computers

MDPI

Article

Empowering Communication: A Deep Learning Framework for Arabic Sign Language Recognition with an Attention Mechanism

R. S. Abdul Ameer, M. A. Ahmed *, Z. T. Al-Qaysi, M. M. Salih and Moceheb Lazam Shuwandy

Department of Computer Science, Faculty of Computers & Mathematics, Tikrit University, Tikrit 34001, Iraq; rafalsaleh@tu.edu.iq (R.S.A.A.); ziadoontareq@tu.edu.iq (Z.T.A.-Q.); mahmaher1989@gmail.com (M.M.S.); moceheb@tu.edu.iq (M.L.S.)

* Correspondence: mohamed.aktham@tu.edu.iq

Abstract: This article emphasises the urgent need for appropriate communication tools for communities of people who are deaf or hard-of-hearing, with a specific emphasis on Arabic Sign Language (ArSL). In this study, we use long short-term memory (LSTM) models in conjunction with MediaPipe to reduce the barriers to effective communication and social integration for deaf communities. The model design incorporates LSTM units and an attention mechanism to handle the input sequences of extracted keypoints from recorded gestures. The attention layer selectively directs its focus toward relevant segments of the input sequence, whereas the LSTM layer handles temporal relationships and encodes the sequential data. A comprehensive dataset comprised of fifty frequently used words and numbers in ArSL was collected for developing the recognition model. This dataset comprises many instances of gestures recorded by five volunteers. The results of the experiment support the effectiveness of the proposed approach, as the model achieved accuracies of more than 85% (individual volunteers) and 83% (combined data). The high level of precision emphasises the potential of artificial intelligence-powered translation software to improve effective communication for people with hearing impairments and to enable them to interact with the larger community more easily.

Keywords: deaf communication; sign language recognition; dynamic hand gestures; deep learning; LSTM networks; attention mechanism; MediaPipe framework; human–computer interaction; multimodal integration; assistive technology

check for updates

Citation: Abdul Ameer, R.S.; Ahmed, M.A.; Al-Qaysi, Z.T.; Salih, M.M.; Shuwandy, M.L. Empowering Communication: A Deep Learning Framework for Arabic Sign Language Recognition with an Attention Mechanism. *Computers* **2024**, *13*, 153. https://doi.org/10.3390/computers13060153

Academic Editors: Selene Tomassini and M. Ali Akber Dewan

Received: 26 April 2024
Revised: 7 June 2024
Accepted: 11 June 2024
Published: 19 June 2024

1. Introduction

People with hearing loss and speech impairments are deprived of effective contact with the rest of the community. According to the statistics of the International Federation of the Deaf and the World Health Organisation (WHO), more than 5% of people around the world are deaf and have severe difficulties communicating with those without hearing impairments, which means approximately 360 million people. Deaf individuals use another method to communicate instead of speech called sign language (SL) [1]. SL facilitates communication between the deaf community and people who are either deaf or nondisabled. SL is a visual communication system that encompasses both manual elements, such as hand gestures, and nonmanual elements, such as facial emotions and body movements [2]. SL is a complicated style of communication based mostly on hand gestures. These gestures are formed by different components, such as hand shape, hand motion, hand location, palm orientation, the movement of the lips, facial expressions, and points of contact between the hands or between the hands and other parts of the body, to express words, letters, and numbers.

Many sign languages exist in the deaf community, roughly one per country, which vary as much as spoken languages [3], e.g., Arabic Sign Language (ArSL), American Sign Language (ASL), British Sign Language (BSL), Australian Sign Language (Auslan), French Sign Language (LSF), Japanese Sign Language (JSL), Chinese Sign Language (CSL), German

Sign Language (DGS), Spanish Sign Language (LSE), Italian Sign Language (LIS), Brazilian Sign Language (LIBRAS), and Indian Sign Language, among others. Sign languages vary in lexicon, grammar, phonology, gesture form, and nonmanual elements, as do alphabets and words. Each language has its own unique features and regional variations, which reflect the diverse cultural and linguistic backgrounds of deaf communities worldwide. This diversity adds another difficulty, which is the lack of a unified sign language that serves universally as a vital means of communication and cultural expression for deaf individuals. Therefore, translating SL is indeed a necessary solution to bridge communication gaps between deaf and hearing individuals [4,5]. The development of automatic sign language translation systems reduces the reliance on human interpreters, lowers communication barriers, and promotes social inclusion in the deaf community. Hand gesture recognition is essential for automatic sign language translation systems. Researchers are increasingly interested in hand gesture recognition to solve communication challenges for deaf individuals, along with advances in gesture-controlled gadgets, gaming, and assistive technology [6].

Sign language recognition (SLR) systems focus on recognising and understanding sign language gestures and translating them into text or speech [7,8]. SLR systems typically involve artificial intelligence techniques to recognise and interpret the movements and forms of hands, fingers, and other relevant body parts used in SL. Several studies on sign language recognition (SLR) have attempted to bridge the communication gap between deaf and hearing individuals by eliminating the need for interpreters. However, sign language recognition systems have several obstacles, including a low accuracy, complex movements, a lack of large and full datasets containing various signals, and the models' inability to analyse them appropriately. Additionally, there are distinct indicators for each language [4,9,10].

This study proposes a deep learning (DL)-based model that leverages MediaPipe alongside RNN models to address the issues of dynamic sign language recognition. MediaPipe generates keypoints from hands and faces to detect position, form, and orientation, while LSTM models recognise dynamic gesture movements. Additionally, we introduce a new Arabic Sign Language dataset that focuses on dynamic gestures, as existing datasets predominantly feature static gestures in ArSL. In contrast, sensor-based solutions such as glove usage are expensive and impractical for everyday use due to power requirements and user annoyance. As a result, we abandoned this approach in favour of a more cost-effective approach involving the use of smartphone cameras to acquire data. The contributions of this study can be summarised as follows:

1. The DArSL50 dataset is a large-scale dataset comprised of 50 dynamic gestures in Arabic Sign Language (ArSL), including words and numbers, resulting in a total of 7500 video samples. This extensive dataset addresses the lack of sufficient data for dynamic gestures in ArSL and supports the development and evaluation of robust sign language recognition systems.
2. The proposed model leverages long short-term memory (LSTM) units with an attention mechanism combined with MediaPipe for keypoint extraction. This architecture effectively handles the temporal dynamics of gestures and focuses on relevant segments of input sequences.
3. The model's performance was evaluated in the following two scenarios: individual volunteer data and combined data from multiple volunteers. This dual evaluation approach ensures that the model is tested for its ability to generalise across different individuals and in different signing styles.
4. The proposed framework is validated for real-time performance.

The rest of this paper is organised as follows. Section 2 describes the methodology of the proposed ArSL recognition system and includes details about the DArSL 50 dataset. The experimental results are reported in Section 3, while an explanation of the results is presented in Section 4. Section 6 concludes the discussion and outlines future research directions.

The following two categories of sign language recognition systems can be distinguished according to the method used for data collection in the academic literature: sensor-based and vision-based [11], as shown in Figure 1.

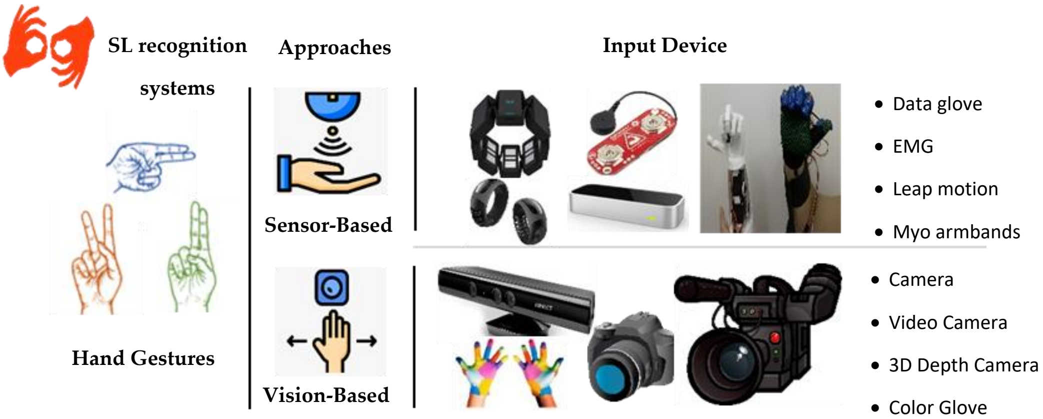


Figure 1. Sign language recognition approaches.

In the sensor-based method, sensors and equipment are used to collect the position, hand motion, wrist orientation, and velocity. Flex sensors, for instance, are used to measure finger movements. The inertial measurement unit (IMU) measures the acceleration of the fingers using a gyroscope and an accelerometer. The IMU is also used to detect wrist orientation. Wi-Fi and radar detect variations in the intensity of communications in the air using electromagnetic indicators. Electromyography (EMG) identifies finger mobility by measuring the electrical pulse in human muscles and then decreasing the biosignal. Other devices include haptic, mechanical, electromagnetic, ultrasonic, and flex sensors [12]. Sensor-based systems have an important advantage over vision-based systems, since gloves can rapidly communicate data to computers [13]. The device-based sensors (Microsoft Kinect sensor, Leap Motion Controller, and electronic gloves) can directly extract features without preprocessing, which means that the device-based sensors can minimise the time needed to prepare sign language datasets, data can be obtained directly, and a good accuracy rate can be achieved in comparison with vision-based devices [14]. Figure 2 demonstrates the primary phases of the SL gesture data collection and detection utilising the sensor-based system. The sensor-based approach has the issue of requiring the end-user to have a physical connection to the computer, making it unsuitable. Furthermore, it is expensive due to the use of sensitive gloves [13]. Despite the accuracy of the data that may be obtained from these devices, whether they wear gloves or are coupled to a computer, gadgets such as a Leap Motion or Microsoft Kinect device remain unpleasant [14].

Another option is the vision-based approach, which involves using a video camera to capture hand gestures. This gesture-detection solution combines appearance information with a 3D hand model. Key gesture capture technology in a vision-based technique was developed in Ref. [13]. Body markers such as colourful gloves, wristbands, and LED lights were used in this study, as well as active light projection systems that make use of the Kinect: Manufactured by Microsoft Corporation, Redmond, WA, USA. and Leap Motion Controller (LMC): Manufactured by Ultraleap Inc., San Francisco, CA, USA). A single camera might be employed with a smartphone camera, a webcam, or a video camera, as well as stereo cameras, which deliver rich information by using numerous monocular cameras. The primary benefit of employing a camera is that it removes the need for sensors in sensory gloves, lowering the system's manufacturing costs. Cameras are fairly inexpensive, and most laptops employ a high-specification camera due to the blurring effect of a webcam [13]. A simplified representation of the camera vision-based method for extracting and detecting hand movements is shown in Figure 3.

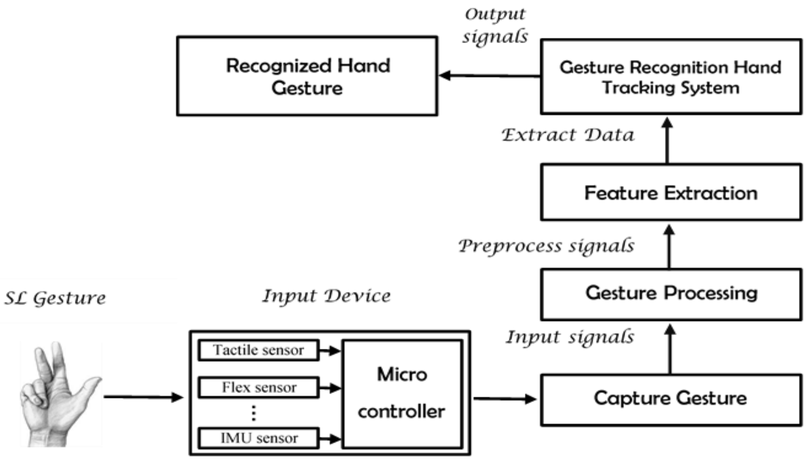


Figure 2. The main phases of recognising the SL gesture data using a sensor-based system [13].

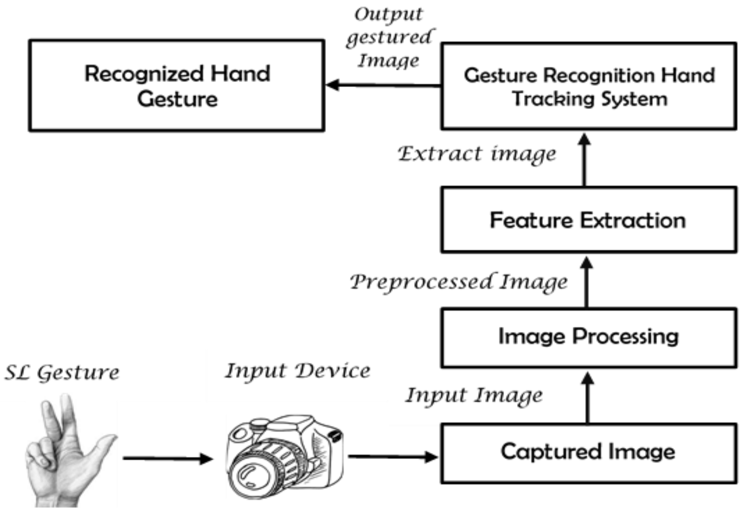


Figure 3. The procedure of vision-based sign language recognition [13].

In the literature, many SLR systems use traditional machine learning algorithms to classify the features of images to recognise SL gestures. In addition, the former uses traditional image segmentation algorithms to segment hand shapes from sign language images or the video frames of sign language video and then uses a machine-learning approach (such as SVM, HMM, or the k-NN algorithm). Using traditional machine learning algorithms has disadvantages related to handicraft features, which have a limited representational capability. It is difficult to extract representative semantic information from complex material, and step-by-step gesture recognition performs poorly in real-time. Other researchers have used deep neural networks to detect and recognise the gestures of SL. Deep neural network models such as CNNs, RNNs, GRUs, long short-term memory (LSTM), and bidirectional long short-term memory (LSTM) networks are used to address the issue of frame dependency in sign movement. These models employ an object-detection neural network to learn the video frame’s features, allowing it to find the hand while also classifying the movements. Compared to traditional image processing and machine learning algorithms, deep neural network-based target detection networks frequently achieve a higher accuracy and recognition speed, as well as better real-time performance, and have become the mainstream method of dynamic target detection. The advantage of deep learning is its ability to automatically learn data representations directly from raw inputs. Deep learning models can autonomously extract features and patterns from complex datasets without the need for manual feature engineering [15].

SLR studies can also be divided into static sign language recognition and dynamic sign language recognition. The former performs gesture recognition by judging the hand posture, and it does not contain dynamic information. The latter contains hand movements and performs gesture recognition based on the video sequence, which is essentially a classification problem. Dynamic sign language recognition is much more difficult to implement than static sign language recognition, but it is more meaningful and valuable.

The following presents a review of SLR studies, including methods and datasets. In Ref. [16], a recognition system was utilised as a communication tool between those who are hearing-challenged and others who are not. This work describes the first automatic Arabic Sign Language (ArSL) recognition system using hidden Markov models (HMMs). A vast number of samples were utilised to identify 30 isolated terms from the standard Arabic Sign Language. The recognition accuracy of the system was between 90.6 and 98.1%. In Ref. [17], ArSL was based on the hidden Markov model (HMM). They collected a large dataset to detect 20 isolated phrases from the genuine recordings of deaf persons in various clothing and skin hues, and they obtained a recognition rate of approximately 82.22%. In Ref. [18], the authors presented an ArSL recognition system. The scope of this study includes the identification of static and dynamic word gestures. This study provides an innovative approach for dealing with posture fluctuations in 3D object identification. This approach generates picture features using a pulse-coupled neural network (PCNN) from two separate viewing angles. The proposed approach achieved a 96% recognition accuracy. Ref. [19] provided an automated visual SLRS that converted solitary Arabic word signals to text. The proposed system consisted of the following four basic stages: hand segmentation, tracking, feature extraction, and classification. A dataset of 30 isolated words used in the everyday school lives of hearing-challenged students was created to evaluate the proposed method, with 83% of the words having varied occlusion conditions. The experimental findings showed that the proposed system had a 97% identification rate in the signer-independent mode. Ref. [20] presented a framework for the field of Arabic Sign Language recognition. A feature extractor with deep behaviour was utilised to address the tiny intricacies of Arabic Sign Language. A 3D convolutional neural network (CNN) was utilised to detect 25 motions from the Arabic Sign Language vocabulary. The recognition system was used to obtain data from depth maps using two cameras. The system obtained a 98% accuracy for the observed data, but the for fresh data, the average accuracy was 85%. The results might be enhanced by including more data from various signers. In Ref. [21], a computational mechanism was described that allowed an intelligent translator to recognise the separate dynamic motions of ArSL. The authors utilised ArSL's 100-sign vocabulary and 1500 video clips to represent these signs. These signs included static signs such as alphabets, numbers ranging from 1 to 10, and dynamic words. Experiments were carried out on our own ArSL dataset, and the matching between ArSL and Arabic text was evaluated using Euclidian distance. The suggested way to automatically find and translate single dynamic ArSL gestures was tested and found to work well and correctly. The test findings revealed that the proposed system can detect signs with a 95.8% accuracy. In Ref. [4], the authors generated a video-based Arabic Sign Language dataset with 20 signs generated by 72 signers and suggested a deep learning architecture based on CNN and RNN models. The authors separated the data preprocessing into three stages. In the first stage, the proportions of each frame decreased to reach a lower total complexity. In the second stage, they sent the result to a code that subtracted every two consecutive frames to determine the motion between them. Finally, in the third stage, the attributes of each class were merged to produce 30 frames, with each unified frame combining 3 frames. The goal of stage three was to decrease the duplication while not losing any information. The primary idea behind the proposed architecture was to train two distinct CNNs independently for feature extraction, then concatenate the output into a single vector and transmit it to an RNN for classification. The proposed model scored 98% and 92% on the validation and testing subsets of the specified dataset, respectively. Furthermore, they attained promising accuracies of 93.40% and 98.80% on the top one and top five rankings of the UFC-101 dataset, respectively. The study by Ref. [22] provides a computer application for translating Iraqi Sign Language into Arabic (text). The translation process began with the capture of videos to create the dataset (41 words). The proposed system then employed a convolutional neural network (CNN) to categorise the sign language based on its attributes to infer the meaning of the signs. The proposed system's section that translates the sign language into Arabic text had an accuracy rate of 99% for the sign words.

Research on Arabic Sign Language recognition lacks common datasets available for researchers. Despite the publication of two volumes of "A Unified Arabic Sign Language Dictionary" in 2008, researchers in this field continue to face a lack of large-scale datasets. As such, each researcher needed to create a sufficiently large dataset to develop the ArSL recognition systems. Therefore, this study endeavoured to create a comprehensive dataset that was explicitly tailored for Arabic Sign Language recognition. Subsequently, this dataset serves as the foundation for the development of an accurate Arabic Sign Language recognition system capable of recognising the dynamic gestures inherent in ArSL.

2. Materials and Methods

The suggested system for recognising dynamic hand gestures uses keypoints that have been extracted. It is a neural network model that is constructed for learning from one sequence to another. Figure 4 depicts the primary phases of the proposed framework for recognising the dynamic gestures of Arab Sign Language.

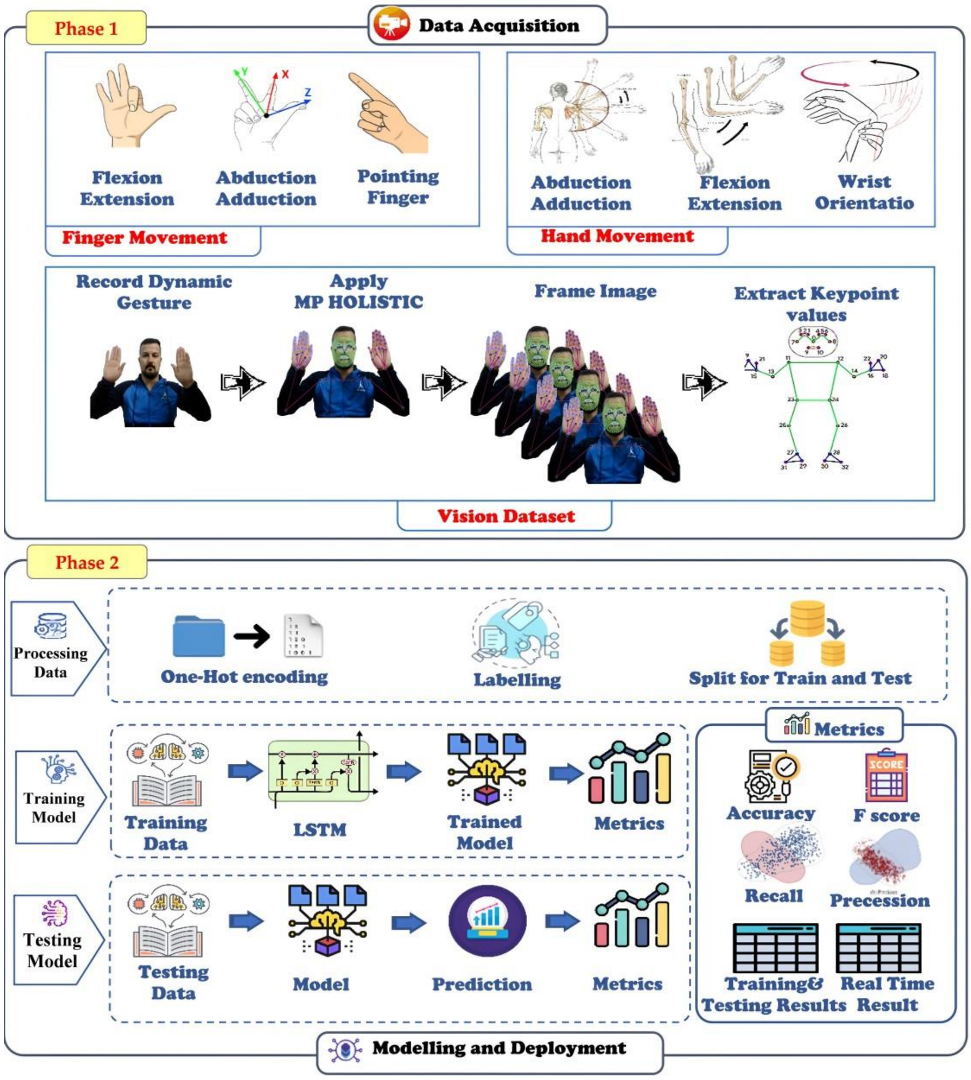


Figure 4. The proposed sign language recognition framework for dynamic Arabic gestures.

The model architecture incorporates both long short-term memory (LSTM) units and an attention mechanism. The model received a series of extracted keypoints from recorded gestures that indicate hand spatial configurations in a frame. The LSTM layer is responsible for processing the input sequence, identifying the temporal dependencies, and encoding the sequential information in its output sequence. The LSTM output sequence was also improved with an attention layer that allows the model to focus on different parts of the input sequence based on how relevant they are to the task at hand. The incorporation of this attention mechanism enhanced the ability of the model to recognise significant temporal patterns and spatial configurations within the sequences of gestures. Ultimately, the output layer generates a probability distribution over the potential classes of hand gestures, enabling the model to categorise the input sequences into predetermined gesture categories.

Anaconda Navigator (Anaconda3) and the free Jupyter Notebook Version 6.4.3 environment service were used to create the framework software package for the selected models. By utilising the Open-Source Computer Vision Library (OpenCV) Version 4.5.3, a specialised photo and video processing library that enables a wide range of tasks, including image analysis, facial recognition, and the identification of sign language gestures, along with the Mediapipe library, which extracts information from multimedia and which is the main tool for tracking motion and video analysis, the MP-holistic model was put into action along with some drawing functions. A dataset was recorded and gathered in which the volunteer represented all of the gestures by recording 30 videos of 30 frames each. The next stage was the conversion of frameworks from BGR to RGB colour coordination, because MediaPipe prefers RGB and Open CV coordination prefers BGR colour coordination. For the application of an activated model in each framework and the extraction of keypoint values, we created subvolumes under a major folder to store video clips for each class, where a separate folder was created for each class and each video under this volume, and these data were the data used to train the learning model to classify these classes. The dataset was collected and recorded using a webcam, and analysed using the MediaPipe model. The volunteer had to follow the criteria, which will be mentioned later, and then perform them. The key values discovered from the multimedia library's total model were extracted and stored for training. Then, we started the pretreatment phase, which involved labelling each class. A label was used to convert the correct name into a binary representation. For example, in our search for 50 classes of (0–49), Class 1 will become [0, 1, 0] and Class 2 will become [1, 0, 0]. A sequential neural network model comprising LSTM layers and fully linked layers was constructed for the classification. The training approach involved utilising data and the "Adam" algorithm to optimise the weight parameters, while the "categorical_crossentropy" function was employed to compute the loss during training. The term "categorical accuracy" refers to the correctness of the categorisation and served as a metric for evaluating the model's performance. The subsequent step involved saving the model, which could then be employed to recover the model and make predictions or to conduct the training. The last phase involved evaluating and using the confusion matrix, accuracy, and classification energy.

2.1. Dataset

In recent years, there has been tremendous development in the field of deep learning algorithms in artificial intelligence (AI). The success of AI applications depends on the quality and quantity of training and testing data. To improve AI systems, vast datasets must be collected and used. As far as we are aware, there is a lack of sufficient datasets for dynamic signals in Arabic Sign Language, which impedes the progress of recognition systems. Thus, it is crucial to create a large-scale dataset for dynamic signals in Arabic Sign Language. Accordingly, we created a DArSL50 dataset with a wide range of Arabic Sign Language dynamic motions. The DArSL50 dataset is comprised of 50 Arabic gestures representing 44 words and 6 digits. Each gesture was recorded by five participants. We selected signs from two dictionaries, " قاموس لغة الاشارة للاطفال الصم" (Sign Language Dictionary for Deaf Children) and "قاموس الاشاري العربي" (The Arabic Sign Language Dictionary for the Deaf).

Figure 5 displays a segment of the sign language database, which includes 50 dynamic signals in the Arabic Sign Language (ArSL) database. Five volunteers recorded each sign, with each participant performing each sign 30 times. Hence, the aggregate number of videos reached 7500, which was calculated by multiplying 50 by 5 and then by 30. The Video Capture function in OpenCV enabled the collection of data, which were then saved in NumPy format for further analysis.

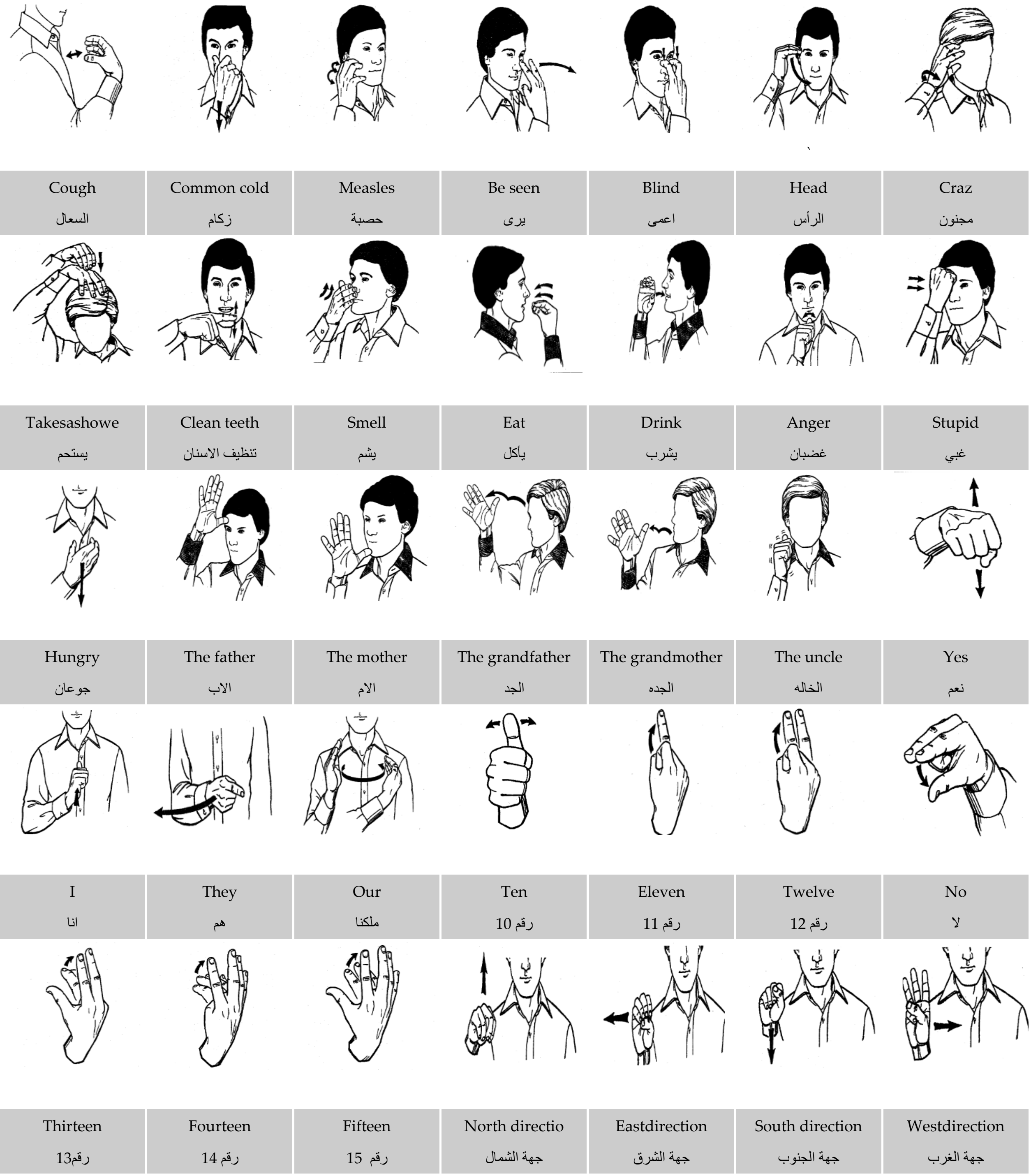


Figure 5. Images from the ArSL Words and Numbers dataset, which includes the lexicon for sign language for children that are deaf and the Arabic Sign Language Dictionary.

To collect the dataset, a series of processes were carried out. Initially, a collaboration was formed with the Deaf Centre, ensuring access to resources and specialised knowledge in Arabic Sign Language. Two dictionaries were examined to understand the signs. This study focused on 50 frequently used words and numbers, with a particular emphasis on those that may be expressed using only the right hand for the sake of simplicity. A group of volunteers was enlisted to imitate the signs, with each sign being replicated 30 times to capture variations. Data collection involved recording videos using a laptop camera, while the OpenCV program analysed the video clips by extracting important characteristics and preparing the data for additional analysis. This meticulous approach resulted in the creation of a complete and representative dataset for the study of ArSL signs. Volunteers of diverse demographics participated without limitations, ensuring inclusivity and diversity within the dataset. In addition, it is important to guarantee that the volunteer's body and all of their movements fit within the camera frame. A consistent and unchanging background setting should be ensured, with a particular emphasis on capturing volunteers' hands and faces. A robust camera tripod was used to generate crisp and dependable video recordings. In addition, it is advisable to establish the duration and frame count of the clip before recording, and to strive for a resolution of 640 × 480 or greater to achieve the best possible quality.

2.2. Feature Extraction Using MediaPipe

Google created MediaPipe, an open-source framework that allows developers to build multimodal (video, audio, and time-series data) cross-platform applied ML pipelines. MediaPipe contains a wide range of human body identification and tracking algorithms that were trained using Google's massive and diverse dataset. As the skeleton of the nodes and edges, or landmarks, they track keypoints on different parts of the body. All of the coordinated points are three-dimensionally normalised. Models built by Google developers using TensorFlow lite facilitate the flow of information that is easily adaptable and modifiable via graphs [23]. Sign language is based on hand gestures and stance estimation, yet the recognition of dynamic gestures and faces presents several challenges as a result of the continual movement. The challenges involved recognising the hands and establishing their form and orientation. MediaPipe was used to address these issues. It extracts the keypoints for the three dimensions of X, Y, and Z for both hands and estimates the postures for each frame. The pose estimation approach was used to forecast and track the hand's position relative to the body. The output of the MediaPipe architecture was a list of keypoints for hand and posture estimation. MediaPipe extracted 21 keypoints for each hand [24], as shown in Figure 6. The keypoints were determined in three dimensions, X, Y, and Z, for each hand. Therefore, the number of extracted keypoints for the hands is determined as follows [25]:

$$\text{keypoints in hand} \times \text{Three dimensions} \times \text{No. of hands} = (21 \times 3 \times 2) = 126 \text{ keypoints.}$$

For the pose estimation, MediaPipe extracted 33 keypoints [26], as shown in Figure 7. They were calculated in three dimensions (X, Y, and Z), in addition to the visibility. The visibility value indicates whether a point is visible or concealed (occluded by another body component) in a frame. Thus, the total number of keypoints extracted from the pose estimate is computed as follows [27]:

$$\text{keypoints in pose} \times (\text{Three dimensions} + \text{Visibility}) = (33 \times (3 + 1)) = 132 \text{ keypoints.}$$

For the face, MediaPipe extracted 468 keypoints [28], as shown in Figure 8. Lines linking landmarks define the contours around the face, eyes, lips, and brows, while dots symbolise the 468 landmarks. They were computed in three dimensions (X, Y, and Z). Thus, the number of retrieved keypoints from the face is computed as follows:

$$\text{Key points in face} \times \text{Three dimensions} = (468 \times 3) = 1404 \text{ keypoints.}$$

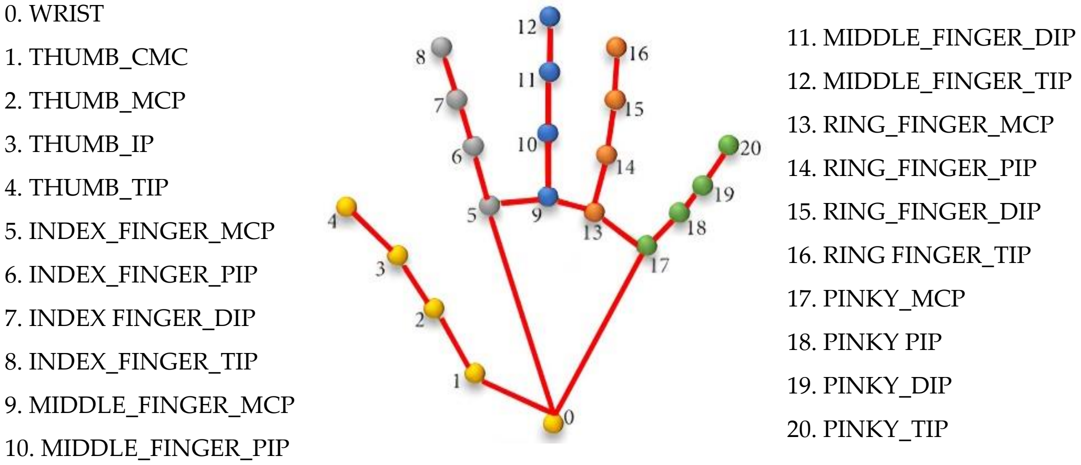


Figure 6. A total of 21 keypoints for the hand.

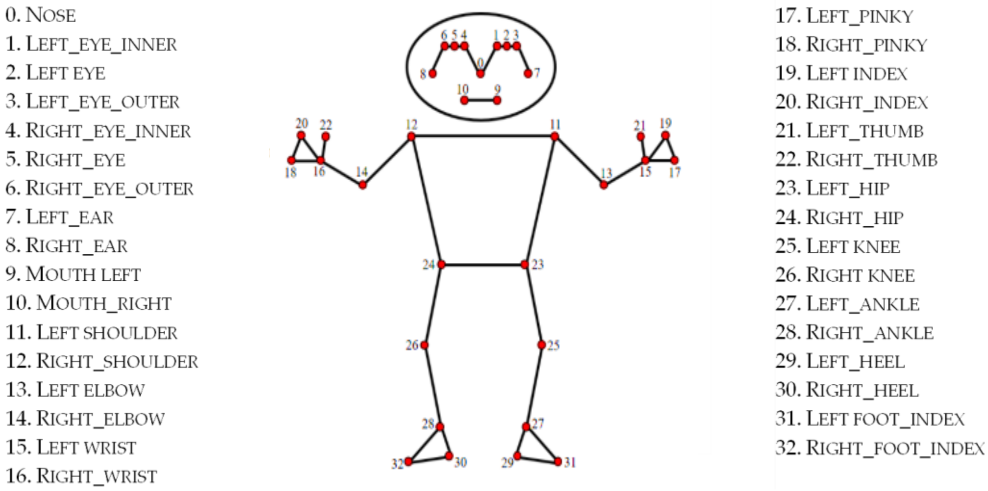


Figure 7. A total of 33 keypoints for the pose.

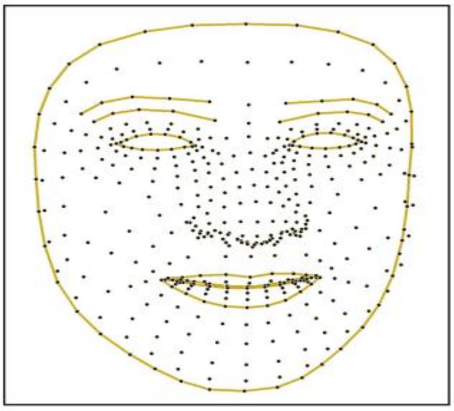


Figure 8. A total of 468 keypoints for the face.

The total number of keypoints for each frame was determined by summing the number of keypoints in the hands, the pose, and the face. This calculation resulted in a total of 1662 keypoints. Figure 9 displays the keypoints retrieved from a sample of frames.

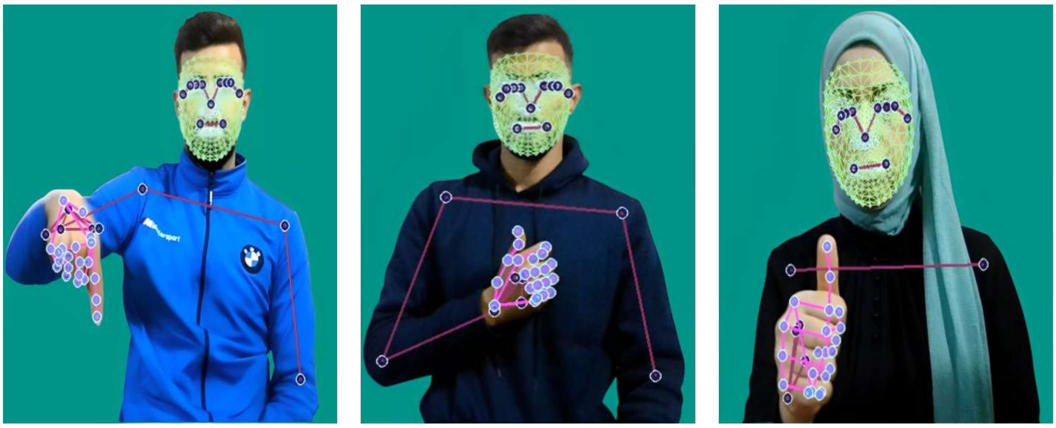


Figure 9. Keypoints that were extracted from a sample of frames.

2.3. Model

To process the dynamic gestures, data were represented as a series of frames, with each frame containing a collection of values representing the features of the hand posture in that frame. A recurrent neural network, specifically long short-term memory (LSTM), was used to process the resulting set of frames. LSTM is a well-known tool for encoding time series by extracting latent sign language expressions [29]. The model used in this study combines LSTM units with an attention mechanism. The model structure comprises the following three primary layers: an LSTM layer, an attention mechanism layer, and an output layer. The LSTM layer consists of 64 units, which contribute the most parameters to the model because of its recurring nature and the related parameters for each unit. The attention mechanism layer introduces a limited number of parameters, consisting of 10 units that govern the attention weights. The output layer, which is responsible for predicting the hand gesture classes, has a set of parameters that are dictated by the size of the context vector generated by the attention mechanism and the number of classes that need to be predicted. In total, the model consists of 89,771 parameters, with the LSTM layer accounting for the largest proportion. This architecture was specifically designed to efficiently handle sequential data, exploit temporal relationships, and dynamically prioritise essential sections of the input sequence, ultimately facilitating precise hand motion detection. The choice of the optimal parameter was pivotal for building these layers. Table 1 displays the utilised model parameters. During the use of the model, the parameters of each layer can be modified by picking values from Table 1 in preparation for the training phase.

Table 1. Model layer parameters.

Parameters	Value
Model	LSTM
Number of Nodes	64
Input Shape	(timesteps, 1662)
Attention Units	10
Activation	'softmax'
Optimiser	'adam'
Epochs	40

The choice of 64 hidden units and the specific activation function (ReLU) was based on preliminary experiments and established practices in similar research domains. An LSTM model with 64 hidden nodes was used to balance the model complexity and computational performance. We wanted a model that could learn complex data patterns without overfitting, which may occur with large networks. Experiments showed that 10 attention units offered enough attentional concentration without too much of a processing burden. We used 'SoftMax' for the activation function because it is common for classification tasks, especially multiclass problems. The LSTM model underwent training for a total of 40 epochs, with early stopping based on validation loss to prevent overfitting. The models' inputs include the sequence length and total number of keypoints. The sequence length is the number of frames contained in each clip. The total number of keypoints was 1662. At this point, the model is ready to accept the dataset and begin the training phase using the sequence of keypoints collected. Thus, the sign movement was examined and a hand gesture label could be used. As a result, DArSL-50 could be accurately detected.

2.4. Experiments

This research collected data from five participants, resulting in two separate scenarios. The first scenario involved creating the model by using the data from each volunteer separately. In the second scenario, the data gathered from the volunteers were combined, and then the suggested model was implemented. In Scenario 1, the dataset comprised data from five volunteers, with each volunteer contributing 1500 data points. For the training set, 1125 data points were selected, representing 75% of the total data, ensuring a comprehensive representation of the variability within the dataset. The remaining 375 data points were allocated to the testing set, representing 25% of the total data. This subset was reserved for evaluating the performance and generalizability of the trained models, as shown in Table 2.

Table 2. Data size, training set, and test set for each volunteer.

Number of Volunteers	Dataset Size	Train	Test	Size Test
One volunteer	1500	1125	375	0.25

In Scenario 2, four datasets were generated by combining the volunteer data. Data-I was composed of data collected from two volunteers, resulting in 3000 data points. Subsequently, Data-II, Data-III, and Data-V were formed by merging the data from three, four, and five volunteers, resulting in dataset sizes of 4500, 6000, and 7500 data points, respectively. To evaluate the proposed model, the dataset was partitioned into training and testing sets using a split ratio of 75–25 respectively. As a result, the training set consisted of 3375, 4500, and 5625 data points, while the testing set contained 1125, 1500, and 1875 data points for the datasets with three, four, and five volunteers, respectively, as shown in Table 3.

Table 3. Data size, training set, and test set for Scenario 2.

Dataset	Number of Volunteers	Dataset Size	Train Size	Test Size
Data-I	Two volunteers	3000	2250	750
Data-II	Three volunteers	4500	3375	1125
Data-III	Four volunteers	6000	4500	1500
Data-IV	Five volunteers	7500	5625	1875

The objective of integrating the dataset with data from numerous individuals was to improve the reliability and applicability of the trained models across a wide variety of signers and signing styles. By integrating the data from several individuals, the models were enhanced to effectively manage variances in gestures and signing styles, resulting in enhanced performance in real-world applications. This training and testing technique allowed for a thorough assessment and validation of the models, ensuring their dependability and efficacy in different settings and populations.

2.5. Evaluation Metrics

Evaluation metrics, such as the accuracy, precision, recall, and F1 score, are commonly used to evaluate the performance of classification models. These metrics provide crucial information about how well the model is doing and where it may require improvement.

Accuracy is the most commonly used simple metric for classification. It represents the ratio of the number of correctly classified predictions to the total number of predictions. A high level of accuracy indicates that the model is making correct predictions overall. The accuracy was calculated using Equation (1), as follows:

$$Accuracy \ = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

Precision measures the proportion of true positive predictions among all positive predictions.

Interpretation: A high precision indicates that, when the model predicts a positive class, it is likely to be correct. The precision is calculated using Equation (2), as follows:

$$Precision \ = \frac{TP}{TP + FP} \tag{2}$$

Recall measures the proportion of true positive predictions among all actual positive instances.

Interpretation: A high recall indicates that the model can identify most of the positive instances. The recall is calculated using Equation (3), as follows:

$$Recall \ = \frac{TP}{TP + FN} \tag{3}$$

The F1 score is the harmonic mean of the precision and recall, providing a balanced measure between the two metrics. The F1 score considers both the precision and recall, making it suitable for imbalanced datasets where one class dominates. The F1 score is calculated using Equation (4), as follows:

$$F1 - Score \ = \frac{(2 \times \ Precision \ \times \ Recall \ )}{(\ Precision \ + \ Recall \ )} \tag{4}$$

where:

The number of true positives (*TPs*) is the number of positive class samples correctly classified by a model. True negatives (*TNs*) are the number of negative class samples correctly classified by a model. False positives (*FPs*) are the number of negative class samples that were predicted (incorrectly) to be of the positive class by the model. False negatives (*FNs*) are the number of positive class samples that were predicted (incorrectly) to be of the negative class by the model. The classification report provides the accuracy, recall, and F1 score for each class, as well as the overall metrics. The assessment measures were used to determine how well the trained models performed on the testing datasets. This showed how well, accurately, and consistently they could recognise Arabic Sign Language gestures.

3. Results

The studies were carried out on a PC with an Intel(R) Core (TM) i7-10750H CPU operating at a base frequency of 2.60 GHz, which has 12 cores and 16,384 MB of RAM. The framework was developed using the Python programming language. The source code for this study may be accessed at the following URL: https://drive.google.com/file/d/1FcXudNQqXb_IzehsdMWb0tSBplcq-8LJ/view?usp=sharing (accessed on 10 June 2024). The dataset was gathered by a team of five volunteers, including a total of 50 distinct categories. Every participant captured recordings for the dataset consisting of 50 classes, and the outcomes were examined using the DArSL50 dataset. The DArSL50 dataset was divided randomly, with 75% used for training and 25% used for testing in the experiment. The performance criteria, such as the accuracy, precision, recall, and F1 score, were assessed under different situations to evaluate the functioning of the suggested system. In the first scenario, we evaluated the classification model with a dataset that included five participants' recordings; each participant provided 1500 data points. A training set was created from 1125 data points (representing 75% of the total), and a testing set was created from 375 data points (representing 25% of the total). Table 4 indicates the performance metrics obtained for each volunteer in Scenario 1.

Table 4. Results for Scenario 1.

Volunteer	Accuracy	Precision	Recall	F1 Score
Volunteer1	0.82	0.84	0.81	0.80
Volunteer2	0.83	0.83	0.83	0.82
Volunteer3	0.85	0.86	0.85	0.83
Volunteer4	0.83	0.84	0.85	0.83
Volunteer5	0.84	0.84	0.83	0.82

The data presented in Table 4 indicate that the third volunteer achieved the highest accuracy, approximately 85%, while the first volunteer achieved the lowest accuracy, approximately 82%. Nevertheless, the dataset's accuracy ratio for all volunteers was highly similar, indicating a highly effective discrimination mechanism for each individual. The results of Scenario 1 provide valuable insights into the model's efficacy in categorising hand movements using the given dataset. Through the evaluation of parameters such as accuracy, precision, recall, and the F1 score, we can determine the model's ability to generalise across various volunteers and accurately recognise gestures. The model's high accuracy, precision, recall, and F1 score demonstrate its effectiveness in recognising hand gestures from varied recordings. This indicates that the model is resilient and generalisable across multiple volunteers and signing styles. Table 5 shows the findings of Scenario 2, which included experiments to recognise dynamic hand gestures for four datasets. These datasets represent a combination of volunteer data.

Table 5. The proposed framework results for Scenario 2.

Dataset	Accuracy	Precision	Recall	F1 Score
Data-I	0.83	0.83	0.83	0.82
Data-II	0.82	0.83	0.83	0.82
Data-III	0.80	0.82	0.80	0.80
Data-IV	0.80	0.82	0.80	0.80

The results presented in Table 5 indicate that the highest level of accuracy, reaching 83%, was achieved by Data-I, which represents the combined data of two participants. However, Data-III and Data-IV achieved the minimum accuracy, which was approximately 80%. The accuracy of the four experiments varied between 83% and 80%, which is near and relevant in terms of the precision and recall. The F1 score, a metric that combines precision and recall using the harmonic mean, provides a well-balanced evaluation of the models' overall performance, with scores ranging from 0.82 to 0.80. By analysing Table 5, it is clear that the best accuracy ever achieved after the merger of volunteers is almost very close to the accuracy of the merger of the five volunteers, which suggests that the system is good with discrimination and has a strong impact, depending on the multiple people and the magnitude of the dataset. Overall, the models had good precision and recall scores, indicating that they could make accurate predictions and successfully detect positive events. These results show that the trained models are effective at recognising Arabic Sign Language. Compared to Data-IV, Table 6 shows the performance metrics (precision, recall, and the F1 score) for recognising 50 different types of ArSL gestures. Every row represents a particular class, and the metrics indicate the model's performance in accurately differentiating between gestures of that class.

Table 6. Results for the scenarios with classification reports for each class of Scenario 5.

Class Label	Dynamic Arabic Gesture	English Meaning	Precision	Recall	F1 Score
0	سعال	Cough	0.71	0.75	0.73
1	زكام	Common cold	0.84	0.82	0.83
2	حصبة	Measles	0.88	0.93	0.90
3	يرى	Be seen	0.73	0.84	0.78
4	اعمى	Blind	0.83	0.67	0.74
5	الرأس	Head	0.97	0.92	0.94
6	يستحم	Takes a shower	1.00	0.97	0.99
7	فرشاة اسنان	Cleaning teeth	0.79	0.98	0.87
8	يشم	Smell	0.86	0.65	0.68
9	يأكل	Eat	0.69	0.81	0.75
10	يشرب	Drink	0.76	0.77	0.76
11	غضبان	Anger	0.97	0.92	0.95
12	جوعان	Hungry	0.97	0.85	0.90
13	ابو	The father	0.97	0.88	0.92
14	ام	The mother	0.90	0.72	0.80
15	جد	The grandfather	0.86	1.00	0.92
16	جدة	The grandmother	0.91	0.94	0.93
17	خالة	The uncle	0.96	0.72	0.83
18	انا	I	0.87	0.84	0.85
19	هم	They	0.92	0.77	0.84
20	ملكنا	Our	0.92	0.87	0.89
21	رقم10	Ten number	0.71	0.75	0.73
22	رقم11	Eleven number	0.81	0.65	0.65
23	رقم12	Twelve number	0.65	0.65	0.67
24	رقم13	Thirteen number	0.65	0.67	0.65
25	رقم14	Fourteen number	0.65	0.68	0.65

Table 6. *Cont.*

Class Label	Dynamic Arabic Gesture	English Meaning	Precision	Recall	F1 Score
26	رقم15	Fifteen number	0.65	0.65	0.66
27	جهة الشمال	North direction	0.68	0.94	0.79
28	جهة الشرق	East direction	0.94	0.72	0.82
29	جهة الجنوب	South direction	0.84	0.65	0.71
30	جهة الغرب	West direction	0.79	0.89	0.84
31	نعم	Yes	0.68	0.83	0.75
32	لا	No	0.74	0.89	0.81
33	يفهم	Understand	0.67	0.88	0.74
34	غبي	Stupid	0.83	0.88	0.85
35	مجنون	Crazy	0.90	0.74	0.81
36	مع السلامة	Goodbye	0.94	1.00	0.97
37	مهم	Important	0.79	0.76	0.77
38	نمو	To grow	1.00	0.98	0.99
39	اسكت(صمت)	Shut up (silence)	0.88	0.88	0.88
40	حالا(الان)	Immediately (now)	0.71	0.89	0.79
41	حزين(دمعة)	Sad (tear)	0.97	0.91	0.94
42	حضور	Presence (coming)	0.95	0.97	0.96
43	ذهاب	To go	1.00	0.91	0.95
44	اهلا	Hello (congratulations)	0.96	0.69	0.80
45	توقف	To stop	0.88	0.94	0.91
46	امانة	Honesty	0.92	0.70	0.71
47	اعطى	To give	0.74	0.94	0.83
48	اهلك(قضى على)	To destroy	0.69	0.88	0.77
49	تخلص من	To get rid of	0.91	0.91	0.91

Table 6 presents a comprehensive analysis of the performance metrics of the model for each class in the classification report. Some classes demonstrate exceptional performance, as seen by their high precision, recall, and F1 score levels. For instance, the classes "Takes a shower", "Our", "The grandfather", and "Understand" exhibit high scores in all measures, indicating that the model accurately recognises these actions. However, specific classes exhibit disparities in performance indicators. For example, the "Blind" class exhibits relatively high precision but lower recall and F1 scores, suggesting that the model can accurately detect certain instances of this gesture but may fail to detect certain actual occurrences.

Classes such as "Common cold", "Measles", and "Stupid" consistently and effectively display strong recognition abilities across all parameters, indicating their robustness in gesture recognition. Conversely, classes such as "North direction", "East direction", and "To grow" display different performance metrics, with higher precision but lower recall values. This suggests that the model might have difficulty in accurately identifying all occurrences of these gestures. Based on the categorisation report results, we discovered that classes 11, 12, 13, and 14 (equivalent to classes 23, 24, 25, and 26, respectively) performed relatively poorly compared to the other classes. This is due to the nature of the movement in these classes, where the distinction between individual movements may be unclear. For example, the movement could be a slight hand gesture with no substantial variations in motion, or the difference between one movement and another may not be obvious enough, making classification more difficult for these classes. High values of accuracy, precision, recall, and the F1 score indicate successful model performance, while lower values may signify areas for improvement in the model's predictive capabilities.

To evaluate the system performance in real-time sign language detection, measurements were made concerning the reading error rate at the first stage. Algorithm 1 presents the approach used to measure the system performance metrics. Each letter was tested individually with five participants, and 40 iterations were applied to each letter to determine the frequency of the recognition. Consequently, the performance of the proposed system can be assessed by calculating the recognition accuracy of each gesture, followed by the total accuracy of the entire system, as shown in Algorithm 1. Errors in the results may be categorised as either "misclassification" (incorrect recognition) or "gesture not recognised" (not detection). The accuracy and error rates are determined using the equations provided below:

$$\text{Accuracy\%} = \frac{\text{detected right}}{\text{Num.of itration}} \times 100 \tag{5}$$

$$\text{Wrong recognise\%} = \frac{\text{detected wrong}}{\text{Num.of itration}} \times 100 \tag{6}$$

$$\text{Not detected\%} = \frac{\text{not detected}}{\text{Num.of itration}} \times 100 \tag{7}$$

Algorithm 1 Inference procedures for real-time sign language detection.

```
Input: D—new data                    {perform dynamic gesture}
Output: M real-time sign language detection model performance metrics
Initialise I $\leftarrow$ 0, D $\leftarrow$ 0, Z $\leftarrow$ 0, E $\leftarrow$ 0 {Initialise counts}
while I < 40 do
    gesture $\leftarrow$ CaptureGesture()       {Capture the gesture}
    if RecogniseGesture(gesture) == DesiredGesture then
        D $\leftarrow$ D + 1                       {Increment correct detection count}
        Display("Gesture is found")
    else
        if gesture == "No detection", then
            Z $\leftarrow$ Z + 1                   {Increment no detection count}
            Display("Gesture is not recognised")
        else
            E $\leftarrow$ E + 1                   {Increment misclassification count}
            Display("Misclassification: Wrong recognition")
        end if
    end if
    I $\leftarrow$ I + 1                           {Increment iteration count}
end while
Display("Total Correct Detections: " + D)
Display("Total Misclassifications: " + E)
Display("Total Nondetections: " + Z)
Display("Total Iterations: " + I)
```

The real-time results are summarised in Table 7, which shows the accuracy, error of incorrect recognition, and error of not detecting each sign. The real-time performance analysis of dynamic Arabic gesture recognition reveals high accuracy for gestures such as “مع السلامة” (Goodbye) and “فرشاة اسنان” (Cleaning teeth), indicating the model’s proficiency with distinct patterns. However, lower accuracy and higher error rates in gestures such as “اعمى” (Blind) and “ يشم” (Smell) suggest difficulties in distinguishing these gestures, highlighting areas for improvement.

Table 7. The Real-Time Performance Result.

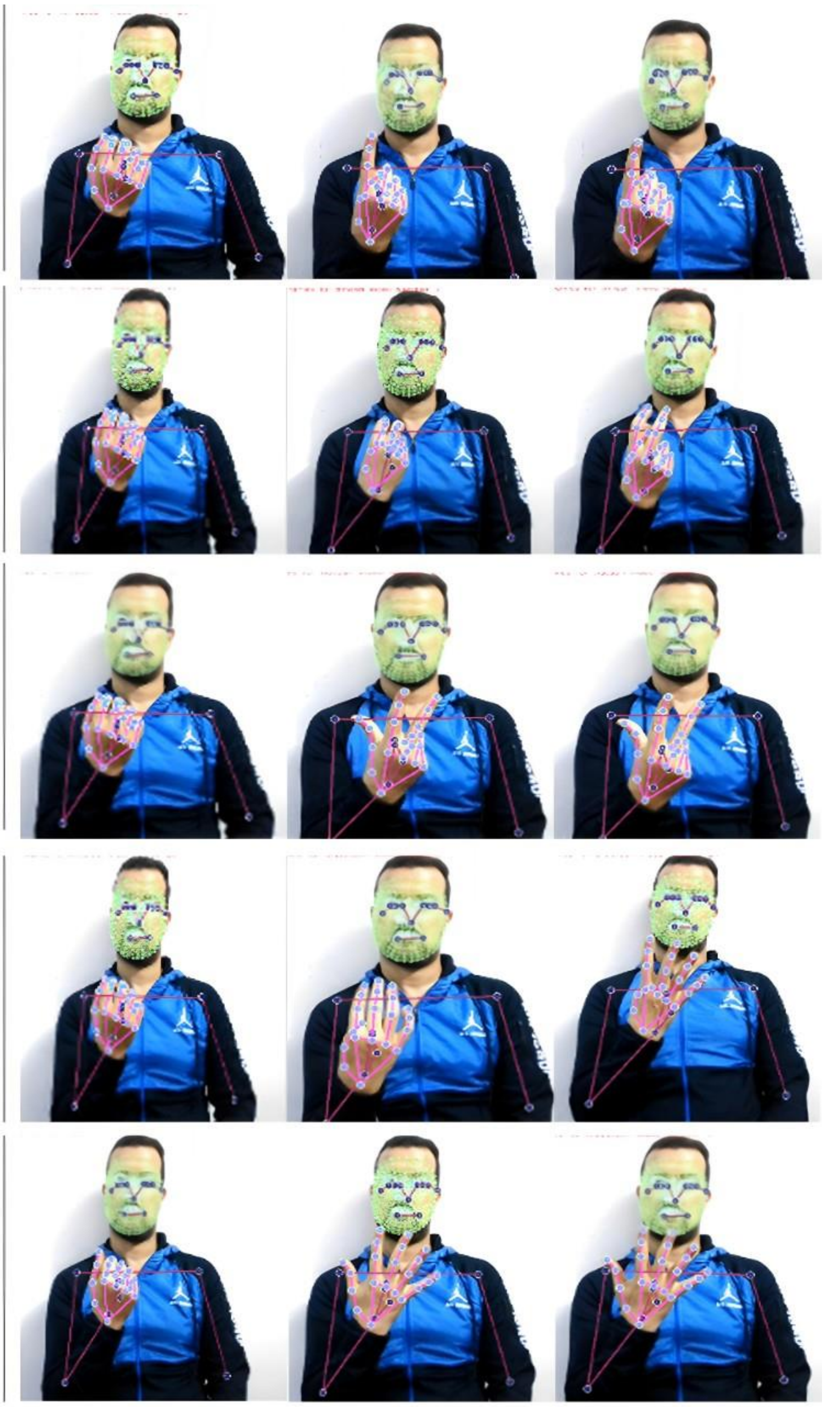
Class Label	Dynamic Arabic Gesture	English Meaning	Accuracy (%)	Err of Wrong Detected (%)	Err of Not Detected (%)
0	سعال	Cough	75	17	8
1	زكام	Common cold	82	11	7
2	حصبة	Measles	93	7	0
3	يرى	Be seen	84	0	16
4	اعمى	Blind	72	12	16
5	الرأس	Head	92	2	6
6	يستحم	Takes a shower	97	0	3
7	فرشاة اسنان	Cleaning teeth	98	0	2
8	يشم	Smell	75	5	20
9	يأكل	Eat	81	10	9
10	يشرب	Drink	77	16	7
11	غضبان	Anger	92	3	5
12	جوعان	Hungry	85	3	12
13	ابو	The father	88	3	9
14	ام	The mother	72	10	18
15	جد	The grandfather	100	0	0
16	جدة	The grandmother	94	0	6
17	خالة	The uncle	72	8	20
18	انا	I	84	13	3
19	هم	They	77	8	15
20	ملكنا	Our	87	0	13
21	رقم10	Ten number	75	20	5
22	رقم11	Eleven number	65	22	13

Table 7. *Cont.*

Class Label	Dynamic Arabic Gesture	English Meaning	Accuracy (%)	Err of Wrong Detected (%)	Err of Not Detected (%)
23	رقم12	Twelve number	65	25	10
24	رقم13	Thirteen number	67	26	7
25	رقم14	Fourteen number	68	28	4
26	رقم15	Fifteen number	65	15	20
27	جهة الشمال	North direction	94	3	3
28	جهة الشرق	East direction	72	6	22
29	جهة الجنوب	South direction	75	6	19
30	جهة الغرب	West direction	89	2	9
31	نعم	Yes	83	8	9
32	لا	No	89	4	7
33	يفهم	Understand	88	0	12
34	غبي	Stupid	88	3	9
35	مجنون	Crazy	74	0	26
36	مع السلامة	Goodbye	100	0	0
37	مهم	Important	76	11	13
38	نمو	To grow	98	0	2
39	اسكت(صمت)	Shut up (silence)	88	6	6
40	حالا(الان)	Immediately (now)	89	0	11
41	حزين(دمعة)	Sad (tear)	91	3	6
42	حضور	Presence (coming)	97	0	3
43	ذهاب	To go	91	0	9
44	اهلا	Hello (congratulations)	85	11	4

The results presented in Table 7 evaluate the real-time recognition proficiency of dynamic Arabic gestures, which achieved an overall accuracy rate of 83.5%. The accuracy of dynamic Arabic gestures indicates a generally high performance for many gestures, such as "مع السلامة" (Goodbye) and "فرشاة اسنان" (Cleaning teeth), with a 100% and 98% accuracy, respectively, and minimal errors. This reflects the model's effectiveness in recognising distinct gestures. Conversely, gestures such as " يشم" (Smell) and "اعمى" (Blind) achieved a moderate accuracy, with significant errors not detected (20% and 16%). Numeric gestures, particularly " رقم11"

(Eleven number) and “ رقم12” (Twelve number), provide lower accuracy and higher error rates, suggesting challenges in distinguishing similar visual patterns. Figure 10 shows examples of complex signs that achieved low accuracy due to similarity problems.



Number 11: The hand is contracted and facing upwards, with the thumb extended. The hand moves from the wrist to the right and left.

Number 12: The hand is contracted and facing upwards, with the index finger extended.

Number 13: The hand is contracted and facing upwards, with the index and middle fingers extended.

Number 14: The hand is contracted and facing upwards, with the index, middle, and ring fingers extended, while the thumb is joined to the palm.

Number 15: The hand is contracted and facing upwards, with all fingers extended.

Figure 10. The similarity between the signs in ArSL.

4. Discussion

The evaluation of the model performance through the comparison of “macro-” and “weighted” averages offers useful insights into how the distribution of classes affects the accuracy of categorisation. While “macro-averages" provide a simple average over all classes, “weighted” averages take into consideration class imbalance by assigning weights to the average based on the number of instances in each class. Our investigation revealed that both types of averages showed similar patterns across different circumstances, indicating the continuous impact of class distribution on the model results. Analysing the outcomes of every scenario clarifies the connection between the model performance, volunteer contributions, and dataset size. The best accuracy and F1 score were obtained in Scenario 1, when each volunteer provided 1500 data points, demonstrating the potency of the individual volunteer datasets. We observed a modest decline in the accuracy and F1 score in Scenario 2, as the dataset size rose with the merged data from several participants. Larger datasets may have advantages, but adding a variety of volunteer contributions could complicate things and impair the model performance according to this tendency. Additional analysis of the classification report offers valuable information about the specific difficulties faced by the model in distinct categories. Classes 10, 11, 12, and 13 demonstrated worse precision, recall, and F1 scores than did the other classes, suggesting challenges in successfully recognising these gestures. This difference highlights the significance of analysing metrics relevant to each class to discover areas where the model may need more refinement or training data augmentation to enhance its performance.

Several factors contribute to these classes' inferior performance. First, the nature of the movements within these classes may provide complexity that is difficult to fully determine. For example, these movements may include subtle gestures or minor differences between different signs, making it difficult for the model to distinguish between them efficiently. Furthermore, the classification model may have problems catching the intricacies of these movements, particularly if they include small fluctuations or sophisticated hand movements that are difficult to identify precisely. Moreover, the minimal size and diversity of the dataset for these classes may have contributed to the poor performance. A larger and more diversified dataset would give the model a broader set of instances, improving its capacity to generalise and identify these complex movements. To summarise, while the model's overall performance is acceptable, further modification and augmentation of the dataset, as well as the model architecture, are required to enhance the classification accuracy for these hard classes. This highlights the need for ongoing research and development efforts in the field of sign language recognition to solve these unique issues while also improving the accessibility and effectiveness of sign language recognition technology. The observed influence of an increasing dataset size emphasises the need for data augmentation and the establishment of larger, more diverse datasets in sign language recognition research. As part of the study's objectives, the goal was to create a comprehensive dataset exclusively for Arabic Sign Language recognition. By expanding the dataset, the model can be trained on a broader collection of instances, boosting its capacity to generalise and reliably identify sign language movements, especially in difficult categories. This is consistent with the overall goal of improving the accessibility and effectiveness of sign language recognition systems, ultimately leading to greater inclusivity and accessibility for people with hearing impairments.

5. A Comparison with Previous Studies

This study focused on the recognition of dynamic gestures performed with a single hand captured using a single camera setup. The primary goal was to recognise isolated dynamic words and dynamic numbers expressed through sign language gestures. The data collection process involved recording sessions where individuals performed these gestures in front of the camera, ensuring that the dataset captured a diverse range of hand movements and expressions, and by limiting the scope to dynamic gestures performed with one hand. Table 8 provides a comparison with prior studies that align with our objectives.

Table 8. Comparison with similar ArSL recognition systems.

Aspect	Proposed Work	Study [4]	Study [23]	Study [17]
Model Used	Long short-term memory (LSTM) with an attention mechanism	Convolutional neural network (CNN)	Convolutional neural network (CNN)	Hidden Markov models (HMMs)
Dataset Size	7500	7200	390	4045
Number of gestures	50 (30 simple, 20 complex)	20 (simple signs)	30 (simple signs)	30 (simple signs)
Gestures	Words and numbers	Words	Words	Words
Balanced data	YES	NO	NO	NO
Preprocessing	No need to convert the frames into greyscale	Convert the frames into greyscale	Convert the frames into greyscale	Convert the frames into greyscale
Feature Extraction Method	MediaPipe framework for hand and body keypoints	An adaptive threshold and adding a unique factor to each class	Two convolution layers with 32 and 64 parameters	Discrete cosine transform (DCT)
Best Accuracy	85% (individual volunteers), 83% (combined data)	92%	99.7%	90.6%
Real-World Applicability	Verified	Not verified	Not verified	Not verified

The dataset size in the proposed work is also significantly larger, at 7500 samples, compared to 7200 in Ref. [4], 390 in Ref. [23], and 4045 in Ref. [17]. A larger dataset contributes to better model generalizability and robustness, ensuring that the model performs well on diverse and unseen data. Moreover, the proposed framework handles 50 gestures, including both simple and complex signs, whereas the other studies focus primarily on simple signs (20 in Ref. [4], 30 in Ref. [23] and Ref. [17]). This broader range of gestures, which includes words and numbers, demonstrates the versatility and applicability of the proposed model for more comprehensive sign language recognition tasks. The data used in the proposed framework are balanced, ensuring that the model is trained on an equal representation of all gesture classes, reducing bias and improving the overall performance. In contrast, the datasets in Refs. [4,17,23] are not balanced, which could lead to skewed results favouring more frequent classes. For data collection, the proposed framework uses recorded videos with keypoint extraction using MediaPipe, a state-of-the-art framework for extracting hand and body keypoints. This method captures more detailed motion data than do the simpler approaches used in other studies, such as the smartphone videos in Ref. [4] and OpenPose version 1.4 in Ref. [17]. In terms of preprocessing, the proposed framework simplifies the process by not converting frames to greyscale, preserving more information from the original videos.

The MediaPipe feature extraction method used in the proposed framework is more advanced than methods, such as adaptive thresholding, convolution layers, and discrete cosine transform (DCT), which have been used in other studies. The proposed framework might not be as accurate as those used in other studies, but it is a strong and flexible solution for sign language recognition because it can better handle complex gestures, has a larger and more balanced dataset, uses advanced data collection and preprocessing methods, and can evaluate performance in real-time.

6. Conclusions

In this study, we attempted to meet the pressing need for effective communication tools for the deaf community by developing a model that can recognise dynamic hand gestures from video recordings. This was accomplished by combining the attention mechanism with LSTM units developed on a new ArSL dataset, namely, the DArSL50_Dataset. Keypoints were extracted from videos in the DArSL50 dataset using the MediaPipe framework. Subsequently, the features were fed into the proposed LSTM model to detect gestures. The results of our method were encouraging, with an average performance of 80–85%. The proposed model architecture demonstrated robustness in classifying hand motions despite variances in signing styles and recording conditions. The attention mechanism enhanced the framework's ability to recognise spatial arrangements and temporal relationships in sign language gestures by selectively focusing on key parts of the input sequences. Our research indicates that our method has considerable promise in enabling smooth communication between deaf and hearing populations. Future research could investigate other model architectures, such as Bi-LSTM, one-dimensional convolutional neural networks, convolutional recurrent neural networks, and transformer models. Additionally, there is potential for the creation of a large-scale dataset encompassing a variety of sign language gestures. Augmentation techniques could also be investigated to further enrich the dataset and improve the model's ability to generalise across various signing styles.

Author Contributions: R.S.A.A.: Conceptualization, Methodology, Writing—Original Draft; M.A.A.: Supervision, Methodology, Project Administration, Writing—Review & Editing; Z.T.A.-Q.: Data Curation, Software, Formal Analysis; M.M.S.: Validation, Investigation, Visualization; M.L.S.: Resources, Writing—Review & Editing. All authors have read and agreed to the published version of the manuscript.

Funding: No funding was received for this study.

Data Availability Statement: The source code for this study may be accessed at the following URL: https://drive.google.com/file/d/1FcXudNQqXb_IzehsdMWb0tSBplcq-8LJ/view?usp=sharing (accessed on 10 June 2024).

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

References

1. Ahmed, A.M.; Alez, R.A.; Taha, M.; Tharwat, G. Automatic translation of Arabic sign to Arabic text (ATASAT) system. *J. Comput. Sci. Inf. Technol.* **2016**, *6*, 109–122.
2. Ahmed, M.A.; Zaidan, B.; Zaidan, A.; Salih, M.M.; Al-Qaysi, Z.; Alamoodi, A. Based on wearable sensory device in 3D-printed humanoid: A new real-time sign language recognition system. *Measurement* **2021**, *168*, 108431. [CrossRef]
3. Alrubayi, A.; Ahmed, M.; Zaidan, A.; Albahri, A.; Zaidan, B.; Albahri, O.; Alamoodi, A.; Alazab, M. A pattern recognition model for static gestures in malaysian sign language based on machine learning techniques. *Comput. Electr. Eng.* **2021**, *95*, 107383. [CrossRef]
4. Balaha, M.M.; El-Kady, S.; Balaha, H.M.; Salama, M.; Emad, E.; Hassan, M.; Saafan, M.M. A vision-based deep learning approach for independent-users Arabic sign language interpretation. *Multimedia Tools Appl.* **2023**, *82*, 6807–6826. [CrossRef]
5. Tharwat, A.; Gaber, T.; Hassanien, A.E.; Shahin, M.K.; Refaat, B. Sift-based arabic sign language recognition system. In *Afro-European Conference for Industrial Advancement: Proceedings of the First International Afro-European Conference for Industrial Advancement AECIA 2014*; Springer: Berlin/Heidelberg, Germany, 2015.
6. Abdul, W.; Alsulaiman, M.; Amin, S.U.; Faisal, M.; Muhammad, G.; Albogamy, F.R.; Bencherif, M.A.; Ghaleb, H. Intelligent real-time Arabic sign language classification using attention-based inception and BiLSTM. *Comput. Electr. Eng.* **2021**, *95*, 107395. [CrossRef]
7. Suharjito; Anderson, R.; Wiryana, F.; Ariesta, M.C.; Kusuma, G.P. Sign language recognition application systems for deaf-mute people: A review based on input-process-output. *Procedia Comput. Sci.* **2017**, *116*, 441–448. [CrossRef]
8. Al-Saidi, M.; Ballagi, Á.; Hassen, O.A.; Saad, S.M. Cognitive Classifier of Hand Gesture Images for Automated Sign Language Recognition: Soft Robot Assistance Based on Neutrosophic Markov Chain Paradigm. *Computers* **2024**, *13*, 106. [CrossRef]
9. Samaan, G.H.; Wadie, A.R.; Attia, A.K.; Asaad, A.M.; Kamel, A.E.; Slim, S.O.; Abdallah, M.S.; Cho, Y.-I. MediaPipe's landmarks with RNN for dynamic sign language recognition. *Electronics* **2022**, *11*, 3228. [CrossRef]
10. Almasre, M.A.; Al-Nuaim, H. Comparison of four SVM classifiers used with depth sensors to recognize arabic sign language words. *Computers* **2017**, *6*, 20. [CrossRef]
11. Al-Shamayleh, A.S.; Ahmad, R.; Jomhari, N.; Abushariah, M.A.M. Automatic Arabic sign language recognition: A review, taxonomy, open challenges, research roadmap and future directions. *Malays. J. Comput. Sci.* **2020**, *33*, 306–343. [CrossRef]
12. Cheok, M.J.; Omar, Z.; Jaward, M.H. A review of hand gesture and sign language recognition techniques. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 131–153. [CrossRef]

13. Ahmed, M.A.; Zaidan, B.B.; Zaidan, A.A.; Salih, M.M.; Bin Lakulu, M.M. A review on systems-based sensory gloves for sign language recognition state of the art between 2007 and 2017. *Sensors* **2018**, *18*, 2208. [CrossRef]
14. Mohammed, R.; Kadhem, S. A review on arabic sign language translator systems. In *Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2021.
15. Jiang, X.; Satapathy, S.C.; Yang, L.; Wang, S.-H.; Zhang, Y.-D. A survey on artificial intelligence in chinese sign language recognition. *Arab. J. Sci. Eng.* **2020**, *45*, 9859–9894. [CrossRef]
16. Al-Rousan, M.; Assaleh, K.; Tala'a, A. Video-based signer-independent Arabic sign language recognition using hidden Markov models. *Appl. Soft Comput.* **2009**, *9*, 990–999. [CrossRef]
17. Youssif, A.A.; AAboutabl, E.; Ali, H.H. Arabic sign language (arsl) recognition system using hmm. *Int. J. Adv. Comput. Sci. Appl.* **2011**, *2*, 45–51.
18. Elons, A.S.; Abull-Ela, M.; Tolba, M. A proposed PCNN features quality optimization technique for pose-invariant 3D Arabic sign language recognition. *Appl. Soft Comput.* **2013**, *13*, 1646–1660. [CrossRef]
19. Ibrahim, N.B.; Selim, M.M.; Zayed, H.H. An automatic Arabic sign language recognition system (ArSLRS). *J. King Saud Univ. -Comput. Inf. Sci.* **2018**, *30*, 470–477. [CrossRef]
20. ElBadawy, M.; Elons, A.S.; Shedeed, H.A.; Tolba, M.F. Arabic sign language recognition with 3d convolutional neural networks. In Proceedings of the 2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS), Cairo, Egypt, 5–7 December 2017; IEEE: Piscataway, NJ, USA, 2017.
21. Ahmed, A.M.; Alez, R.A.; Tharwat, G.; Taha, M.; Belgacem, B.; Al Moustafa, A.M.; Ghribi, W. Arabic sign language translator. *J. Comput. Sci.* **2019**, *15*, 1522–1537. [CrossRef]
22. Mohammed, R.; Kadhem, S.M. Iraqi sign language translator system using deep learning. *Al-Salam J. Eng. Technol.* **2023**, *2*, 109–116. [CrossRef]
23. Halder, A.; Tayade, A. Real-time vernacular sign language recognition using mediapipe and machine learning. *J. Homepage* **2021**, *2582*, 7421.
24. Zhang, F.; Bazarevsky, V.; Vakunov, A.; Tkachenka, A.; Sung, G.; Chang, C.L.; Grundmann, M. Mediapipe hands: On-device real-time hand tracking. *arXiv* **2020**, arXiv:2006.10214.
25. Wu, T.-L.; Senda, T. Pen Spinning Hand Movement Analysis Using MediaPipe Hands. *arXiv* **2021**, arXiv:2108.10716.
26. Bazarevsky, V.; Grishchenko, I.; Raveendran, K. BlazePose: On-device Real-time Body Pose tracking. *arXiv* **2020**, arXiv:2006.10204.
27. Chen, K.-Y.; Shin, J.; Hasan, A.M.; Liaw, J.-J.; Yuichi, O.; Tomioka, Y. Fitness Movement Types and Completeness Detection Using a Transfer-Learning-Based Deep Neural Network. *Sensors* **2022**, *22*, 5700. [CrossRef]
28. Kartynnik, Y.; Ablavatski, A.; Grishchenko, I.; Grundmann, M. Real-time facial surface geometry from monocular video on mobile GPUs. *arXiv* **2019**, arXiv:1907.06724.
29. Alnahhas, A.; Alkhatib, B.; Al-Boukaee, N.; Alhakim, N.; Alzabibi, O.; Ajalyakeen, N. Enhancing the recognition of Arabic sign language by using deep learning and leap motion controller. *Int. J. Sci. Technol. Res.* **2020**, *9*, 1865–1870.

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