Research Article

Arabic Sign Language Recognition System for Alphabets Using Machine Learning Techniques

Gamal Tharwat ¹, Abdelmoty M. Ahmed ^{1,2} and Belgacem Bouallegue ²

¹Department of Computers and Systems Engineering, Faculty of Engineering, Al-Azhar University, Cairo, Egypt

²Department of Computer Engineering, King Khalid University, Abha, Saudi Arabia

Correspondence should be addressed to Abdelmoty M. Ahmed; abd2005moty@yahoo.com

Received 14 August 2021; Revised 17 September 2021; Accepted 1 October 2021; Published 25 October 2021

Academic Editor: Yang Li

Copyright © 2021 Gamal Tharwat et al. 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.

In recent years, the role of pattern recognition in systems based on human computer interaction (HCI) has spread in terms of computer vision applications and machine learning, and one of the most important of these applications is to recognize the hand gestures used in dealing with deaf people, in particular to recognize the dashed letters in surahs of the Quran. In this paper, we suggest an Arabic Alphabet Sign Language Recognition System (AArSLRS) using the vision-based approach. The proposed system consists of four stages: the stage of data processing, preprocessing of data, feature extraction, and classification. The system deals with three types of datasets: data dealing with bare hands and a dark background, data dealing with bare hands, but with a light background, and data dealing with hands wearing dark colored gloves. AArSLRS begins with obtaining an image of the alphabet gestures, then revealing the hand from the image and isolating it from the background using one of the proposed methods, after which the hand features are extracted according to the selection method used to extract them. Regarding the classification process in this system, we have used supervised learning techniques for the classification of 28-letter Arabic alphabet using 9240 images. We focused on the classification for 14 alphabetic letters that represent the first Quran surahs in the Quranic sign language (QSL). AArSLRS achieved an accuracy of 99.5% for the K-Nearest Neighbor (KNN) classifier.

1. Introduction

Sign language (SL) develops naturally like the languages spoken within the deaf community, and each sign language has its own rules; in addition, the understanding of sign language outside the deaf community is almost non-existent or missing and thus communication is very difficult between deaf people and ordinary individuals. Some deaf children are born to ordinary parents, and thus a language gap exists within the family. Moreover, there is no standardized form of sign languages, which makes teaching a difficult challenge for a deaf person [1]. Deaf people also suffer from an understanding of the teachings of a true religion. Deaf Muslims have no ability, for example, to neither learn the Holy Quran nor understand its meanings.

Sign language is a combination of descriptive and nondescriptive signs as well as alphabet (fingerspelling) signs. However, in the case of Arabic Sign Language (ArSL),

there is no standardized language coordination, which makes learning or translating it a difficult challenge for Arab deaf communities [2].

Sign languages are not standard; they differ as a spoken language; even the Arabic sign language differs from one country to another and differs according to dialects. In general, deaf societies in Arab countries use many sign languages, such as Egyptian, Jordanian, Tunisian, and Gulf. Although these languages may participate in some signs, there is a gap due to the lack of education and the inability to communicate between the deaf and hearing people.

These challenges in the ArSL and the differences between it and the spoken language increase the need for machine translation between it and the spoken Arabic language, in addition to the ArSL recognition systems also. These systems can also help the deaf to integrate into different levels of education and enable them to access scientific knowledge through their mother tongue [3, 4].

In general, there are three levels of systems for recognizing the ArSL, namely, recognizing the gestures of the Arabic alphabet, recognizing isolated gestures (word level), and recognizing continuous gestures (the level of sentences). Figure 1 illustrates ways to identify the Arabic sign language and systems according to the types of datasets used.

In this paper, we will focus on alphabetic Arabic sign language recognition systems (AArSLRS), which will help the deaf and dumb community to overcome the challenges of communication as well as begin to learn the alphabets of the Arabic language, which is the language of the Holy Quran, and then they can identify through these systems the first Quranic surahs with the dashed letters, which are 29 surahs, by identifying the dashed letters, which are 14 letters, arranged in alphabetical order: “ح ر س ص ط ع ق ك ل م ن ه ا ي.” According to previous studies, research on this is still a difficult task for researchers as the automatic ArSL recognition systems lack performance accuracy and the field of recognition is very limited as well. In this article, we presented a technique for identifying the Arabic alphabet, focusing on the “Al-Fath of the Quranic surahs.” Our main contributions include the following:

- (i) Forming and designing datasets for Arabic sign language alphabet that consist of 8400 True (RGB) images for use as input and training data for classifiers in order to supervised learning.
- (ii) Developing an effective system for identifying fixed signs for the Arabic sign language alphabet to help deaf people learn about the first Quranic surahs “Al-Fath of the Quranic surahs” with the dashed letters.
- (iii) The integration between features that takes from different signers and then integrating their data in order to enhance the recognition accuracy; this process includes several challenges, such as synchronizing the rate of data capturing and data unification and normalization in order to produce a unified data vector.

An introduction to the proposed AArSLRS and the contributions of researchers in this paper is presented in the first section. The literature review in this field has been described in Section 2. In Section 3, the hearing impaired communication methods and the similarities between the fingers spelling and the alphabet in the Arabic language were presented; then the research methodology and design of the proposed system was discussed in Section 4. In Section 5, the results of computer simulation of the system are presented, and finally the conclusions are presented in Section 6.

2. Literature Reviews

There is a lot of research efforts in the field of developing recognition systems for sign languages around the world as well as in the Arab world with their scarcity, which sometimes focuses on systems based on vision or based on sensor gloves. We will focus in this research on systems based on vision, especially those dealing with the identification of alphabets in the Arabic sign language. In this section, we will

review the most important literature in this field in the last 10 years, with our focus on studies interested in the systems for identifying the alphabet of the Arabic sign language and showing the sizes of datasets used in these studies as well as focusing on algorithms and techniques used in this field.

Zabulisy et al. [5] proposed a vision-based hand gesture recognition system for Human-Computer Interaction. In vision-based approaches introduced to overcome these problems, Mohandes [6–9] introduced a prototype system to recognize the Arabic sign language based on Support Vector Machine (SVM) and also an automatic translation system to translate Arabic text to Arabic sign language.

AlJarrah and Halawani [10] developed a neuro-fuzzy system that deals with images of bare hand signs and achieved a recognition rate of 93.55%. In [11], Al-Rousan and Hussain built an adaptive neuro-fuzzy interference system for letter recognition. A colored glove was used to ease the process of segmenting the hands region. The recognition accuracy achieved was 95.5%.

Assaleh and Al-Rousan [12] used polynomial classifiers to perform the alphabet recognition. Second-order polynomial classifier per class is created for the training stage, and then the feature vectors are generated from the dataset. The error rates of both training set and testing set are 1.6% for training and 6.59% for testing data. In [13], Maraqa and Abu-Zaiter introduced an alphabet signs recognition system using Elman and fully recurrent networks for the recognition process. 900 samples of signs are provided by 2 signers representing 30 gestures. The Elman network achieved an accuracy rate of 89.66% and the fully recurrent network improved the accuracy to 95.11%.

El-Bendary et al. [14] developed an Arabic alphabet signs translator with an accuracy of up to 91.3%. The inputs of their system are features extracted from a video of signs and the output is text representing the letter. The features used are rotation, scale, and translation invariant. In the feature segmentation part, there is a small pause between the signs of each letter. Those pauses are used to detect the letter numbers and the related video frames. For each frame, the distances between three different black pixels are used to construct the feature vector. The signs of the alphabet are divided into three different categories. In the recognition stage, a Multilayer Perceptron (MLP) neural network and a Minimum Distance classifier (MDC) are used.

Hemayed and Hassanien [15] introduced an Arabic alphabet signs recognition system that converts recognized signs to speech. The system is much closer to the real-life application but unable to perform real-time recognition. It focuses on static and simple moving gestures. To convert the images to feature vectors, Principal Component Analysis (PCA) is applied. KNN is used in the classification stage.

In [16], Ahmed et al. proposed the development of a system for automatic translation of the Arabic sign language. As for the Arabic Text System (AArSLRS), this system relies on building two datasets for Arabic alphabet gestures. This system introduces a new manual detection technique that has been proposed that detects and extracts Arabic sign gestures from the image or video depending on the coverage

FIGURE 1: Arabic sign language recognition approaches.

of the hand and used different statistical classifiers and compare these results to get a better classification.

There are several recent research articles for ArSLR in literature reviews. We present a brief summary of the previous ArSLR systems in Table 1. In [3, 12, 17, 19, 20], some of the suggested systems and models are shown to identify isolated Arabic words. Some are for continuous identification systems, while others are for identifying the Arabic alphabet with a note that all of them have a limited dataset size. However, there are some exceptions where the dataset in one of the Arabic alphabet recognition systems reaches 54,049 samples as in [17] and in identifying gestures at the word level to 2323 samples as in [12].

It is noted that the highest accuracy of recognition was achieved in these systems in [19, 21, 22] with values up to 98%. These are unreliable values due to the small size of the dataset and therefore cannot be considered as a reference in measuring the quality of translation or recognizing the gestures of the Arabic sign language.

The only deep belief network system [23] was applied to only 200 samples, causing system performance degradation and reaching 85%. All systems in [3, 10, 12, 13, 19, 20, 24, 25] used traditional classifiers, such as the heuristics approach, KNN, neural network, repetitive neural network, mysterious neural compiler, and classifier supporting a vector, and measured by Euclidean distance and in some of these systems and models suggest a way to identify on the hand, according to the vision-based approach, depending on the wearing of colored gloves or specific-color gloves [21]. However, these models are not applicable in real life with restrictions of lighting and noise conditions that affect the quality and accuracy of gesture representation.

It may be useful to make an efficient and effective interface system; the human plays an important role. Graph convolutional neural networks, a novel deep learning framework, addressed in order to differentiate the four-class motor imaginary. To find the motor imaginary, four tasks are preferred with the prediction of highest accuracy [26]. This framework will be used in future work when creating a tool for deaf and dumb learning.

3. Hearing-Impaired Communication Methods

Sign languages are the usual communication media of deaf communities. Just like any spoken languages, they evolve naturally within deaf communities. Wherever these communities exist, and develop, they are deprived of the spoken language of the region. Arabic Sign Language, American Sign Language, and French Sign Language are distinctive sign languages used by corresponding communities in these regions.

There are many differences between sign and spoken languages. However, the main difference between them is the way the communicative pieces are made and perceived [27]. Other than sign languages, there are communication means used by deaf communities such as fingerspelling and cued speech.

The alphabet in the Arabic language represents the main phonemes that make up the word in the spoken language; likewise, the fingerspelling is the visual acoustics (optical phonemes) and constitutes the indicative symbols of the alphabet in sign language.

Fingerspelling is a visual representation of the alphabet and its signs using one hand or both hands together. Fingerspelling is part of the Arabic sign language and is used for various purposes. It can be used to represent words that have no equivalent sign, or to confirm or clarify or used by deaf people and teachers and translators to spell people's names. Or it is used in educational activities as a mental framework to teach the deaf child to read and write in Arabic. Or when teaching sign language to those interested in it, use some form of pronunciation formation or combine it to represent a guiding word [1].

Figure 1 shows how the shapes of the ArSL alphabet symbols are similar to the shapes of the letters in spoken Arabic. With descriptions of the similarities between them in writing as well as in the visual representation of the sign in the ArSL in Table 2.

One of the methods of communication based on phonetics and the characteristics of spoken languages is the cued speech [28], and this method makes use of moving lips and hand gestures to represent phonemes. As cued speech aims

TABLE 1: Survey of recent Arabic SLR systems.

Ref/year	Methods of collection dataset	Dataset size and type	Classifier	Accuracy in %
[10]/2001	Images of bare hand	Static alphabet-Arabic manual alphabets (60 examples for each sign where it was made use of 40 training samples and 20 for testing) = 1800 samples for 30 Arabic alphabet characters	Neuro-fuzzy classifier	93.55%
[11]/2001	Colored images with glove	Static alphabet-Arabic manual alphabets	ANFIS	95.5%
[12]/2005	Colorful glove with six colored areas in different colors	Static alphabet-recognition of ArSL alphabet, number of samples is 2323 for 30 different characters for 42 different gestures with a rate of 1625 for training and 698 for testing	Polynomial classifier	93.41% when the testing data is different from the training data and 98.41% when the training data is the same as the testing data
[13]/2008, 2012	Wearing colored gloves with colored markers for the fingers and wrist	Static alphabet (30 characters) number of images (900 for training, 300 for testing)	Elman and fully recurrent networks	The accuracy rate is 89.66% when using the Elman network and when using the fully RNN the accuracy rate has improved to 95.11%
[14]/2010	Free or bare hands	Videos containing sequences of characters-Arabic manual alphabet (15 characters)	(MDC) (MLP)	91.3% 83.7%
[15]/2010	Free or bare hands	Static alphabet and number (150 images)	Euclidean distance measure based on PCA	97.0%
[16]/2016	Free or bare hands and wearing colored gloves	Static alphabet (2 datasets) (dataset 1 by bare hands 700 samples for 28 characters and dataset 2 by wearing colored gloves = 700 samples for 28 characters)	C4.5 (J48), MLP, KNN (IBK), and Naïve-Bayesian classifiers	80.67%, 88.66%, 90.7%, 84.4% for dataset 1 and 89.5%, 94.11%, 97.5%, and 96.63% for dataset 2
[17, 18]/2019	Free or bare hands and wearing colored gloves	A dataset consisting of 54,049 gray scale images representing 32 alphabet letters for the Arabic sign language was collected by 40 signers	CNN	97.6%

to help deaf people understand spoken languages, overcome lip reading problems.

As cued speech helps the language of lip reading within the reach of the deaf and dumb people with hearing impairment, by replacing the invisible joints that participate in the production of sound (vocal cords, tongue, and jaw) with hand gestures, while maintaining the visible joints (lips). Reading different lips and hand gestures as an alternative to the audio system in the human body where the deaf person can distinguish between phonemes that have similar shapes or movement of lips. Where in this type of communication methods, audio information is exchanged through the movement and shape of the lips and the shape of the hand configuration and placed with the face.

Figure 2 shows the similarity between hand formation when representing the shapes of Arabic characters in sign language and the letters of the Arabic alphabet.

4. AArSLRS: Arabic Alphabet Sign Language Recognition System

In this section, we provide a general description of Arabic Alphabet Sign Language Recognition System (AArSLRS); the schematic structure of the system is illustrated in

Figure 3. The system functioning is divided into four main phases, namely:

- (1) Images or video (input data) acquisition
- (2) Images or video preprocessing
- (3) Features extraction
- (4) Classification and recognition of alphabet letters

The details of each stage will be discussed in the following sections.

4.1. Images or Video (Input Data) Acquisition. The first phase for AArSLRS is the capturing of the video using webcam where different alphabets were taken into consideration. 28 different alphabets were considered from 10 people.

In AArSLRS, to achieve high-accuracy gesture recognition in our system, 100 images were taken for each one alphabet. These images are included in the databases for use in the various stages of the system. The images captured by the signers are modified to obtain the clarity of the required images. Where gestures are captured using the vision-based approach, gestures that a number of signers perform by wearing glove in a dark color in different lighting environments with a light background, or without wearing a

TABLE 2: The similarity between forms the ArSL and Arabic language alphabets.

Alphabet	Similarity	Alphabet	Similarity	Alphabet	Similarity
ا	Matches with letter “ا” with thumbs up	ب	The index finger indicates a single point under the shape of “ب”	ت	The index and middle fingers indicate two points above the shape of “ت”
ث	The index, middle, and ring fingers indicate to three points above the “ث”	ج	Thumb position indicates a point inside the letter “ج”	ح	Thumb equal to the level of the other fingers as there is no point to the letter “ح”
خ	Thumb height above other fingers indicating a point above the letter “خ”	د	Matches the shape of the letter “د” with a sharp angle between the thumb and the index finger on the left	ذ	Matches the shape of the alphabet there is a sharp angle between the thumb and the index finger on the left and the presence of the middle on the top of the index finger indicating the presence of a point on the letter “ذ”
ر	Corresponds to the shape of the alphabet, where the thumb with the curvature of the index finger and the presence of the fingers closed, represents the shape of the letter “ر”	ز	Matches the shape of the alphabet, where the thumbs with the index finger curve and the presence of the middle on top of 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 the dots above the letter “ش”	ص	Approximate at the format with the letter “ص” with a note placed thumb	ض	Likeness as close when the thumb raised indication of a point above the 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 the fingers to indicate a point above the letter “ظ” and the index finger up	ع	Match with the alphabet letter “ع”
غ	Match with the shape of the alphabet letter and raise the thumb above the palm of the hand to indicate the presence of point above the letter “غ”	ف	Corresponds with the shape of the alphabet “ف” and the index finger position over the thumb to indicate a point above the letter	ق	Match with the shape of the alphabet “ق” with index and middle fingers above the thumb to indicate 2 points above letter
ك	Match with the shape of alphabet letter and put the thumb at the middle of the palm of the hand to indicate the presence of hamza ء for the character “ك”	ل	Match with the alphabet letter “ل”	م	No match with the alphabet letter “م”
ن	Match with alphabet letter “ن” with the absence of a point above the letter	هـ	Match with the alphabet letter “هـ”	و	Match with the alphabet letter “و”
ي	Match with the alphabet letter “ي”

FIGURE 2: Alphabet forms in ArSL (fingerspelling) [1].

glove (bare hand), with a dark background, the output of this stage is a color picture collection (RGB) representing the hand gestures corresponding to each letter of the ArSL. Table 3 shows the types of datasets created for use in training and testing.

In this part, we will describe a set of data that was created through the integrated webcam for the laptop as well as a smart camera for a mobile type Huawei that enabled us to take and store several consecutive pictures. Four full datasets were developed for ArSL images for the deaf. The dataset will

FIGURE 3: Main phases the Arabic Alphabet Sign Language recognition system (AArSLRS).

provide researchers and those interested in this field with an opportunity to explore and develop automated systems for the deaf and hard of hearing using machine learning techniques, human interaction with the computer, computer vision, and deep learning algorithms, where four datasets of images were developed and used to use as groups in training and testing the system after processing, and the following table shows a description of the four datasets. These four datasets designed entirely for ArSL will also contribute to the identification of the first Quranic surahs that start with dashed alphabetic characters, which helps the deaf to recognize the Quranic alphabet, and the five datasets will be made available to the public and to all researchers. The dataset consists of 9240 images of Arabic sign language gestures for the Arabic alphabet, and it consists of 28 letters collected from 10 locations in different age groups and with different sizes of the hand signed for the gesture. Various dimensions, angles, and complex and different backgrounds were present in the images that could be processed using digital image preprocessing techniques to remove noise, center, resizing and enhancement image, etc., from the process of preprocessing digital images.

4.2. Images Preprocessing and Hand Detection. The second stage in our system is the preprocessing that is the data preparation process. The datasets have been made robust as separate images were taken of more signers with different manual sizes and complex backgrounds and contain round samples and variable angles between 60 and -60 degrees to make the system robust. Color images are processed to improve image quality. The color image is converted to a gray scale image of 256 density levels and resizes to a

640×480 -pixel image. Some filtering methods can be applied to remove unwanted noise.

The preprocessing stage aims to convert data into a format that can be processed more easily and effectively as this stage consists of several methods of optimization and image enhancement, segmentation, and morphological filtering.

The preprocessing image operations are discussed in detail in Section 4.2.1. This stage depends on the steps of hand detection, where techniques of optimization and image enhancement, segmentation, and morphological filtering are applied to the color image in several ways to obtain the best sample to help us later extract the best features and then get the best accuracy.

At this phase, we will represent several steps preprocessing images of gestures alphabet of ArSL; these steps are shown in Figure 4.

4.2.1. First Step. Convert RGB Image to Gray Scale Image: One of the most important preprocessing images is converting it, or video-frame, from RGB to a gray-scale image. After a conversion, we have a gray-scale image. We can perform some operations to enhance image, segment, and perform morphology of the image to remove noise by using techniques such as digital image processing and computer vision, after which you can perform the stage of detection of important areas (hand shape) in the image.

The image of the real color sign is converted into a gray image in preparation for image processing and extracting important objects from the image by converting the three-dimensional image color matrix to the gray image matrix with one dimension indicating the color density in the gray

TABLE 3: Arabic sign language dataset of alphabet letters.

Dataset no.	Description and samples %
Dataset 1 (training set) 2800 samples	<p>Dataset no. 1 for the Arabic alphabet gestures for the deaf language, made with a bare hand, light background and different lights “true color” where 100 samples were taken for each letter of the Arabic alphabet</p>
Dataset 2 (training set) 2800 samples	<p>Dataset no. 2 for the Arabic alphabet gestures for the deaf language, made with a bare hand, dark background and different lights “true color” where 100 samples were taken for each letter of the Arabic alphabet</p>
Dataset 3 (training set) 2800 samples	<p>Dataset no. 3 for the Arabic alphabet gestures for the deaf language, made with a right hand and wearing gloves, white background and different lights “true color” where 50 samples were taken for each letter of the Arabic alphabet</p>
Dataset 4 (testing set) 840 samples	<p>Dataset no. 4 for the Arabic alphabet gestures for the deaf language, made with a bare hand “two hands” and wearing glove with different background, different angles, different lights, “true and gray image” where 50 samples were taken for each letter of the Arabic alphabet</p>

matrix saturation, hue and ignoring luminance by preserving the color brightness with special weights according to (R, G, B) of the color compounds in the real image, where the color compounds are weighted according to the formula used in the color scheme of change from RGB to gray scale.

$$I = 0.2989 * R + 0.5870 * G + 0.1140 * B. \quad (1)$$

The gray images were used in the research because most image processing deals with gray images such as filtering, and the expression of image information as a single vehicle indicates the intensity of the color in the image while maintaining the clarity of the image details is easier, in addition to the image cutting techniques used in the research dealing with gray images.

4.2.2. Second Step. Image Enhancement, Segmentation, and Morphological Filtering: In the second step, we do the segmentation process to convert the gray scale alphabet images to binary. Also, we adjusted the contrast of the image,

through noise removal, edge detection, and image processing structurally so as to be able to detect hand in the gray scale image. This step was carried out in two different methods; the image being produced has some noise. So, it is best to filter this noise using different filtering methods, where these methods will help us to get a smooth and full circumference of gesture and detected the edges of the hand in the gesture, which represents an alphabet letter.

The outputs obtained at this point are a binary image obtained by filtering the image with a gray scale and threshold (see Figure 3).

In this paper, we will use different techniques to implement image enhancements in order to segment it, perform morphological processing to remove noise, and detect important objects (hand shape) in the image represented by the following three methods.

4.2.3. Method 1. Threshold Method: In the first method, the image is segmented using threshold, where a fixed color threshold was chosen with a value of (50) and all pixels with

a density greater than the proposed threshold are given value (255) and represent the intensity of the white color and denote the background of the image, and pixels with a color density less than or equal. The proposed threshold value (0) represents the intensity of the black color and indicates the shape of the hand or any other element in the image with a dark color (converging with the color of the glove) depending on the color variation in the image (this method is appropriate for datasets 2 and 3).

The global threshold method is used to adjust the contrast of the image, so that the value of this threshold in a gradient of gray that is equal to or less than this threshold indicates the value of black and the value that is greater than the value of this threshold indicates white and all matrix values become either black = 0 or white = 1, and we have a black-and-white "binary" image, after which a new noise-free image is produced. Table 4 shows the result of this method.

We chose the threshold (50) on the basis that the signer wears a dark glove, because the dark color is affected by illumination in a lower percentage which reduces the noise in the image and facilitates the process of processing it; we can also control the threshold value by increasing or decreasing according to the color of the gloves used, to obtain more accurate results.

Gray scale level range for image is from the value 0, which represents the black color to the value 255, which represents the white color; when we studied the intensity of the captured images and noticed the appropriate values that enable us to separate the image into two different areas in density and resolution, we found that the value (50) is an experimentally appropriate value.

4.2.4. Method 2. Sobel Filter Method: In the second method, the image is segmented using Edge Detection technology using a Sobel filter method that detects the strong edges of the hand from pixels in which a high-density derivative is valued. It does not care about the weak edges connected with it. Also, Sobel filter method does not smooth the edges, and therefore we do not lose information from the shape of the hand and the zigzag of the ends of the fingers.

This method relies on the use of nonlinear filters to adjust image contrast and remove noise, Sobel technique to detect the edges of the hand in the image and 2D linear correlation to enhance edges.

We used median filter to adjust image contrast and remove noise, so choosing a mask 3*3 of adjacent pixels, where median filter orders values every three pixels adjacent in input image ascending by density, and choosing the middle pixel to be a pixel contrast in the output image.

This filter is characterized by eliminating the extreme values of pixels without affecting the clarity of the image.

Image segmentation based on the edge detection technique, using Sobel edge detector, is a simple nonlinear edge detection technique that is looking for places where the density changes quickly to put the value of 1 whereabouts edge and 0 otherwise. It applies in gray scale image and restores black and white image (BW) with the same size of original image. It detects strongly edges of hand and does not

care about the weak edges such as the canny method, as not to lose information from the shape of the hand and contours of ends of the fingers, as in the Laplacian method. Table 5 shows the comparison between the results of edge detection in several methods.

Use the two-dimensional (2D) convolution equation with a mask 3*3 it has deployed in order to fill in the gaps in the edges and fine-tune the image to reduce the number associated elements with improving the edges.

$$msk = \begin{bmatrix} 00000 \\ 01110 \\ 01110 \\ 01110 \\ 00000 \end{bmatrix}. \quad (2)$$

Table 6 shows the output of this method.

4.2.5. Hand Edge Detection. After detecting the hand area of the binary image in one of the two proposed methods, it is extracted using the addressing of the detected elements in the image and given labels in the form of serial numbers (0, 1, 2, ...) and the formation of a matrix the size of the binary matrix of the signal character image includes the labels of the areas connected to the standard 8 pixels adjacent where the designation (0) indicates the background and the label (1) denotes the first body, and so on for the rest of the designations. Indexes of pixels that make up an item are searched for and recorded within a beam. Then, the area of each area titled with the same name is calculated and the object with the largest area is shown in a separate image. It represents the shape of the hand only. We choose the largest area, given that the hand performs the signal in front of the computer camera and occupies the largest portion of the captured image.

Improving the edges is through the implementation of the expansion process of morphological objects in order to communicate with each other; this process was repeated several times in order to close the gaps and connect edges.

In this step, the hand is extracted from the image captured in the first stage after preprocessing in the second step; in Section 4.1, the results of this step will be presented with a comparison between the methods used and the detection ratios of the letter gestures for each of the datasets of the three datasets used in AArSLRS.

4.3. Features Extraction. In the previous phase, the hand was described and extracted according to one of the methods used previously; at this phase, we determine the best features in the gesture of a sign character and distinguish it from other gestures, which are then used in the training or testing process for the dataset.

Feature extraction phase is necessary because certain features have to be extracted so that they are unique for each gesture or sign following the determination that a sign is present.

FIGURE 4: Cycle of second stage in AArSLRS.

TABLE 4: The result of hand detection by global threshold method.

Original image (ص ك ه ي ع)	Grayscale image (ص ك ه ي ع) ^a	The image after cutting by using the threshold = (50) (ص ك ه ي ع)	Morphology method to close holes (10 times) (ص ك ه ي ع)

The proposed system depends on choosing a good description for the gesture and its features so that it is distinguished from any other sign. We use in this paper the description based on shape, since features are calculated by a contour-based or on the region-based methods; we have chosen a set of features to describe the hand gesture. Table 7 outlines the features used in AArSLRS.

We have implemented a feature extraction stage on the three training datasets where each group consists of 2,800 different examples of 28 alphabets, and we calculated the calculation of the proposed features for each example, where they were stored in a vector consisting of 15 values, and then each vector was stored for each sample in forming 2800 row number and 15 columns, the same thing we did for the three datasets, where the rows represent examples of alphabet gesture samples, and the columns are the features of each sample alphabet, and each dataset is stored as an Excel file with the extension (.csv).

4.4. Classification and Recognition of Alphabet Letters. In this phase, we will use two methods statistical classification and neural networks classification in order to know the alphabet fingers in our system.

To categorize a new gesture (hand shape) into one of the existing classes, we use statistical classification methods or neural networks to construct a classifier and train it under the supervision of training data (data sets produced in the previous step of the system).

The main objective of the AArSLRS is to create a mapping between the alphabet letters played by dumb, and

the signs of letters were stored in database signs that were generated in the previous phase.

This process will be designed using one of the supervised learning algorithms, to perform the task of classifying a new sign alphabet to the corresponding alphabetic letter in the Arabic language based on the data sets resulting from extracting the features of the hand shape, which represents the training data set.

In the implementation of the classification phase based on statistical classification algorithms (C4.5, Naïve-Bayesian, and KNN), multilayer perceptron (MLP) network algorithm was used and the results were compared in order to choose the best seed in terms of accuracy and speed.

Classification algorithms also were applied using a WEKA software tool to compare the results with the results of classification using the AArSLRS to the same dataset; the results are presented in Section 4.2.

5. Experimental Results of AArSLRS

In this part, we will present the results of the proposed system to recognize alphabet for Arabic sign language where the proposed system designed and implemented was AArSLRS; the system translates and recognizes gestures using one or both hands. The signers are not required to wear any glove-based sensors or use any devices to interact with the system. We will also display the results of the classification application using the WEKA software tool for classification data, compare them with the results of the

TABLE 5: Edge detection of letters (yaa and seen و س ي).

Original image (س يـ)	Sobel method (س يـ)	Roberts method (س يـ)	Canny method (س يـ)	Laplacian method (س يـ)

TABLE 6: The result of hand detection by Sobel filter method.

Original image (طـ ٥)	Grayscale image (طـ ٥)	The image after cutting by using the Sobel (طـ ٥)	Morphology method to close holes (10 times) (طـ ٥)

TABLE 7: The hand features used in AArSLRS.

Features	Feature description	Feature equation
F1	The ratio coordinates of the hand center of mass on the x -axis, to the width of around box by hand	$F1 = CX/x.width$
F2	The ratio coordinates of the hand center of mass on the y -axis, to the length of around box by hand	$F2 = CY/y.length$
F3	The ratio between the width of around box by hand to the length of around box by hand	$F3 = x.width/y.length$
F4	The ratio pixels of between the object (hand) area to the area of around box by hand	$F4 = area (object)/area of bounding box$
F5	The ratio between object (hand) perimeter to object (hand) area	$F5 = perimeter (object)/area (object)$
F6	The ratio between object major axis length to area of bounding box	$F6 = object major axis length/area of bounding box$
F7	The ratio between 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)	$F8 = X \text{ top-left}$ $F9 = X \text{ top-right}$ $F10 = X \text{ right-top}$ $F11 = Y \text{ right-top}$ $F12 = X \text{ right-bottom}$ $F13 = Y \text{ right-bottom}$ $F14 = Y \text{ left-bottom}$ $F15 = Y \text{ left-top}$

proposed system, and apply that to the three datasets. The graphical user interface (GUI) of the proposed system has been implemented by MATLAB, and Figure 5 shows the GUI system.

5.1. The Results of the Image Preprocessing and Hand Detection Phase. In this section, the results of implementing the hand detection step and extracting it from each alphabet gesture for each of the three datasets mentioned previously in the image acquisition phase used in this system will be presented and depending on the two methods used in this step.

Table 8 shows some of the letters of the alphabet and the shape of the hand after extracting it from the background in the pictures of the alphabet gestures after processing.

In the previous table, models for hand extraction and detection using the two selected methods are shown. Figures 6–8 show the results of comparison between the two methods for hand extraction and detection in each of the three datasets.

The accuracy of hand detection was evaluated in each method by calculating the number of examples in which the hand was successfully detected and isolated relative to the total number of examples according to equation 3 and ignoring the examples in which the hand was detected with the presence of noise, and considering them unacceptable because the noise affects negatively significantly; when calculating the properties of a shape, the hand gives wrong results in the classification phase. Table 9 below illustrates the proportion of the hand detection (Hand

FIGURE 5: Snapshot of the GUI for AArSLRS.

Detection Rate: HDR) in each set according to each method.

$$H\ DR = \frac{\text{Correct Samples without Noise}}{\text{Total Samples}} * 100. \quad (3)$$

5.2. Experimental Results of Classification Phase Using WEKA Tool. In this section, we will display the results of the recognition of sign language represented by the indicative fingering alphabet by using one of the learning algorithms supervised as mentioned above, relying on one of the three datasets resulting from the stage of extracting hand shape features which represent training data for the selected classifiers.

We will first show the results of the implementation of the classification and recognition phase using the Weka software tool, where a number of experiments were conducted as follows.

5.2.1. Experiment 1: Method of Estimate Error Classification. The WEKA software tool was used to apply classification based on statistical classification algorithms (C4.5, Naive Bayes, and KNN) and a Multilayer Perceptron (MLP) network learning algorithm and a comparison of results in order to choose the best classification in terms of accuracy and speed.

To estimate the error classification, the Holdout technique for sample selection or vote was employed to assess system performance evaluation.

In Table 10, we present the results of comparing the performance of the four classifiers that are used about the rate of letters gestures classified correctly (according to all datasets, using the Holdout method rates from 66% to 75%).

According to the previous table, we noticed that the KNN classifier is the highest in the classification accuracy, so we used it to classify a sample that represents the dashed letters of the earliest Quranic surahs, and they are arranged as follows: "ح ر س ص ط ع ق ك ل م ن ه ي ا". Table 11 illustrates the rate of recognition of each letter from these letters using KNN classifier and the number of samples that were recognized correctly.

We used a dataset containing 1400 instances for 14 characters used in the training process with 66% and testing with 34%, where 915 cases were classified correctly and 23 instances were classified incorrectly from 938 instances representing 66% of the training set with accuracy rate 97.548%.

5.2.2. Experiment 2: The Effect of the Number of Hidden Neurons on the Accuracy Rate Using MLP. The number of hidden neurons in the structure of neural networks plays an important role in the recognition and classification process. Table 12 shows the results of the performance of MLP network classifier according to the numbers of correctly classified instances, and the classification time for several experimental values for the number of neurons in the hidden layer.

We note from the results that the performance of the neural network increases with the increase in the number of neurons in the hidden layer as shown in Figure 9, but the classification time increases significantly.

5.2.3. Experiment 3: The Performance of the Classifiers Used in AArSLRS. The classification was performed, the performance of the classifiers used in the AArSLRS was compared in terms of the number of correctly classified instances and

TABLE 8: The hand detection in AArSLRS by two methods.

Hand Detection Case	Feature description				Feature equation			
Hand Detection
Hand Detection with noise
No Hand Detection

FIGURE 6: Hand detection ratio by two methods for dataset no. 1.

FIGURE 7: Hand detection ratio by two methods for dataset no. 2.

FIGURE 8: Hand detection ratio by two methods for dataset no. 3.

TABLE 9: 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

TABLE 10: Recognizing gestures letters accuracy by Holdout.

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

TABLE 11: Confusion matrix of 14-letter recognition rate by KNN classifier.

	أ	ح	ر	س	ص	ط	ع	ق	ك	ل	م	ن	ه	ي	<-- classified as	Class
59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$a = \text{alef}$	أ
0	75	0	0	0	0	0	0	0	0	0	0	0	0	0	$b = \text{haa}$	ح
0	0	68	0	0	0	0	0	0	0	0	0	0	0	0	$c = \text{ra}$	ر
0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	$d = \text{sen}$	س
0	0	0	0	72	0	0	0	0	0	0	0	0	0	0	$e = \text{sad}$	ص
0	0	0	0	0	69	0	0	0	0	0	0	0	0	0	$f = \text{tah}$	ط
0	0	0	0	0	0	62	0	0	0	0	0	0	0	0	$g = \text{ayn}$	ع
0	0	0	0	0	0	0	74	0	0	0	0	0	0	0	$h = \text{ka}$	ق
0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	$i = \text{kaf}$	ك
0	0	0	0	0	0	0	0	0	70	0	0	0	0	0	$j = \text{la}$	ل
0	0	0	0	0	0	0	0	0	0	61	0	0	0	0	$k = \text{mem}$	م
0	0	0	0	0	0	0	0	0	0	0	59	0	0	0	$l = \text{non}$	ن
0	0	0	0	0	0	0	0	0	0	0	0	62	0	0	$m = \text{he}$	ه
0	0	0	0	0	0	0	0	0	0	0	0	0	64	0	$n = \text{ya}$	ي

TABLE 12: The effect of neurons number in MLP network on the accuracy and time of classification.

No. of neurons	$H = 10$	$H = 15$	$H = 20$	$H = 28$
Time taken to build model (dataset 1)	10.32	16.34	21.03	29.36
Performance of classification (dataset 1)	86.14	86.99	86.35	89.01
Time taken to build model (dataset 2)	13.39	16.69	24.91	31.1
Performance of classification (dataset 2)	91.95	94.77	95.52	96.1
Time taken to build model (dataset 3)	13.76	18.46	26.43	32.62
Performance of classification (dataset 3)	97.65	98.05	98.41	98.52

FIGURE 9: Graph describing the role of number of hidden neurons in the performance of MLP.

TABLE 13: The effect of neurons number in MLP network on the accuracy and time of classification.

Classifier name	Cross-validation with fold = 10				No. of repetitions = 100	
	Correctly classified instances		Incorrectly classified instances		Root mean squared error	Kappa statistic
	Number (%)	Percentage	Number (%)	Percentage		
C4.5	97.7143	2736	2.2857	64	0.9763	0.0372
MLP	94.5357	2647	5.4643	153	0.9433	0.0556
KNN	99.7143	2792	0.2857	8	0.997	0.0143
Naïve-Bayesian	86.3571	2418	13.6429	382	0.8585	0.0931

FIGURE 10: A graph that describes the correct and incorrect classes for each classifier using in AARSLRS.

FIGURE 11: The relationship of Kappa statistic and the RMSE for each classifier using in AARSLRS.

FIGURE 12: KNN classifier screen in AARSLRS.

FIGURE 13: Decision tree (C4.5) classifier screen in AArSLRS.

TABLE 14: Recognizing gestures' letters accuracy by AArSLRS using MATLAB software.

K- neighbor value	Correlation	K = 1 and fold = 10		
		Cosine	Euclidean	Cityblock
The proportion of correct classified examples	0.91844	0.92019	0.97681	0.98249

the number of incorrectly classified instances for dataset 3. Table 13 shows these results.

The percentage of correctly classified instances is as high as possible with a KNN classifier and as little as possible with a classifier Naïve-Bayesian as shown in the diagram in Figure 10 that describes the correct and incorrect classes for each classifier.

The prediction scale (Kappa statistic) is successfully linked to the classification of the correct examples, which is the largest scale possible in a KNN classifier, and the least possible in the Naïve-Bayesian classifier; also, the Root Mean Squared Error (RMSE) is inversely proportional to the rudder of the classification, so it is the largest possible in a classifier Naïve-Bayesian, and it is the lowest possible in the KNN classifier, and Figure 11 shows the relationship of both the prediction scale (Kappa statistic) and the RMSE for each of the four classifiers.

5.2.4. Experiment 4: Results of Implementing Classification in AArSLRS Using MATLAB. The two most accurate classifiers were chosen among the four studied classifiers in order to test them within AArSLRS using MATLAB software as shown in Figures 12 and 13.

Table 14 shows the results of a classification by KNN algorithm for $k = 1$ in the AArSLRS using several options for calculating the distance, repeating the classification 100 times, and calculating the arithmetic mean of the accuracy by cross-validation method by fold = 10.

It is clear from the above table that AArSLRS achieves higher classification accuracy when using cityblock distance measurement while ready-made software such as WEKA is used for Euclidean distance measurement only. The possibility of changing the distance used in the AArSLRS proposed system can be considered an important feature that is not available in ready-made software.

The research achieved the desired goal by studying the analytical and operational system to build a system known as the sign language alphabet as part of the Arabic sign language. It is demonstrated that the method of image

segmentation using the global color threshold of dark glove color is an effective solution to overcome the problem of lighting with hand detection of the image. It has been shown that the method of electing samples of testing and training in works affects the accuracy of the classification and that the method of fixation (cross-validation) is better; the value of the transit group size is fold = 10 suitable in the search problem. The classification accuracy was calculated using neural networks and proved to be increasing as the number of hidden layer neurons increased for the research problem. It has been shown that the value of neighborhood k in the KNN classification algorithm affects the rating accuracy and the choice of $k = 1$ is the best in the search problem. It has been shown that the KNN algorithm compared to the C4.5 algorithm and the front-back propagation neural networks represented by the MLP network and the Naïve Bayes probabilistic probability is the best for the time, accuracy, and estimate error.

6. Conclusion

In this paper, the stages of AArSLRS were introduced to identify the Arabic sign language represented by the alphabet of the fingers, where three groups of images were designed for gestures of the sign letters, each dataset consisting of 2800 different examples representing 28 sign letters used for training and a fourth group of 840 examples of indicative letter used for the testing process. The hand shape was revealed by applying the proposed hand detection algorithm. According to two different methods of cutting and processing images, we obtained a hand detection accuracy of up to 98.64%, and the features of the hand shape in the resulting images were studied and extracted, and a database was generated in the size of 2800 different examples representing 28 indicative characters, each indicative character corresponding to an item represented by a beam of 15 values characterizing the shape of the hand taking into account the problems of rotation, displacement, and resizing of a single shape. We used the manually generated dataset to classify

hand gestures using the statistical classification C4.5, Naïve Bayes, KNN and the use of a classification using the MLP neural network. We compared the classification results in terms of the best performance and time required for examination and training and the measurement of prediction success, in addition to the Root Mean Squared Error (RMSE) for the classification of search examples. In terms of KNN performance to implement the AArSLRS, the classification results were compared using the proposed system with other classification software such as the WEKA software tool, the system was tested to recognize the earliest Quranic surahs that start with dashed letters and achieve an accuracy rate of 97.548%. Generally, the system achieved a recognition accuracy of a rate higher than 99.5%. The proposed AArSLRS can be used and developed in educational tools for deaf and dumb children, as well as in future translation systems for the meanings of the Holy Quran.

Data Availability

The data and MATLAB codes are available any time if needed from author Abdelmoty M. Ahmed, Email: abd2005moty@yahoo.com; amoate@kku.edu.sa.

Conflicts of Interest

No potential conflicts of interest were reported by the authors regarding the publication of this paper.

Authors' Contributions

Gamal Tharwat supervised the study and made considerable contributions to this research by critically reviewing the manuscript for significant intellectual content. Abdelmoty M. Ahmed designed the research plan, organized and ran the experiments, contributed to the presentation, analysis, and interpretation of the results, added, and reviewed genuine content where applicable. Belgacem Bouallegue made considerable contributions to this research by critically reviewing the literature review and the manuscript for significant intellectual content.

Acknowledgments

The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through General Research Project. The funding was provided by the Deanship of Scientific Research at King Khalid University through General Research Project (Grant no. GRP/332/42).

References

- [1] A. M. Ahmed, R. Abo Alez, G. Tharwat et al., "Towards the design of automatic translation system from Arabic sign language to Arabic text," in *Proceedings of the 2017 International Conference on Inventive Computing and Informatics (ICICI)*, pp. 325–330, IEEE, Coimbatore, India, November 2017, p.
- [2] M. A. Abdel-Fattah, "Arabic sign language: a perspective," *Journal of Deaf Studies and Deaf Education*, vol. 10, no. 2, pp. 212–221, 2005.
- [3] A. M. Ahmed, R. Abo Alez, G. Tharwat, M. Taha, B. Belgacem, and A. M. J. Al Moustafa, "Arabic sign language intelligent translator," *The Imaging Science Journal*, vol. 68, no. 1, pp. 11–23, 2020.
- [4] M. Mohandes, J. Liu, and M. Deriche, "A survey of image-based Arabic sign language recognition," in *Proceedings of the 2014 IEEE 11th International Multi-Conference on Systems, Signals & Devices (SSD14)*, pp. 1–4, IEEE, Barcelona, Spain, February 2014, p.
- [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, "Automatic translation of Arabic text to Arabic sign language," *AIML Journal*, vol. 6, no. 4, pp. 15–19, 2006.
- [7] M. Mohandes, S. Aliyu, and M. Deriche, "Arabic sign language recognition using the leap motion controller," in *Proceedings of the 2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*, pp. 960–965, IEEE, Istanbul, Turkey, June 2014, p.
- [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, no. 2, pp. 422–433, 2012, p.
- [9] M. Mohandes, "Arabic sign language recognition," in *Proceedings of the International Conference of Imaging Science, Systems, and Technology*, pp. 753–759p, Las Vegas, NV, USA, 2001.
- [10] O. Al-Jarrah and A. Halawani, "Recognition of gestures in Arabic sign language using neuro-fuzzy systems," *Artificial Intelligence*, vol. 133, no. 1-2, 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 Applied Signal Processing*, vol. 2005, no. 13, p. 507614, 2005.
- [13] M. Maraqa and R. Abu-Zaiter, "Recognition of Arabic Sign Language (ArSL) using recurrent neural networks," in *Proceedings of the 2008 First International Conference on the Applications of Digital Information and Web Technologies (ICADIWT)*, pp. 478–481, IEEE, Ostrava, Czech Republic, August 2008, p.
- [14] N. El-Bendary, H. M. Zawbaa, M. S. Dauod, A. E. Hassanien, and K. Nakamatsu, "Arslat: Arabic sign language alphabets translator," in *Proceedings of the 2010 International Conference on Computer Information Systems and Industrial Management Applications (CISIM)*, pp. 590–595, IEEE, Krakow, Poland, October 2010, p.
- [15] E. E. Hemayed and A. S. Hassanien, "Edge-based recognizer for Arabic sign language alphabet (ArS2V-Arabic sign to voice)," in *Proceedings of the 2010 International Computer Engineering Conference (ICENCO)*, pp. 121–127, IEEE, Cairo, Egypt, December 2010.
- [16] A. M. Ahmed, R. Abo Alez, G. Tharwat, and M. Taha, "Automatic translation of Arabic sign to Arabic text (ATA-SAT) system," *Journal of Computer Science and Information Technology*, vol. 6, pp. 109–122, 2016.

- [17] G. Latif, N. Mohammad, J. Alghazo, R. AlKhalaf, and R. AlKhalaf, "Arasl: Arabic alphabets sign language dataset," *Data in Brief*, vol. 23, Article ID 103777, 2019.
- [18] G. Latif, N. Mohammad, 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, no. 4, pp. 715–724, 2020.
- [19] M. Abdo, A. Mahmoud Hamdy, S. Abd El-Rahman Salem, and E. Mostafa Saad, "Arabic alphabet and numbers sign language recognition," *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 11, pp. 209–214, 2015.
- [20] S. M. Shohieeb, H. K. Elminir, and A. M. Riad, "Signsworld atlas; a benchmark Arabic sign language database," *Journal of King Saud University-Computer and Information Sciences*, vol. 27, no. 1, pp. 68–76, 2015.
- [21] T. Shanableh, K. Assaleh, and M. Al-Rousan, "Spatio-temporal feature-extraction techniques for isolated gesture recognition in Arabic sign language," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 3, pp. 641–650, 2007.
- [22] M. Mohandes and M. Deriche, "Image based Arabic sign language recognition," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 3, pp. 86–89, 2005.
- [23] A. Hasasneh and S. Taqatqa, "Towards Arabic alphabet and numbers sign language recognition," *Global Journal of Computer Science and Technology*, vol. XVII, pp. 15–23, 2017.
- [24] H. Luqman and S. A. Mahmoud, "Transform-based Arabic sign language recognition," *Procedia Computer Science*, vol. 117, pp. 2–9, 2017.
- [25] M. ElBadawy, A. S. Elons, H. A. Shedeed, and M. F. Tolba, "Arabic sign language recognition with 3d convolutional neural networks," in *Proceedings of the 2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS)*, pp. 66–71, IEEE, Cairo, Egypt, December 2017, p.
- [26] X. Lun, S. Jia, Y. Hou et al., "GCNs-net: a graph convolutional neural network approach for decoding time-resolved eeg motor imagery signals," vol. 14, pp. 1–10, 2020, <http://arxiv.org/abs/2006.08924>.
- [27] K.-L. Yan, "A comparison of sign language and spoken language rate of articulation and primacy effect," 1993.
- [28] T. E. Allen, M. Diane Clark, A. del Giudice et al., "Phonology and reading: a response to wang, trezek, luckner, and Paul," *American Annals of the Deaf*, vol. 154, no. 4, pp. 338–345, 2009.