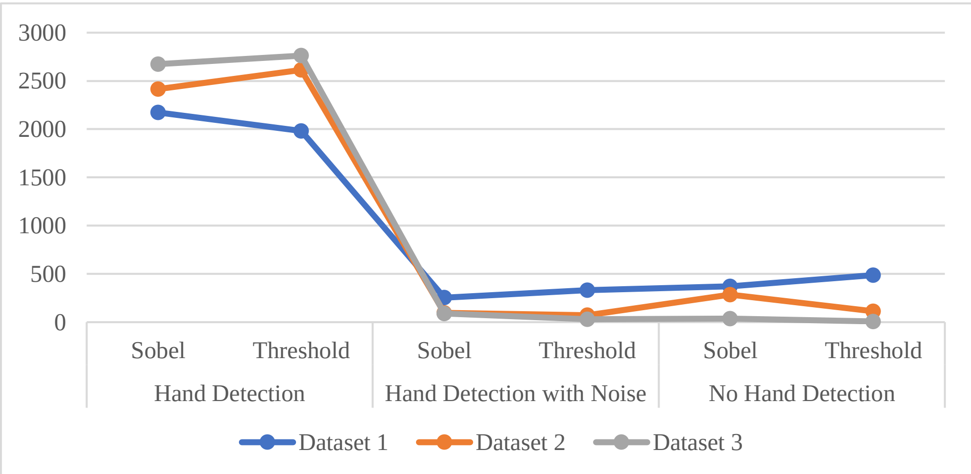


Figure 2: A comparative analysis between the two detection methods across the datasets

The accuracy of hand detection in each method is evaluated by calculating the percentage of successfully detected and isolated hands without noise, relative to the total number of samples (Equation 1). Samples where the hand is detected with the presence of noise are considered unacceptable, as noise significantly impacts the classification phase's accuracy. Table 5 presents the hand detection rate (HDR) for three datasets using two distinct methods: the Sobel method and the Threshold method.

$$HDR = \text{Correct Samples without Noise} / \text{Total Samples} * 100 \quad (1)$$

Table 5: The ratio of the hand detection in each dataset

No. of Dataset	Hand Edges Detection Method	
	Sobel Method	**Threshold Method**
Dataset 1	77.61%	70.79%
Dataset 2	86.29%	93.36%
Dataset 3	95.46%	98.64%

From Table 5, Dataset 1 exhibits a hand detection ratio of 77.61% with the Sobel method and 70.79% with the Threshold method. In Dataset 2, the hand detection ratio improves significantly to 86.29% with the Sobel method and 93.36% with the Threshold method. Dataset 3 shows the highest hand detection ratios, with 95.46% using the Sobel method and 98.64% using the Threshold method.

The table highlights an increasing trend in hand detection ratios from Dataset 1 to Dataset 3 for both detection methods. Dataset 3 consistently demonstrates the highest hand detection ratios compared to the other datasets, indicating superior performance in hand detection. The Threshold method consistently outperforms the

Sobel method across all datasets, showing higher hand detection ratios.

These findings suggest that the Threshold method is more effective in detecting hand edges compared to the Sobel method. Additionally, the increasing hand detection ratios across datasets indicate improvements in image quality or preprocessing techniques. Overall, the table underscores the importance of selecting appropriate detection methods to enhance hand detection accuracy in SL recognition systems.

5.2. Results of SL Classification Phase

In this section, we present the results of the SL recognition using the indicative fingering alphabet. We utilize supervised learning algorithms and three datasets obtained from the extraction of hand shape features for training the classifiers. We conduct three experiments to evaluate different aspects of the classification process, as shown in the following subsections.

5.2.1. Evaluation of Classification Methods

In this experiment, we utilize the WEKA software tool to compare the performance of four classifiers: C4.5, Naive Bayes, KNN, and MLP. The holdout technique is used to estimate the classification error. Table 6 presents the accuracy of letter gestures recognition using the holdout method across different classifiers and rates within three distinct datasets. Each dataset is evaluated under two rates: 66% and 75%.

Table 6: Accuracy of letter gestures recognition using holdout with different classifiers and rates

Dataset No.	C4.5		Naïve-Bayesian		KNN		MLP	
	66%	75%	66%	75%	66%	75%	66%	75%
Dataset 1	83.65%	78.00%	84.11%	83.04%	90.58%	88.38%	86.90%	84.19%
Dataset 2	94.05%	89.57%	89.50%	88.90%	97.67%	96.81%	95.72%	95.28%
Dataset 3	97.74%	95.05%	97.73%	98.24%	99.62%	99.02%	98.71%	98.59%

Initially in Dataset 1, the classifiers exhibit accuracies ranging from 78.00% to 90.58%. Notably, the KNN classifier demonstrates consistent high performance, achieving accuracies between 83.04% and 90.58%. Additionally, the MLP classifier performs competitively, with accuracies ranging from 84.19% to 86.90%.

While in Dataset 2, improvements in accuracy are observed across all classifiers compared to Dataset 1. The accuracies range from 88.90% to 97.67%, indicating a notable enhancement. Particularly noteworthy is the consistent high accuracy of the KNN classifier, which ranges from 88.90% to 97.67%.

Finally, Dataset 3 showcases the highest accuracy among all datasets and classifiers. With accuracies ranging from 95.05% to 99.62%, Dataset 3 demonstrates a significant improvement over the previous datasets. Once again, the KNN classifier stands out for its superior performance, achieving accuracy rates ranging from 95.05% to 99.62%.

The findings from Table 6 underscore the correlation between dataset size, holdout rate, and recognition accuracy for SL gestures. As the dataset size increases and holdout rates improve, there is a corresponding enhancement in recognition accuracy. Moreover, the consistent superior performance of the KNN classifier across all datasets and rates highlights its effectiveness in recognizing letter gestures, which led us to utilize it for classifying a sample representing the dashed letters of the earliest Quranic surahs. These insights emphasize the critical importance of dataset quality and appropriate classifier selection in enhancing the accuracy of SL gesture recognition systems.

5.2.2. Impact of Hidden Neurons on MLP Accuracy

In this experiment, we investigate the impact of varying the number of neurons in the hidden layer (H) on the accuracy (P) and classification time (T) of the MLP network classifier across different datasets, as shown in Table 7.

Table 7: The effect of neurons number in MLP network on the accuracy and time of classification

Dataset No.	H=10		H=15		H=20		H=28	
	T	P	T	P	T	P	T	P
Dataset 1	10.32	86.14	16.34	86.99	21.03	86.35	29.36	89.01
Dataset 2	13.39	91.95	16.69	94.77	24.91	95.52	31.1	96.1
Dataset 3	13.76	97.65	18.46	98.05	26.43	98.41	32.62	98.52

From Table 7, as the number of neurons increases, there is a noticeable improvement in the accuracy of classification. For instance, in Dataset 1, the accuracy increases from 86.14% with 10 neurons to 89.01% with 28 neurons. Similar trends are observed in Dataset 2 and Dataset 3, where higher neuron counts correspond to higher accuracy rates. However, the increase in accuracy comes at the cost of longer classification times. As depicted in the table, the classification time (T) also rises with the number of neurons. For instance, in Dataset 1, the classification time increases from 10.32 units with 10 neurons to 29.36 units with 28 neurons. This pattern holds true across all datasets, indicating a trade-off between accuracy and computational efficiency.

Generally, Table 7 underscores the importance of selecting an appropriate number of neurons in the hidden layer to balance classification accuracy and computational resources. While higher neuron counts enhance classification accuracy, they also lead to longer processing times. Therefore, achieving an optimal balance between accuracy and efficiency is crucial when designing neural network classifiers for real-world applications.

5.2.3. Comparison of Classifier Performance inside the proposed model

In this experiment, we compare the performance of a set of classifiers in terms of the number of correctly and incorrectly classified instances for dataset 3, as shown in Table 8.

Table 8: Comparing the performance of the classifiers used in the proposed model

Classifier Name	Cross-Validation with Flod=10			
	Correctly Classified Instances		Incorrectly Classified Instances	
	Number	Percentage	Number	Percentage
C4.5	2736	97.7143 %	64	2.2857 %
Naïve-Bayesian	2418	86.3571 %	382	13.6429 %
KNN	2792	99.7143 %	08	0.2857 %
MLP	2647	94.5357 %	153	5.4643 %

From Table 8, KNN exhibits the highest accuracy, correctly classifying 99.71% of instances. In contrast, Naive Bayes achieves the lowest accuracy, with 86.36% of instances classified correctly. The KNN classifier's superior performance is reflected in its ability to correctly classify instances while minimizing misclassifications. With only 8 instances incorrectly classified, KNN demonstrates its effectiveness in accurately predicting outcomes based on neighboring data points. Conversely, Naive Bayes exhibits a lower accuracy rate, with 13.64% of instances misclassified. While Naive Bayes is known for its simplicity and ease of implementation, it may struggle with complex datasets or when assumptions about independence among features are violated. The results for C4.5 and MLP classifiers fall between those of KNN and Naive Bayes. C4.5 achieves a classification accuracy of 97.71%, while MLP achieves 94.54%. These classifiers offer reasonable accuracy but may not match the performance of KNN in this context.

Therefore, Table 8 highlights the importance of selecting an appropriate classifier based on the specific characteristics of the dataset and the desired outcome. While KNN demonstrates superior performance in dataset 3, the choice of classifier should consider factors such as dataset complexity, computational efficiency, and the nature of the classification problem.

6. Recommendations for Deaf Community Empowerment

Our experimental endeavors yield helpful insights that can be harnessed to improve communication challenges and foster learning opportunities within the deaf community. The advancements in ArSL recognition systems, as explained through our findings, offer a promising opportunity to empower individuals with hearing impairments. By using the outcomes of our experiments, adapted recommendations emerge to catalyze meaningful interventions:

1) *Integration of ArSL Recognition Systems:*

The robust performance exhibited by ArSL recognition systems underscores their potential as pivotal tools for enhancing communication within the deaf community. It is imperative to integrate these systems into educational frameworks and daily interactions to facilitate seamless communication exchanges. By employing the capabilities of our proposed system, which eliminates the need for glove-based sensors, individuals can effortlessly translate and comprehend ArSL gestures, thereby bridging communication divides.

2) *Customized Learning Platforms:*

The proficiency attained in recognizing alphabetic gestures in ArSL paves the way for the development of designed learning platforms. These platforms can be tailored to accommodate diverse learning styles and proficiency levels, catering specifically to the needs of individuals navigating the distinctions of ArSL. By employing supervised learning algorithms and meticulously selected datasets, educational modules can be designed to facilitate learning experiences. Such platforms hold huge potential to empower individuals in learning ArSL, thereby fostering inclusivity and empowerment within the deaf community.

3) *Empowerment through Technological Innovation:*

Adopting technological innovation emerges as a cornerstone in overcoming communication barriers and fostering holistic development within the deaf community. The advent of ArSL recognition systems signs a paradigm shift in augmenting accessibility and inclusivity. It is imperative to harness these advancements to develop intuitive applications and assistive devices that empower individuals with hearing impairments. By fostering collaborative partnerships between technologists, educators, and community stakeholders, we can catalyze the co-creation of solutions that resonate with the unique needs and aspirations of the deaf community.

7. Conclusion

This paper presents a system for identifying ArSL using finger alphabet gestures. The system includes the design of three image groups for training and one group for testing, resulting in a dataset of 2800 examples representing 28 sign letters. A hand detection algorithm is applied to accurately detect hand shapes, achieving a detection accuracy of 98.64%. Hand shape features are extracted, and a database is generated, considering rotation, displacement, and resizing issues. Classification experiments are conducted using statistical classifiers (C4.5, Naïve Bayes, KNN) and an MLP neural network, comparing performance, training time, and prediction success. The KNN classifier demonstrates the best performance in implementing the proposed model, achieving a recognition accuracy rate exceeding 99.5%. The proposed system has potential applications in educational tools for deaf and dumb children and future Quran translation systems.

Acknowledgment

The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through Large Groups [grant number RGP.2/253/44].

Conflict of Interest

No potential conflict of interest was reported by the author(s).

References

[1] A. M. Ahmed, R. A. Alez, G. Tharwat, M. Taha, W. Ghribi, A. S. Badawy*, et al.*, "Towards the design of automatic translation system from Arabic Sign Language to Arabic text," in *2017 International Conference on Inventive Computing and Informatics (ICICI)*, 2017, pp. 325-330.

[2] M. A. Abdel-Fattah, "Arabic sign language: a perspective," *Journal of deaf studies and deaf education,* vol. 10, pp. 212-221, 2005.

[3] A. M. Ahmed, R. Abo Alez, G. Tharwat, M. Taha, B. Belgacem, and A. M. Al Moustafa, "Arabic sign language intelligent translator," *The Imaging Science Journal,* vol. 68, pp. 11-23, 2020.

[4] M. Mohandes, J. Liu, and M. Deriche, "A survey of image-based arabic sign language recognition," in *Systems, Signals & Devices (SSD), 2014 11th International Multi-Conference on*, 2014, pp. 1-4.

[5] X. Zabulis, H. Baltzakis, and A. A. Argyros, "Vision-Based Hand Gesture Recognition for Human-Computer Interaction," *The universal access handbook,* vol. 34, p. 30, 2009.

[6] M. Mohandes, "Arabic sign language recognition," in *In International conference of imaging science, systems, and technology*, Las Vegas, Nevada, USA, 2001, pp. 753-762.

[7] M. Mohandes, "Automatic translation of Arabic text to Arabic sign language," *AIML Journal,* vol. 6, pp. 15-19, 2006.

[8] M. Mohandes, M. Deriche, U. Johar, and S. Ilyas, "A signer-independent Arabic Sign Language recognition system using face detection, geometric features, and a Hidden Markov

Model," *Computers & Electrical Engineering,* vol. 38, pp. 422-433, 2012.

[9] M. Mohandes, S. Aliyu, and M. Deriche, "Arabic sign language recognition using the leap motion controller," in *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*, 2014, pp. 960-965.

[10] O. Al-Jarrah and A. Halawani, "Recognition of gestures in Arabic sign language using neuro-fuzzy systems," *Artificial Intelligence,* vol. 133, pp. 117-138, 2001.

[11] M. Al-Rousan and M. Hussain, "Automatic recognition of Arabic sign language finger spelling," *International Journal of Computers and Their Applications,* vol. 8, pp. 80-88, 2001.

[12] K. Assaleh and M. Al-Rousan, "Recognition of Arabic sign language alphabet using polynomial classifiers," *EURASIP Journal on Advances in Signal Processing,* vol. 2005, p. 507614, 2005.

[13] M. Maraqa and R. Abu-Zaiter, "Recognition of Arabic Sign Language (ArSL) using recurrent neural networks," in *Applications of Digital Information and Web Technologies, 2008. ICADIWT 2008. First International Conference on the*, 2008, pp. 478-481.

[14] M. Maraqa, F. Al-Zboun, M. Dhyabat, and R. A. Zitar, "Recognition of Arabic sign language (ArSL) using recurrent neural networks," *Journal of Intelligent Learning Systems and Applications,* vol. 4, pp. 41-52, 2012.

[15] N. El-Bendary, H. M. Zawbaa, M. S. Daoud, A. E. Hassanien, and K. Nakamatsu, "ArSLAT: Arabic sign language alphabets translator," in *Computer Information Systems and Industrial Management Applications (CISIM), 2010 International Conference on*, 2010, pp. 590-595.

[16] E. E. Hemayed and A. S. Hassanien, "Edge-based recognizer for Arabic sign language alphabet (ArS2V-Arabic sign to voice)," in *Computer Engineering Conference (ICENCO), 2010 International*, 2010, pp. 121-127.

[17] G. Latif, N. Mohammad, R. AlKhalaf, R. AlKhalaf, J. Alghazo, and M. Khan, "An Automatic Arabic Sign Language Recognition System based on Deep CNN: An Assistive System for the Deaf and Hard of Hearing," *International Journal of Computing and Digital Systems,* vol. 9, pp. 715-724, 2020.

[18] Y. Hou, S. Jia, X. Lun, Z. Hao, Y. Shi, Y. Li*, et al.*, "GCNs-net: a graph convolutional neural network approach for decoding time-resolved eeg motor imagery signals," *IEEE Transactions on Neural Networks and Learning Systems,* 2022.

[19] A. M. A. Moustafa, M. S. Mohd Rahim, M. M. Khattab, A. M. Zeki, S. S. Matter, A. M. Soliman*, et al.*, "Arabic Sign Language Recognition Systems: A Systematic Review," *Indian Journal of Computer Science and Engineering,* vol. 15, pp. 1-18, 2024.

[20] K.-l. Yan, "A comparison of sign language and spoken language rate of articulation and primacy effect," 1993.

[21] A. M. Ahmed, R. A. Alez, M. Taha, and G. Tharwat, "Propose a New Method for Extracting Hand using in the Arabic Sign Language Recognition (Arslr) System," *International Journal of Engineering Research & Technology (IJERT),* vol. 4, pp. 2278-0181, 2015.

[22] A. M. Ahmed, R. A. Alez, M. Taha, and G. Tharwat, "Automatic translation of Arabic sign to Arabic text (ATASAT) system," *Journal of Computer Science and Information Technology,* vol. 6, pp. 109-122, 2016.

[23] A. M. Ahmed, R. A. Alez, G. Tharwat, W. Ghribi, A. S. Badawy, S. B. Changalasetty*, et al.*, "Gestures Arabic Sign Language Conversion to Arabic Alphabets," in *2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, 2018, pp. 1-6.

[24] A. Ahmed, R. A. Alez, G. Tharwat, M. Taha, B. Belgacem, A. M. Al Moustafa*, et al.*, "Arabic sign language translator," *Journal of Computer Science,* vol. 15, pp. 1522-1537, 2019.

[25] M. M. Khattab, A. M. Zeki, A. A. Alwan, A. S. Badawy, and L. S. Thota, "Multi-frame super-resolution: A survey," in *2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, 2018, pp 1-8.

[26] M. M. Khattab, A. M. Zeki, A. A. Alwan, and A. S. Badawy, "Regularization-based multi-frame super-resolution: a systematic review," *Journal of King Saud University-Computer and Information Sciences,* vol. 32, pp. 755-762, 2020.

[27] M. M. Khattab, A. M. Zeki, A. A. Alwan, B. Bouallegue, S. S. Matter, and A. M. Ahmed, "Regularized multiframe Super-Resolution image reconstruction using linear and nonlinear filters," *Journal of Electrical and Computer Engineering,* vol. 2021, pp. 1-16, 2021.

[28] M. M. Khattab, A. M. Zeki, A. A. Alwan, B. Bouallegue, S. S. Matter, and A. M. Ahmed, "A Hybrid Regularization-Based Multi-Frame Super-Resolution Using Bayesian Framework," *Computer Systems Science and Engineering,* vol. 44, pp. 35-54, 2022.

[29] A. M. Soliman, M. M. Khattab, and A. M. Ahmed, "Recognize The Alphabet of Fingerspelling Using Statistical Classifiers to Facilitate Communication Between Hearing-

Impaired Persons and Others," *Journal of Qena Faculty of Arts,* vol. 31, pp. 1-30, 2022.

[30] H. A. AbdElghfar, A. M. Ahmed, A. A. Alani, H. M. AbdElaal, B. Bouallegue, M. M. Khattab*, et al.*, "A Model for Qur'anic Sign Language Recognition Based on Deep Learning Algorithms," *Journal of Sensors,* vol. 2023, 2023.

[31] H. A. AbdElghfar, A. M. Ahmed, A. A. Alani, H. M. AbdElaal, B. Bouallegue, M. M. Khattab*, et al.*, "QSLRS-CNN: Qur'anic sign language recognition system based on convolutional neural networks," *The Imaging Science Journal,* pp. 1-13, 2023.

[32] A. M. Soliman, M. M. Khattab, and A. M. Ahmed, "Arabic Sign Language Recognition System: Using an Image-Based hand Gesture Detection Method to help Deaf and Dump Children to Engage in Education," *Journal of Qena Faculty of Arts,* vol. 32, pp. 1-28, 2023.

[33] A. M. A. Moustafa, M. S. Mohd Rahim, B. Bouallegue, M. M. Khattab, A. M. Soliman, G. Tharwat*, et al.*, "Integrated Mediapipe with a CNN Model for Arabic Sign Language Recognition," *Journal of Electrical and Computer Engineering,* vol. 2023, 2023.

التعرف على الحروف في لغة الإشارة العربية:

من منظور التعلم الآلي

الملخص:

اكتسب التعرف على الأنماط في أنظمة التفاعل بين الإنسان والحاسب اهتمامًا كبيرًا في السنوات الأخيرة، وخاصة في مجال رؤية الحاسب وتطبيقات التعلم الآلي. واحدة من التطبيقات المهمة هي التعرف على إيماءات اليد المستخدمة في التواصل مع الأشخاص الصم، وتحديدًا في تحديد الحروف المتقطعة في سور القرآن الكريم. يقترح هذا البحث نموذجًا جديدًا للتعرف على لغة الإشارة العربية القائمة على الحروف الأبجدية. يتكون النظام من أربع مراحل: جمع البيانات، المعالجة المسبقة للبيانات، استخراج الميزات، والتصنيف. يتيح النموذج المقترح ثلاثة أنواع من مجموعات البيانات: الأيدي العارية أمام خلفية داكنة، الأيدي العارية أمام خلفية فاتحة، والأيدي التي ترتدي قفازات ذات لون داكن. تبدأ العملية بالتقاط صورة لإيماءة الأحرف الأبجدية، ثم يتم فصل اليد وعزل الخلفية. بعد ذلك يتم استخراج ميزات اليد بناءً على الطريقة المختارة. من حيث التصنيف، يتم استخدام تقنيات التعلم النظامي لتصنيف حروف الأبجدية العربية المكونة من ٢٨ حرفًا باستخدام ٩٢٤٠ صورة. يركز البحث على تصنيف الحروف الأبجدية الـ ١٤ التي تمثل فواتح سور القرآن في لغة الإشارة العربية. تظهر النتائج التجريبية أن النموذج المقترح الجديد يحقق دقة ممتازة تبلغ ٩٩.٥٪ باستخدام مصنف أقرب جار.

الكلمات المفتاحية: الحروف؛لغة؛ الإشارة؛ العربية؛التعلم الآلي.