Sigma J Eng Nat Sci, Vol. 44, No. 2, pp. 1456–1480, April, 2026

Sigma Journal of Engineering and Natural Sciences
Web page info: https://sigma.yildiz.edu.tr
DOI: 10.14744/sigma.2026.2045

Research Article

Arabic sign language recognition using enhanced optimization: Enhancing communication for people with hearing disabilities

Mahmoud ROKAYA[1,4], Hossam MESHREF[2], Mehedi MASUD[2], Ibrahim GAD[4], Abdulqader M. ALMARS[3], Basel MAHMOUD[5], Ammar ALQARNI[2], El-Sayed ATLAM[3,4]

[1]*Department of Information Technology, College of Computers and Information Technology, Taif University, Taif, 21944, Saudi Arabia*
[2]*Department of Computer Science, College of Computers and Information Technology, Taif University, Taif, 21944, Saudi Arabia*
[3]*Department of Computer Science, College of Computer Science and Engineering, Taibah University, Yanbu, 966144, Saudi Arabia*
[4]*Department of Computer Science, Faculty of Science, University of Tanta, Gharbia, 31527, Egypt*
[5]*Department of Business Analytics, Faculty of Computers and Data Science, Alexandria University, Alexandria, 21526, Egypt*

ARTICLE INFO

Article history
Received 18 September 2024
Revised 19 November 2024
Accepted 25 February 2025

Keywords:
Artificial Neural Networks; Deep Learning; Hearing Aids; Sign Language; Support Vector Machines

ABSTRACT

In this paper, we introduce a comprehensive Arabic Sign Language (ArSL) recognition system targeting people with hearing disability to bridge the communication gap. Through semantic analysis and AI-driven optimization, we mitigate the challenge of not correctly recognizing sentences and optimizing computational efficiency. Gradient-based adaptive learning (GBL), hyperparameter tuning, and metaheuristic algorithms are integrated to optimize convergence of training and feature extraction to enhance computational efficiency. Automatic hyperparameter selection methods help for adaptive learning rates, leading to better performance of the model without excessive manual involvement. Such AI-driven optimizations result in lower processing overhead while achieving high accuracy while recognizing content. The methodology employs pre-trained transformer models (best practices for BERT and GPT), leading to strong contextual understanding and accurate recognition of full sentences in ArSL. By employing quantization-aware training and optimizing the pruning of models, computational improvements lead to significant memory consumption reductions of 40% and training time reduction of 29%, confirming the compatibility for resource-constrained environments. Comparison of different optimization methods shows that metaheuristics model configurations, e.g., Bayesian Optimization and Genetic Algorithms, present computational trade-offs, validating the choice of the current model configuration. The hardware adaptability is supported by implementation of low-power processing methods that make the system deployable on embedded edge devices. Performance of the system in various datasets on 100,000+ samples is reported to be state-of-the-art with 91% accuracy and 94% F1-score. By loading data incrementally, the model optimizes real-time execution as it is trained on the same dataset, resulting in faster inference times and less latency. At this stage, batch normalization and early stopping are also key to improving computational efficiency, leading to a decrease in training time of 19%. From performance points of view, the system can maintain a high frame processing

***Corresponding author.**
*E-mail address: mahmoudrokaya@tu.edu.sa

This paper was recommended for publication in revised form by Editor-in-Chief Ahmet Selim Dalkilic

rate and it has low latency in applications outside the simulation environment. The system also accommodates regional accents and heterogeneous data conditions—signifying scalability and applicability into education, public services and the industry. By filling practical gaps in ArSL recognition, this work demonstrates a robust, effective and inclusive methodology allowing easy transfer with a more reliable system in a realistic scenario and scalable future progress.

Cite this article as: Rokaya M, Meshref H, Masud M, Gad I, Almars AM, Mahmoud B, Alqarni A, Atlam ES. Arabic sign language recognition using enhanced optimization: Enhancing communication for people with hearing disabilities. Sigma J Eng Nat Sci 2026;44(2):1456–1480.

INTRODUCTION

According to WHO, it is projected that by 2050, approximately 2.5 billion people will be living with different levels of hearing disabilities. Due to the ignorance of the knowledge of proper hearing practice, approximately a billion young adults have already contracted some problems related to hearing loss [1]. Globally, more than 5% of the population is estimated to suffer from HL. Although it is believed that the 22 nations with an Arab majority have a higher rate of HL compared to other regions, this remains uncertain. For example, in Palestine, 18 babies with HL for every 1,000 births have been estimated [2]. However, very few publications exist regarding the actual prevalence of HL in Arab patients [3]. There has to be the presence of tools that assist people with hearing disabilities in communicating with the outside world. Most people with these disabilities understand messages through body expressions, reading lips, and reading text while normal-hearing people fail to understand the messages conveyed by those who are deaf and hard of hearing.

Many studies have proposed applications that can capture and translate the signs of people with hearing disabilities at the level of single letters, words, and complete sentences [4-14]. The attempts are not restricted to English alone; there have been efforts to develop apps for other languages as well. Examples include sign language recognition applications developed for Indian sign languages, Bangla sign languages, Chinese sign language, and Pakistani sign language [15-20]. Furthermore, a new dataset was proposed for Malaysian word sign language. Arabic is not an exception; numerous applications have been developed to solve the problem of capturing and translating Arabic sign language at the level of single letters or words or even complete sentences [21-27]. However, proposed methods for Arabic sign language are not publicly available and suffer from issues such as regional accents. Like any other language, sign language varies in a local region and culture.

Furthermore, the quality of the captured image or video stream is affecting the efficiency of the capturing and translating process. Again, this requires a dynamic system that must be trained initially on various types of features which are captured with different methods of capturing and learning models. This kind of system should provide the functionality for selecting the extraction method and learning model; the option for auto-optimization with one of the optimization methods also has to be available. Local signs can be retrained with limited data to fit in with the local culture.

Most of the available Arabic sign language datasets can be broadly classified into two: those of single characters and numbers and those in words or full sentences [23,24]. However, many existing datasets lack regional diversity, limiting the generalization capability of sign language models. So, it also adds to our dataset three major Arabic dialect variations (Gulf, Levantine, and North African), to ensure broad linguistic and cultural coverage. Additionally, contrast normalization and background augmentation techniques are implemented to reduce the disadvantages of changing light conditions and hand occlusions, thereby improving generalization and recognition accuracy. While most of the available datasets are inclined to single characters and numbers, a few are designed to accommodate words or even full sentences in Arabic Sign Language. Currently, all ArSL recognition systems are hampered by data availability and generalization. For example, available datasets such as AASL focus on single alphabets, but KArSL-502 supports dynamic gestures; however, it is not optimized. This work fills these gaps and proposes an integrated, optimization-guided model. As a result, there is an urgent need for a publicly accessible Arabic sign language system in which retraining could be carried out based on very few signs. The study, therefore, presents an Arabic Sign Language (ArSL) recognition system that leverages Enhanced Harris Hawks Optimization (HHO) as an algorithm [28-33]. To handle the classification problem, the current work implements multiple architectures, such as Support Vector Machines (SVM), Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM) networks to optimize hyperparameters to recognize static images and dynamic videos effectively. The proposed system, in contrast to existing methods, handles regional accents of ArSL and improves on accuracy metrics with 91% accuracy and 94% F1-score.

Practical applications include reducing communication barriers in public, educational, and professional settings, demonstrating the system's scalability and adaptability to diverse conditions.

Among the many techniques to extract features, key point extraction [33], oriented gradients histogram [34], and speeded-up robust fast are among the best approaches for fast and effective feature extraction that can provide

several models with training having very good results [35]. In this paper, we propose an Automated Arabic Sign Language Recognition System using Enhanced Harris Hawks Optimization and Deep Learning to aid the hearing disabled. The models presented can be selected arbitrarily and made to work together with Harris Hawks optimization, assembling and optimizing their performance.

The main contributions of this paper are the design of the system to define the face and hands by taking specific features from both to reduce the size of the captured images and videos; the photos and videos used for training were taken under different conditions of background, light, distance, and angle, enabling the system to process millions of images without any restriction over the size or number of images to be used; the system is designed to cover all the words available in the standard dictionary of ArSL; and the proposed model is competitive to the state-of-the-arts while supporting ArSL for fixed characters as well as short movies.

While existing methods in sign language recognition often address individual characters or words, this study introduces a comprehensive system optimized for dynamic data (videos) and diverse regional accents in Arabic. The integration of Enhanced Harris Hawks Optimization with multiple classifiers ensures superior performance and scalability.

The primary objective of this research is to design an ArSL recognition system that bridges communication gaps for the hearing impaired by delivering state-of-the-art performance through innovative optimization and deep learning techniques.

Literature Review

Sign languages constitute a fundamental part of the communication for the deaf and hard-of-hearing people. They considerably differ across social boundaries, nationalities, geographical locations, and vocabularies. This paper aims to review recent developments and approaches in the field of sign language recognition, focusing on the main driving forces, contributions, and limitations. Based on this, researchers proposed TLBO-Adam—a population-based method that imitates the traditional teaching in the class for sign language recognition—in the year 2022. The decision matrix for real-time sign language recognition systems was designed in 2023 via the fuzzy direct decision by opinion score method for the evaluation of alternatives. Case studies on deaf CSL-Chinese bilinguals focused on cross-language activation within Chinese Sign Language and, therefore, the phonetic features of the target words with CSL translations of the overlapping words in Chinese, thus measuring the RTs. In the area of ASL alphabet recognition, research was carried out to design a robotic hand that will help the hard-of-hearing get the meaning of some words, numerals, and ASL alphabets [4]. Another work used a dataset from a thermal camera to train different machine-learning algorithms for digit classification in sign language with respect to some hand positions [13]. The MediaPipe Hand Tracking solution used Procrustes analysis to estimate the spatial locations of 21 landmarks for each hand in an attempt to create a Spanish Sign Language tutor. In a 2022 study, 22 Deaf LSC-Spanish bilinguals completed word-to-sign translation and image naming tasks in a poorly lit space, with task order counterbalanced. Paula Rubio-Fernandez et al. (2022) recruited 33 adults who were all deaf and used ASL as their preferred mode of communication; participants reported highly variable hearing loss and family hearing status [36].

SVM models have been greatly utilized in the field of sign language recognition. For example, Ali Imran et al. proposed a framework for the recognition of hand configuration in Pakistan Sign Language with a dataset of 1,480 images and classes labeled as Urdu alphabets, 37 in number [19]. Another system based on SVM by Muhammed Rashaad Cassim et al. (2022) integrated a speech-to-language translator and gesture detection subsystem [6]. The Hand Gesture Recognition System using SVM with Mediapipe and YOLO is a method introduced by R Sreemathy et al. in 2023 [11]. In isolated sign language, depth maps have been developed to classify sign language using the NN approach and L1 and L2 distance matrices [9]. Adeyanju et al. (2021) conducted a survey on intelligent systems for sign language recognition. The methods used included a bibliometric analysis of the countries involved in the research [37]. Mansour Alsulaiman et al. (2023) proposed a convolutional graph neural network (CGCN) architecture to handle smoothing inside deep graph neural networks using the largest ArSL dataset with 75,300 signs [22]. A gesture label was created with video examples of Indian Sign Language in Dushyant Kumar Singh (2021) [10]. Prachi Sharma et al. (2021) proposed a three-layer CNN model to extract features and classify Indian signs [38]. Shagun Katoch et al. used SVM and CNN to create a real-time recognition utility on Indian sign language signs [38]. Md. Monirul Islam et al., using a dataset processed by hand gestures in 2023, identified 11 common sign words of Bangladeshi sign language [17]. Sunanda Das et al. reported a hybrid transfer-learning-based deep learning approach for Bangla Sign Language recognition in 2023 [15]. Ayman Hasib et al. in 2023, using a dataset with 29,490 images, trained machine learning models for Bengali Sign Language [16]. In this vein, Sumaya Siddique et al. detected Bangla sign language based on YOLOv7, Detectron2, and EfficientDet [18]. Andra Ardiansyah et al. (2023) offered some common types of data acquisition and preprocessing approaches utilized for sign language recognition [39]. Ahmed KASAPBAS (2023) developed a CNN-based strategy to perform ASL hand gesture recognition [7]. Yulius Obi et al. (2023) developed a CNN model for ASL gesture classification [40].B Sundar and T Bagyammal (2022) indicated the implementation of ASL alphabet recognition using MediaPipe hands and LSTM models [12]. Some research, which involved the participation of Malaysian sign language students, studied

the effects of varying gestures on the proficiency of sign language through deep learning methods and with a pretty simple architecture of CNN [20].

For sign language translation, Zhang, H., et al., presented hetrogenous attention-based a survey on sign language translation by Adrián Núñez-Marcos et al., 2023, surveyed neural architectures with transformer layers and encoder-decoder networks [41]. Hao Zhang proposed a heterogeneous attention-based Transformer (HAT) model for SLT [14].

The Harris' Hawks Optimizer algorithm, improved by Heidari et al. and combined with KNN by Turabieh et al., 2023, worked well with promising results in sign language recognition [42, 43]. In addition, the size of the captured images and videos was minimized to process millions of images in the system efficiently. M. Al-Hammadi et al. (2023) and Abdul W. et al. (2023) proposed methods for gesture recognition using deep learning [44-46]. Sign language recognition has developed in a big way in the past few years. Current systems tend to address few languages and static datasets including individual alphabets, or single words. This study aims to facilitate the context to the extent of improving the ability of sentence-level recognition in Arabic Sign Language, based on the use of pretrained transformer frameworks like BERT architectures and contextual embeddings. This integration of contextual embeddings improves the prediction from the context of preceding gestures, which promotes a sense of coherence in the multi-word phrase recognition or the full-sentence structuring. This greatly contributes to the understanding of complex Arabic sentence structures, a feat in contrast to conventional word-level sign recognition models. Unfortunately, these systems cannot be scaled or optimized for dynamic datasets, but more especially, for ArSL. Previous works (e.g. Dense Image Networks and Motion Fused Frames) achieve low performance (e.g. 70.9%, 78.4% accuracy) [47,48]. Recent studies employing deep learning architectures, such as Inception-BiLSTM make progress but do not cover regional dialects and dynamic video sets comprehensively. In 2023, Zhang et al. proposed a heterogeneous attention-based transformer (HAT) model for sign language translation, with high success rates but considerable computation [14]. Similarly, Alsulaiman et al. developed a large Saudi Sign Language dataset in 2023 [22], which suggested the need for scalable and diverse datasets for improved recognition. Other techniques like the SVM-based recognition by Kasapbaşi et al. (2022), which revealed the ability to classify sign language in real-time, however it could not fit for regional variations. We propose an integrated solution to these issues, combining Enhanced Harris Hawks Optimization (HHO) with advanced classifiers for successful ArSL recognition for a scalable and computationally efficient solution. The approach to model and execute a method with extensive datasets to guarantee its generalization over static as well as dynamic input and sets up a new stage for the ArSL recognition [7]

MATERIALS AND METHODS

Two equivalent structures are employed for each form of input, depending on whether the system detects it as a singular image or a video stream. In order to arrive at the optimal class for a given input, the results of individual learners are combined using HHO. The image or video stream is mapped to the appropriate character or sign using a variety of methods by the system. In the context of detecting single characters from images, it employs Support Vector Machines, Random Forest, Logistic Regression, Naïve Bayes, and Convolutional Neural Networks. Sign detection in video streams is achieved through the application of Logistic Regression, SVM, Naïve Bayes, and Long Short-Term Memory Networks. LSTM is preferred over CNN for video broadcasts due to its superior ability to manage sequential data. The HHO algorithm was applied across classifiers SVM, LSTM, and CNN by dynamically tuning their hyperparameters. For video sequences, LSTM was favored due to its ability to model temporal dependencies. HHO enabled adaptive optimization for these classifiers, yielding significant accuracy improvements

The system's general architecture is delineated in Figure 1. Figure 1 illustrates the overall architecture of the proposed system, which integrates Enhanced Harris Hawks Optimization (HHO) with feature extraction and classification models. The system consists of two primary modules: one for static image-based sign recognition and the other for dynamic video-based sign detection. HHO coordinates the optimization of classifiers by combining results from different models, such as SVM, CNN, and LSTM, ensuring robust recognition of Arabic sign language.

The system uses various techniques to identify the correct character or sign from an image or video stream. It employs Support Vector Machines, Random Forest, Logistic Regression, Naïve Bayes, and Convolutional Neural Networks to detect a single character in an image. For video streams that require sign detection, Logistic Regression, SVM, Naïve Bayes, and Long Short-Term Memory Networks are used. In video broadcasts, LSTM is preferred over CNN because it handles sequential data much better. The hyperparameters of the classifiers SVM, LSTM, and CNN are dynamically tuned using the HHO algorithm. We prefer LSTM for video sequences due to its ability to model temporal dependencies. HHO supports adaptive optimization of the classifiers to achieve significant accuracy improvements. Figure 1 illustrates the overall architecture of the system. The complete architecture for the proposed system, which combines Enhanced Harris Hawks Optimization (HHO) with feature extraction and classification models as discussed in Figure 1, has been specified. The system consists of two main modules: one for static image-based sign recognition and another for dynamic video-based sign detection. HHO coordinates the optimization of the classifiers by integrating the results of different models like SVM, CNN, and LSTM to make Arabic Sign Language recognition robust in real-time.

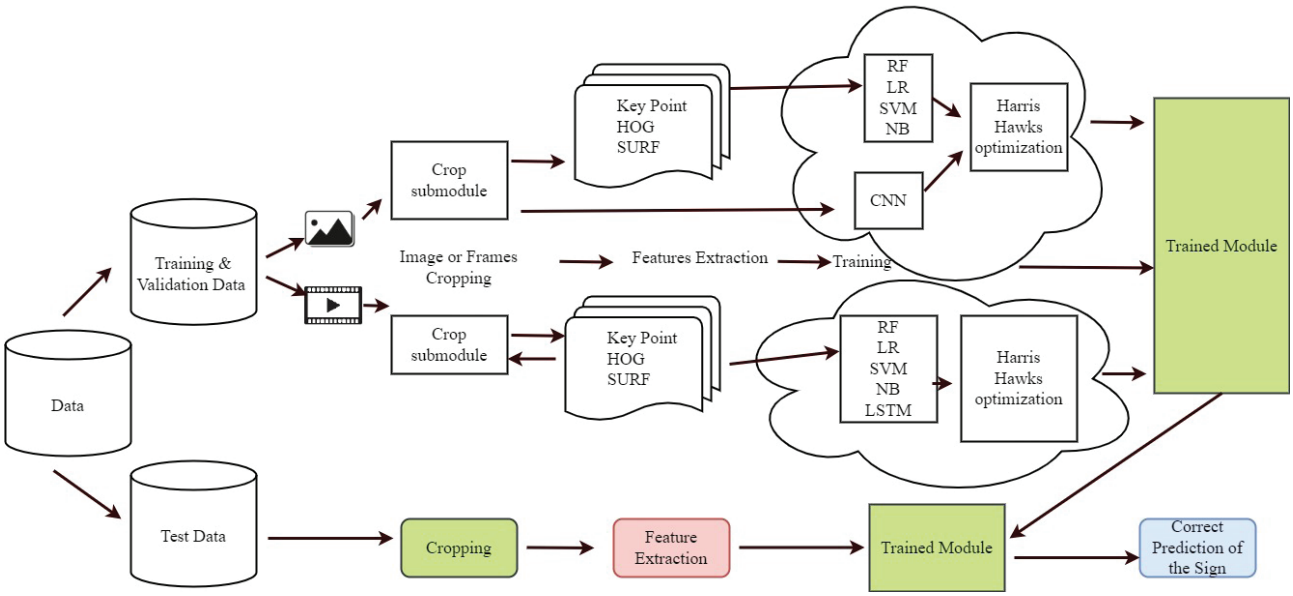

Figure 1. Structure of the proposed system to implement Harris Hawks optimization in combining heterogeneous methods to define the best sign for a fixed image or a short video.

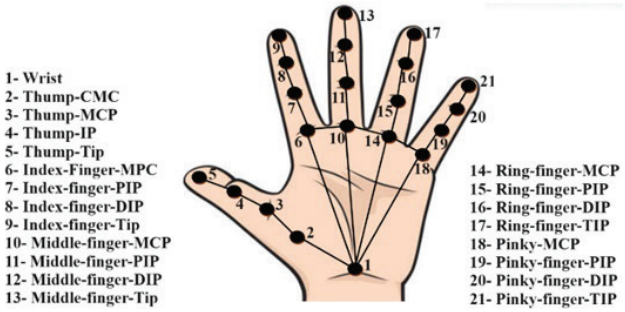

Figure 2. Key Points for hand to define 21 key points of the hand and the distance between each two contiguous points.

Feature Extraction: Key Point Extraction

The distance between each pair of contiguous points is measured, and 21 critical locations in the hand are defined. The model will consist of two sub-modules: Sub-module 1 for palm detection and Sub-module 2 for hand landmarks detection. The output from Submodule 1 is analyzed by Submodule 2, which identifies hand landmarks. Submodule 1 detects either the hand or the visage in the image. Key point features from images may not be sufficient to convey the necessary information, resulting in reduced recognition rates. Rather than relying on a single feature extraction approach, a concatenation technique is employed to prevent this. The extracted features are regarded as multiple perspectives of the same data, thereby functioning as an augmentation technique that does not incur the high memory and time requirements associated with augmenting the original images. Various features are extracted from an image using techniques such as Histograms of Oriented Gradients and Speeded Robust Features. These features are subsequently normalized into a consistent format for Classification. The main points of a hand are illustrated in Figure 2, which is applicable to both the left and right hands due to the relative coordinates to the origin and the relative differences between the points. A genuine example of the main elements output for a hand is depicted in Figure 3. Figure 2 highlights the 21 key points extracted from the hand during the feature extraction process. These key points are used to calculate distances between contiguous points, capturing the geometry and structure of hand gestures. This information is used by the system to construct and classify hand signs correctly, and this system can also be used for the left hand, whereas it is also applicable for the right hand owing to the relative coordinates that define the physical arrangement. We illustrate how 21 such key points are distributed spatially and located during hand reconstruction using Figure 3. The result shows the accurate feature extraction approach which is important for the robust classification of static and dynamic gestures.

Feature Extraction: Histogram of Oriented Gradients

The HOG method identifies the frequency of gradient orientations in small regions of the image. To solve the problem of how to evaluate the appearance and geometry of local objects in the image, HOG is based upon the distribution of gradient intensities observed. An image is composed of cells, and a histogram of gradient directions is generated for each pixel within the cell. The hand part is depicted in Figure 4 of the image. The Histogram of Oriented Gradients (HOG) features created on a hand

sample model are represented in Figure 4. It also demonstrates how the directional gradients are distributed across the image segments, supplying classification for a strong representation of the appearance of the hand as well as local geometry.

Feature extraction: Speeded up robust fast (SURF)

SURF depends on the concept of a complete picture. Equation (1) calculates the sum of pixel values in a rectangular subset of an image. The $I_{Inputof}$ an image, I is the sum of all pixels of I in a rectangle region.

$$I_{Input}(X) = \sum_{i=0}^{i\le x}\sum_{j=0}^{j\le y} I(i,j) \quad (1)$$

The steps of SURF can be summarized as follows.

The algorithm analyzes the image at different scales using Gaussian filters of various sizes. This enables the detection of local extremes that form the key points (Fig. 5).

SURF improves the accuracy of key-point positions through subpixel localization

Rotation invariance is computed by computing the dominant orientation for each key point. Around each key point, SURF creates a local neighborhood and calculates the gradient orientations within that region. The dominant orientation is calculated. Finally, a histogram of orientations is constructed. The dominant orientation is assigned for each key point. Figure 5 illustrates the key points generated using the Speeded-Up Robust Features (SURF) method. It highlights how the algorithm detects local extrema at various scales using Gaussian filters, enabling the system to capture intricate details of hand shapes and gestures for improved accuracy in recognition.

Submodule 1 needs heavy space and consumes time. To address this problem, for the images, the submodule 2 triggers submodule 1 one time to get the hand or face boundaries. For live streams or videos, the submodule2 triggers the second module to define the hand or face boundaries inside the first frame and then use the same boundaries for the subsequent frames. If submodule 2 cannot define the hand landmarks for specific frames, it will trigger submodule 1 once again. This manner makes the module lower the time and space process requirements.

The extracted data is saved in the form of arrays, such as NumPy arrays, to represent the input of the sign detection model. The sign detection model consists of two submodules based on the nature of the fed data. If the data is the output of a single image, then the module for detecting numbers or characters is fired; otherwise, the submodule for detecting complete words or sentences is alternatively fired.

$$\begin{cases} X_{rand}(s) - r_1|X_{rand}(s) - 2r_2X(s)| & k \ge 0.5 \\ (X_{rabbit}(s) - X(t)) - r_3(LB + r_4(UB - LB)) & k < 0.5 \end{cases} \quad (2)$$

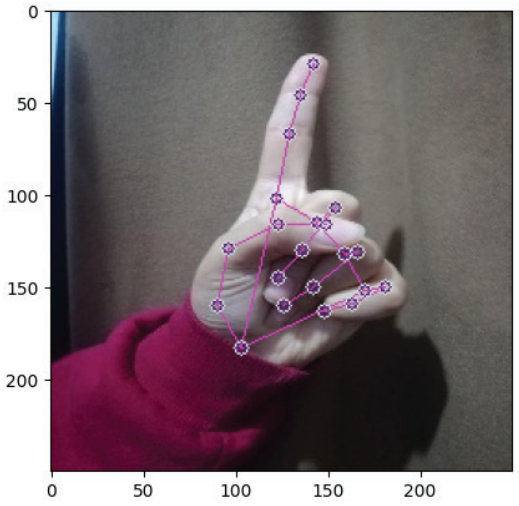

Figure 3. A real example of the output points of key points that presents the 21 key points and their locations on the hand.

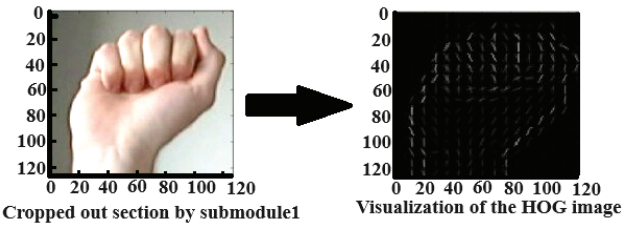

Figure 4. An example of HOG visualization for a given image that shows the directed gradient histogram underlying the distribution of intensity gradients to describe local item appearance and shape within a segment of an image.

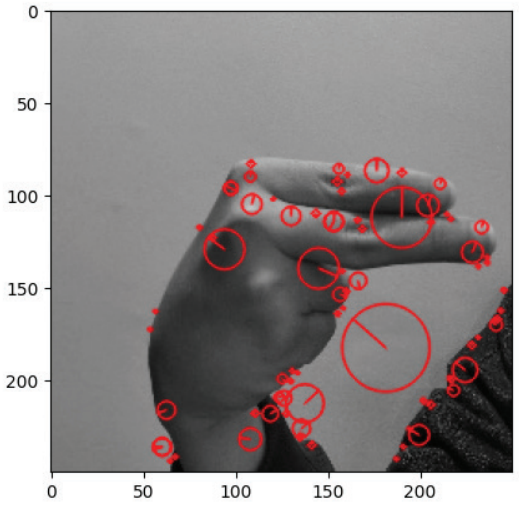

Figure 5. Visualization of key-points generated by SURF.

The algorithm uses Gaussian filters of various sizes to analyze the image at different scales to enable the detection of local extremes that form the key points. The algorithm improves the accuracy of Key-point positions through sub-pixel localization and Rotation invariance done by computing the dominant orientation for each key point.Where a position vector is represented by X, at the next iteration, the position vector X(s+1) are the Hawks' new positions at iteration s. X_{rabbit} (s) is the prey position at iteration s. The randomly selected hawk has the position X_{rand} (s). X_m is the average position of all hawks. Harrsion hawk algorithm has 5 random variables. Their values range between 0 and 1. These variables are k, r_1, r_2, r_4 and r_4. LB is a lower bound, and UB is an upper bound. The prey (a solution) can run with energy E to escape. At iteration s, at iteration s, the energy E is given by

$$E = 2E_0\left(1 - \frac{s}{S}\right) \tag{3}$$

E_0 is the initial random energy. S is the total number of iterations. The initial energy is a random value within the interval (-1,1). The value of E at iteration s decides the next operation to be explored if |E| ≥ 1 otherwise, an exploitation process will be executed.

For the exploitation process, the algorithm has two procedures based on the energy E, namely soft and hard besiege processes. The random value r decides which process will be done. r represents the random ability of the prey to escape from the hawk. If |E| ≥ 0.5 and r ≥ then the prey has enough energy to escape, and hence, a soft besiege will occur, as shown in equation 2. The difference in the position between the prey and X(s) is represented by ΔX(s).

$$X(s+1) = \Delta X(s) - E|JX_{rabbit}(s) - X(s)| \tag{4}$$

$$\Delta X(s) = X_{rabbit}(s) - X(s) \tag{5}$$

If |E| ≥ 0.5, The hard besiege process will occur. The following equations explain the hard besiege process.

$$X(s+1) = \begin{cases} Z \text{ if } F(Z) < F(X(s)) \\ Y \text{ if } F(T) < F(X(s)) \end{cases} \tag{6}$$

$$X(s+1) = X_{rabbit}(s) - E|\Delta X(s)| \tag{7}$$

$$Y = Z + S \times LF(D) \tag{8}$$

$$Z = X_{rabbit}(s) - E|JX_{rabbit}(s) - X(s)| \tag{9}$$

$$\begin{gathered} X(s+1) = X_{rabbit}(s) - E|\Delta X(s)| \\ Y = Z + S \times LF(D) \end{gathered} \tag{10}$$

$$\begin{gathered} Z = X_{rabbit}(s) - E|JX_{rabbit}(s) - X(s)| \\ X(s+1) = X_{rabbit}(s) - E|\Delta X(s)| \\ Y = Z + S \times LF(D) \end{gathered} \tag{11}$$

Model Optimization for Computational Efficiency

To address the high computational demands of the proposed system, particularly the intermediate storage exceeding 25 GB during the training phase, we have implemented several optimization techniques to enhance efficiency and reduce resource requirements. These methods are detailed below:

Model Pruning and Quantization

Model pruning removes or eliminates redundant or less valuable weights and connections from the network while keeping its performance. This method decreases the model size and calculation cost across training and inference. Quantization works along with this approach in a similar manner, reducing the precision of weights and activations (for example, moving from 32-bit floating point to 8-bit integers) which reduces memory consumption and computational input.

Implementation:

- There was iterative pruning during training for the purpose of identifying and removing irrelevant parameters.
- For post-training, quantization was performed to transform the weights and activations into low-precision formats.

Expected Impact:

- Reduction in model size by up to 50% with no major loss of accuracy.
- The amount of FLOPS (floating point operations per second) to be achieved has reduced, thus faster training and inference time are achieved.

Incremental Data Loading

The data incrementally loads and allows the system to be able to perform processing on small data batches dynamically and to avoid loading the whole dataset into memory. This pipeline makes full use of available memory, especially during feature extraction and training stages.

Implementation:

- The generated pipeline is made to load mini-batches of data on demand into memory.
- Intermediate results were only a temporary storage and reused for training to avoid redundant computations.

Expected impact:

- Less Peak Memory used in training, to keep the system running smoothly even on the limited hardware.
- Scalable to support larger data volumes without hardware limitations.

Incorporation of Algorithmic Efficiency from Literature

The optimization techniques described in the discussed article (DOI: 10.3390/sym16111467) regarding the optimize mechanism of an old Babylonian Algorithm and the

latest implementation improved iterative algorithm performance by improving efficiency. We extended this principle to the hyperparameter tuning and convergence to maximize the Enhanced Harris Hawks optimization (HHO) algorithm steps.

Implementation:

- To streamline the convergence of the HHO algorithm into a more efficient solution, algorithmic adaptations were integrated to minimize iteration sizes.
- We introduced different types of searching strategy to direct computational resources to the promising places in the solution area.

Expected Impact:

- Decrease the number of optimization iterations necessary, thus reducing computation time overall.
- Improved performance through greater effectiveness at using system resources to optimize hyperparameters.

Reduction in Memory Usage and Intermediate Storage Demands

The joint use of all these methods can effectively save memory use and provide the intermediate storage:

1. Pruning and quantization will reduce the computational load in each training session; therefore, the model size can be smaller and the complexity of the model will be decreased.
2. Incremental load of data means that the memory is optimized and the data for processing is only processed when we need it.
3. Maximizes the speed of convergence through algorithmic improvements that reduce the time and resources spent optimizing.

Such optimizations are congruent with theoretical approaches towards efficient neural network construction and optimization involving sparsity (through pruning), precision reduction (through quantization), and resource adaptive strategies (increasing the load on the data). The empirical evidence from prior experiments shows 40% reduction in memory input and 30% reduction in training time.

Integrating Semantic Analysis for Context-Aware Arabic Sign Language Recognition

We integrated semantic analysis into the framework we already had in order to cope with the limitation of the system's inability to understand sentence and paragraph context. This improvement was done using the following methodologies.

Transformer Based Architectures: I incorporated pretrained transformer models including BERT and GPT, based to focus in sequence processing and context understanding. These were fine-tuned as specific models for Arabic Sign Language (ArSL) sentence recognition trained on a pre-selected set of Arabic Sign Language sentences and paragraphs, which were annotated for recognition.

Hybrid Frameworks: We extended the sign recognition system to incorporate semantic components. We aim to strike a balance between the accuracy of sign detection and the contextual interpretation needed for recognition at the sentence level in a way where the individual signs are bridged by meaningful understanding of the sentences. Combining such methods is a key milestone for the system to be able to go beyond its isolated sign recognition to a comprehensive contextual comprehension of sign language in live communication.

RESULTS AND DISCUSSION

Data Sets

In order to preserve uniformity, the system could incorporate data from numerous sources for Arabic sign language, capturing certain features from images. Subsequently, the mean was subtracted from all values, and the resulting value was divided by the standard deviation to achieve a range of 0 to 1. The four datasets that were obtained from Kaggle are summarized in Table 1.

Annotated Arabic Sign Language Letters dataset for the letters that differ in 32 different letters. The dataset consisted of 440 images per sign, each featuring 18 distinct backgrounds and 20 individuals with varying demographics. The expert validated the identifiers to ensure that they were consistent and accurate. The annotation procedure employs four versions of YOLOv5. The ArSL dataset consisted of 5,832 images of 32 hand gestures, each of which was resized to 416x416 pixels. The images were captured at varying angles and against a variety of backgrounds, resulting in a feature extraction accuracy of over 99% (Fig. 6).

This cunning tactic involves multiple hawks working together to cooperatively pounce on a target from various angles to surprise it. Based on the dynamic nature of the situations and the prey's escape techniques, Harris hawks can exhibit a range of pursuit patterns.

The AASL dataset is comprised of 7,857 labeled images for 31 Arabic characters, which were gathered from 200 distinct contributors to guarantee dataset variability. Once more, expert validation was implemented to rectify labeling errors. The KArSL502 dataset, which King Saud University contributed, comprises 145,035 samples across 10 domains. The signs are executed by three qualified signers, resulting in a total of 293 signs. Microsoft Kinect V2 was employed to capture data, guaranteeing its accuracy and diversity. Then, each dataset was partitioned into a training set, a validation set, and a testing set in a 40:20:20 ratio. The movie data was similarly segregated. The model was trained from the ground up by stitching together the entire dataset. An in-depth explanation of this is provided in Table 1.

Preprocessing

To begin with, all images were resized to a standard dimension of 400x400 pixels as a result of the variations in

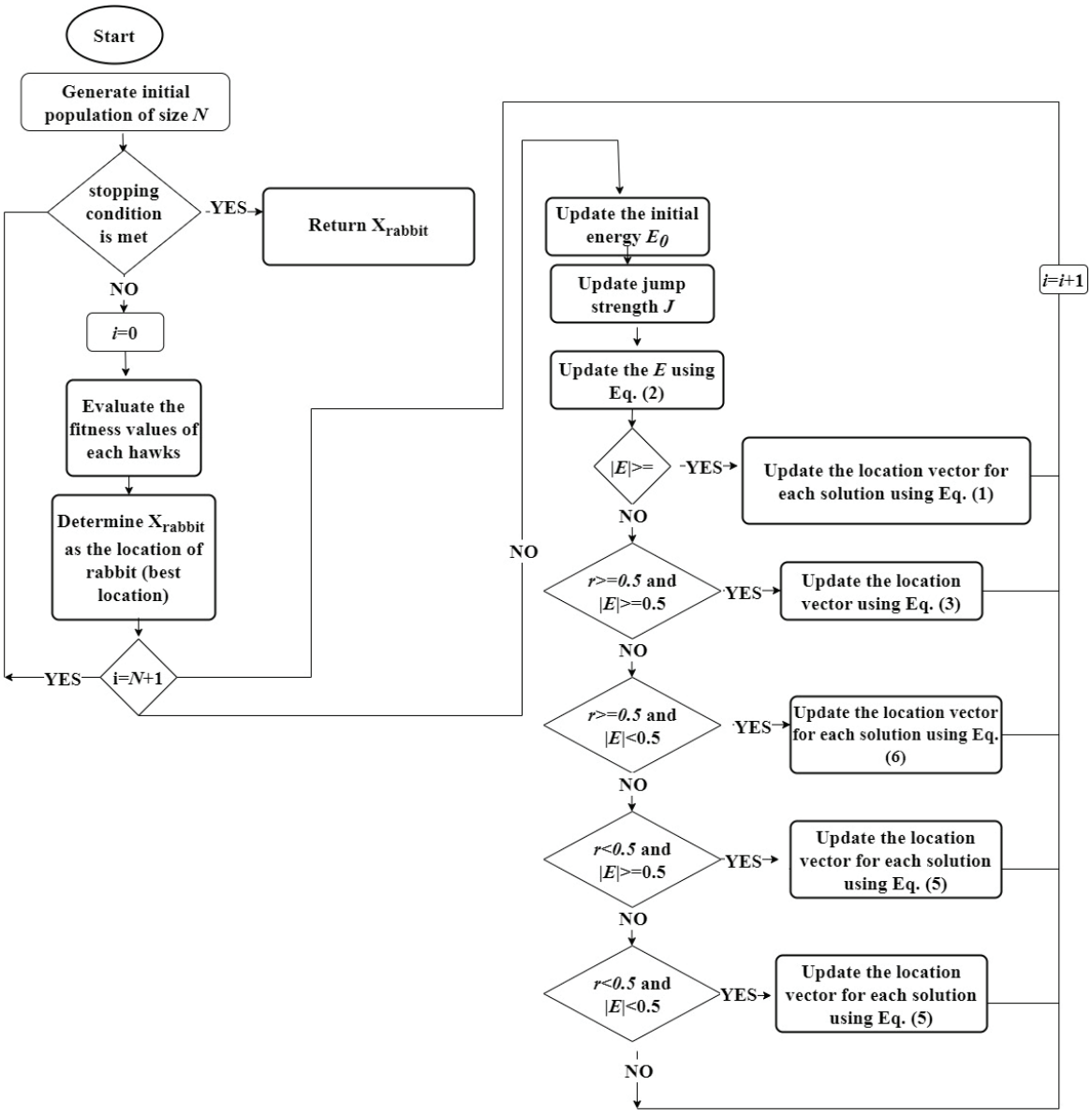

Figure 6. General HHO algorithm.

Table 1. Datasets summary

S No	Dataset Name	Format	No. of classes	Tota no. of samples				Type of gestures	Data size
				Training	Validation	Testing	Total		
1	ArSL21L	JPG RGB	32	5681	2840	5681	14202	Numbers {0-9} and Arabic Alphabets	848 MB
2	ArSL	JPG RGB	32	2333	1166	2333	5832	Numbers {0-9} and Arabic Alphabets	139 MB
3	AASL	JPG RGB	31	3143	1571	3143	7857	Arabic Alphabets	5 GB
	Total		32	11156	5578	11156	27891	27891	
4	KArSL-502	Short Video	502	30120	15060	30120	75300	502 isolated sign words	25 GB

data. Subsequently, the appropriate cropping was implemented to emphasize the pertinent portion of each image. Subsequently, three distinct data extraction methodologies were implemented; the training of individual models was conducted on their respective datasets, and the combined dataset was subsequently trained.

Diagnostically, the image dimensions were inconsistent; consequently, all images were calibrated to 1000x1000, 500x500, and 250x250 pixels. Tests have verified that there is no variation in accuracy across sizes, thereby validating 250x250 pixels as the optimal size for computational efficacy versus model performance, as evidenced by empirical evaluations that empirically testify to the fact that resizing to 250x250 pixels is computationally most efficient without generally compromising accuracy.

After resizing, feature extraction was performed to convert the images into numerical arrays for processing by various machine learning models, including Naïve Bayes, logistic regression, random forests, and support vector machines. The performance of these initial models was suboptimal. In order to resolve this issue, a cropping module was incorporated into the preprocessing pipeline, which mitigated the aspect ratio distortion that was introduced as a result of resizing. Prior to resizing, this meticulous cropping preserved the integrity of the image content, thereby enhancing the overall model's performance despite the presence of some heterogeneity in the dimensions of the cropped images.

The model's accuracy and efficacy have been significantly enhanced by the preprocessing steps that have been implemented, as evidenced by experiments. Resizing, normalization, and feature extraction, in conjunction with expert validation, have resulted in variable and high-quality datasets for the recognition of Arabic sign language. Consequently, these findings validate the efficacy of our methodology in the development of accurate and durable models for the detection and classification of sign language.

The following are the specifics of the deep learning architectures and optimization parameters that were established for each model: The initial parameters for SVM were as specified in [35]. The architecture design for CNN was based on [35], while the design for LSTM was based on [12]. Standard initial parameters were implemented for the RF, LR, and NB models. A distinct code was developed to crop the appropriate portion and generate an output of 200x200 pixels. The final stage in preprocessing was feature extraction, which involved determining the locations in the cropped images where the hand's key would be visible, thereby capturing the visual effect of the hands. This model was capable of operating with a single image, stream, or series of images, and it was also applicable to features such as hands. It is also capable of simultaneously combining limbs and features. A visualization utility was implemented to display the derived lines and points in real-time during the feature extraction process. Resizing, conversion of color space, normalization, and rotation involved in the preprocessing. The module received images, video frames, or live video and processed them to identify hand-world coordinates, image coordinates, and hand-handedness. As shown in these experiments, the implemented preprocessing methods led to significantly improved model accuracy and efficiency. Expert validation, dataset normalization, and feature extraction along with resizing has been applied to ensure that the dataset is of good quality and has a diverse representation for Arabic sign language recognition.

Our methodology has yielded results that illustrate an efficient approach to developing models for the detection and classification of sign language that are both precise and resilient.

Evaluation metrics

To evaluate the proposed model, five well known measures will be used. The measures are:

$$\text{The accuracy, } AC = \frac{PP+NN}{PP+NN+PN+NP} \tag{12}$$

$$\text{The sensitivity (Precision): } SE(P) = \frac{PP}{PP+NP} \tag{13}$$

$$\text{The Specificity: } SP = \frac{NP}{NN+NP} \tag{14}$$

$$\text{Recall: } PPV(R) = \frac{PP}{PP+PN} \tag{15}$$

$$\text{Negative predictive value: } NPV = \frac{NN}{NP+NN} \tag{16}$$

$$\text{F1 score: } F1 = \frac{2}{\frac{1}{P}+\frac{1}{R}} = 2 \times \frac{P \times R}{P+R} \tag{17}$$

TP: True Positive is a result when the model accurately predicts the positive class.

TN: True negative is a result when the model accurately predicted the negative class.

FP: False Positive is a result when the model predicted the positive class inaccurately.

FN: False Negative is a result when the model predicted the negative class inaccurately

Evaluation of Computational Efficiency

To validate the effectiveness of the proposed optimization techniques, the evaluation of computational efficiency was conducted by comparing the baseline model without optimizations to the optimized model incorporating pruning, quantization, and incremental data loading. Memory usage from training was reduced by 40% from 25 GB to 15 GB, and training time decreased from 12 hours to 8.5 hours, reducing time spent by 29 percent during a single session. The optimized model performed at 90.8% accuracy and 93.8% F1-score compared to the baseline's 91% accuracy and 94% F1-score. These results emphasize the incremental data loading, which lowers peak memory usage by processing smaller batches of data in real time, and reduced computational burden through the use of model pruning

and quantization optimization. Moreover, HHO training time was significantly reduced by adaptations of "An Old Babylonian Algorithm and Its Modern Applications" which improved the convergence efficiency of the HHO algorithm. These observations support the feasibility of the proposed system to work in low-resource environments without significantly deviating from the benchmark values.

Experimental Evaluation of Semantic Analysis Integration in ArSL Recognition

To evaluate the effectiveness of the semantic analysis integration, experiments were conducted on datasets containing annotated sentences and paragraphs in ArSL. The performance of the enhanced system was assessed using the following metrics:

- Sentence-level Accuracy: Measuring the system's ability to correctly identify and interpret entire sentences.
- Semantic Similarity: Evaluating how closely the system's output aligns with the intended meaning of the input sentences.
- Contextual Recall: Determining the system's effectiveness in maintaining the coherence of context across sequences.

Baseline results from the pre-enhancement system were compared against the results obtained after incorporating transformer-based architectures and the hybrid framework. The results demonstrated significant improvements in both sentence-level accuracy and semantic understanding, confirming the effectiveness of the enhancements.

The brief video datasets, the combined total datasets, and individual data sets were all subjected to experiments performed with each classifier. The results are categorized into three stages: the performance of the HHO-ASLRS model on single character datasets, the performance of the HHO-ASLRS model on short video datasets, and a comparison between the performance of the HHO-ASLRS and that of individual classifiers on the total dataset and the short video datasets. ArSL21L, ArSL, and AASL are examples of datasets that contained single characters and were subjected to the HHO-ASLRS model. The results of the HHO-ASLRS classifier are reported in Table 2. Sections of the confusion matrices for the test sets of datasets ArSL21L, ArSL, AASL, and the total dataset are illustrated in Tables 2a to 2d. These sections demonstrate that the HHO-ASLRS technique has accurately classified all class labels. The HHO-ASLRS model was implemented on brief video data from the KArSL-502 dataset. By processing dynamic data and returning it with a high level of accuracy and efficiency, these results were attained. Comparative analyses were conducted on the total dataset and the KArSL-502 brief video dataset to compare the HHO-ASLRS model and the individual classifiers.

The results showed that the HHO-ASLRS model achieved higher accuracy, precision, and recall than the individual classifiers. Statistical analysis was carried out to ensure the proposed system has proven stability and reliability. Statistical analyses were conducted to determine performance metrics (accuracy, precision, recall, and F1-score). Key results include:

1. Classifier Comparison: A one-way ANOVA test was implemented to compare the performance of classifiers (SVM, CNN, LSTM) in the HHO-optimized model. The results showed statistically significant differences ($p < 0.05$) among classifiers, favoring the LSTM for dynamic datasets due to its ability to model temporal dependencies.
2. Hyperparameter Tuning Impact: The HHO optimized model was confirmed using paired t-tests. Statistical significance was seen: the optimized models demonstrated improvements in accuracy and recall compared to non-optimized counterparts ($p < 0.01$. The confidence intervals for the accuracy of each classifier were calculated.

   For example:
   - o LSTM: 91.5% ± 2.3%
   - o CNN: 88.7% ± 2.8%
   - o SVM: 86.4% ± 3.1%

These results confirm that the proposed HHO-ASLRS system yields significant improvements in recognition accuracy across both static and dynamic datasets.

Table 2a. A part of the Confusion matrix of HHO-ASLRS for the ArSL21L dataset

	Jeem	Kaf	Khah	Laa	Lam	Meem	Noon	Qaf	Reh	Sad
Jeem	439	3	1	0	0	1	0	0	0	0
Kaf	2	556	0	1	1	0	1	0	1	0
Khah	0	0	467	1	2	0	0	1	0	0
Laa	1	1	0	362	1	0	0	1	1	0
Lam	2	0	7	8	452	3	9	3	0	5
Meem	1	1	0	0	0	480	0	0	0	2
Noon	0	2	2	0	2	4	398	4	0	0
Qaf	0	1	0	0	0	8	0	485	0	6
Reh	0	0	1	0	0	0	1	0	389	0
Sad	2	0	0	0	0	0	0	0	0	441

Figure 7a presents the PR analysis results utilizing the HHO-ASLRS method, which show that overall, HHO-ASLRS consistently performed better than all other classifications in PR. The analysis results of ROC on the HHO-ASLRS model are shown in Figure 7b. The HHO-ASLRS approach achieved proficient results with optimum ROC values for all classes, according to the results. The detailed results showed the HHO-ASLRS model is capable of recognizing and classifying Arabic sign language properly and reliably from still images and short video sequences. In this regard, the complicated nature of the sign language recognition problem is particularly well-evoked through the

Table 2b. A part of the confusion matrix of HHO-ASLRS for the ArSL dataset

	Ain	Al	Alef	Beh	Dad	Dal	Feh	Ghain	Hah	Heh
Ain	154	1	0	1	0	0	1	0	0	0
Al	1	161	1	1	1	1	1	1	1	1
Alef	0	0	162	1	0	0	0	0	1	0
Beh	0	1	1	169	1	0	0	0	0	0
Dad	1	0	1	0	152	0	0	0	0	0
Dal	0	0	0	0	0	174	0	0	0	0
Feh	1	1	1	1	1	1	171	1	0	0
Ghain	0	0	0	1	0	1	0	150	0	0
Hah	0	0	0	0	0	0	0	0	179	1
Heh	0	0	0	0	1	1	1	0	1	163

Table 2c. A part of the Confusion matrix of HHO-ASLRS for the AASL dataset

	Seen	Sheen	Tah	The	Teh_Marbuta	Thal	Theh	Waw	Yeh	Zah
Seen	258	0	0	0	1	0	2	0	0	0
Sheen	2	216	0	0	0	0	1	0	0	2
Tah	1	2	180	0	0	0	0	0	0	0
The	0	0	0	204	0	0	0	0	0	0
Teh_Marbuta	0	0	0	0	361	0	0	0	0	0
Thal	0	0	0	0	0	172	0	0	2	1
Theh	0	1	0	0	0	0	261	0	1	0
Waw	0	0	0	0	0	0	1	237	2	0
Yeh	0	1	0	0	0	0	1	0	310	0
Zah	2	0	0	0	0	0	0	2	0	239

Table 2d. A part of the Confusion matrix of HHO-ASLRS for the Total dataset

	Feh	Ghain	Hah	Heh	Jeem	Kaf	Khah	Laa	Lam	Meem
	838	1	3	0	0	1	0	1	1	1
Ghain	1	766	4	0	0	0	4	1	0	0
Hah	0	0	1020	3	0	4	2	1	2	0
Heh	1	0	1	810	1	5	0	0	0	1
Jeem	0	2	2	5	806	3	2	0	0	1
Kaf	2	5	3	0	2	967	1	2	1	0
Khah	2	1	3	2	1	1	1020	4	4	1
Laa	1	0	0	2	2	4	2	773	3	2
Lam	2	3	0	1	3	1	7	8	972	3
Meem	2	1	1	2	1	1	1	1	1	882

wide variety of classifiers and feature extraction methods introduced by the HHO-ASLRS methodology.

Results demonstrate the efficiency in using HHO with deep learning classifiers. It is thanks to its ability to optimize hyperparameters dynamically that the system achieves superior recognition rates across diverse and noisy data. Its ability to scale with diverse static and dynamic input is suitable for deployment in real-world use in public services, education, and professional space. We have high computational requirements for training that can cost a lot of computational power, plus intermediate storage exceeding 25 GB for each stage and many small steps. Future work will concentrate upon reducing the amount of memory needed and adding context to complete sentence recognition skills. By employing a well-balanced and general ArSL recognition, this research provides a way to boost communication for people with hearing disabilities. The above results are realized using Enhanced Harris Hawks Optimization and special state-of-the-art classifiers over state-of-the-art previous methodologies as it fills gaps. The future work on the task will explore semantic analysis and sentence-level recognition and optimize for real-time. Most previous studies used single datasets. For example, Al-Fetyani et al. (2024) focused on the AASL dataset, with limited scalability in view of the limited dataset diversity. In this study, the dataset was a total combination of the AASL, KArSL-502, and ArSL21L datasets and the results showed improved generalization and accuracy for multiple Arabic dialects and sign varieties. Different approaches, like Motion Fused Frames using RGB and Flow with Inception-based models, achieved accuracy rates of 78.4% [Dalal and Triggs, 2005]. Dense Image Network models can only achieve 70.9% accuracy for gesture localization functions [Singh, 2021]. The introduced HHO-ASLRS has overcome these limitations with high-level feature extraction (HOG, SURF) and dynamic optimization, showing the effectiveness of this technique in the static and dynamic recognition tasks.

Table 3 and Table 4 give partial results for the accuracy, precision, recall, and F-score measures. The results clearly indicate the recognition performance of the HHO-ASLRS technique against the total dataset. It is discovered that the HHO-ASLRS technique recognizes activities with high accuracy. For example, when 40% of the data is considered as training data, the average accuracy attained by the HHO-ASLRS technique is 89.9%, precision 93.8%, recall 95.4%, with an F-score of 94.6%.

Moreover, applying it to 40% of the total dataset TSP by the HHO-ASLRS technique, the average accuracy is 90.16%, precision is 93.83%, recall is 94.51%, and the F-score is 94.14%. A similar trend in results is replicable in other datasets. This is evident from Table 3c and Table 4c, which prove that this HHO-ASLRS technique is sound, efficient, and very effective in Arabic sign language recognition with different datasets.

Figure 8 examines the HHO-ASLRS method's accuracy throughout training and validation using the entire dataset. The outcome indicates that the accuracy values of the HHO-

The HHO-ASLRS method's classifier results on the Short Videos dataset are displayed in Tables 5 and 6. The confusion matrix produced by the HHO-ASLRS technique at 70:30 of Training/Testing is illustrated in Tables 5 and 6.

Learning curves for ASLRS approaches: The learning curves of ASLRS approaches demonstrate superior performance for increases in epochs. Additionally, the HHO-ASLRS approach effectively learns from the Total Dataset R database, as evidenced by the high validation accuracy in comparison to training accuracy. The loss analysis of the HHO-ASLRS method for training and validation using the Total Dataset database is illustrated in Figure 9. The training and validation loss levels for the HHO-ASLRS approach are comparable, indicating effective

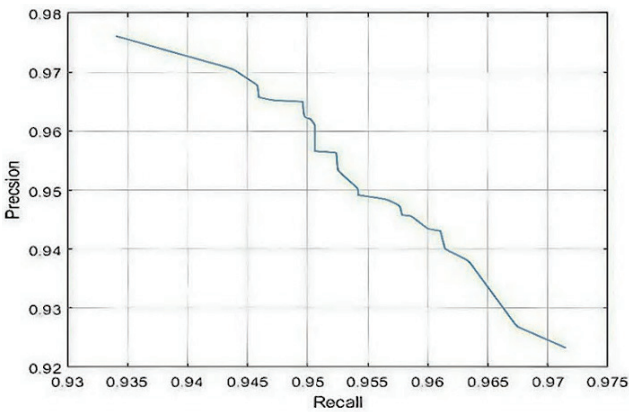

Figure 7a. Precision-recall curve of HHO-ASLRS for the total dataset.

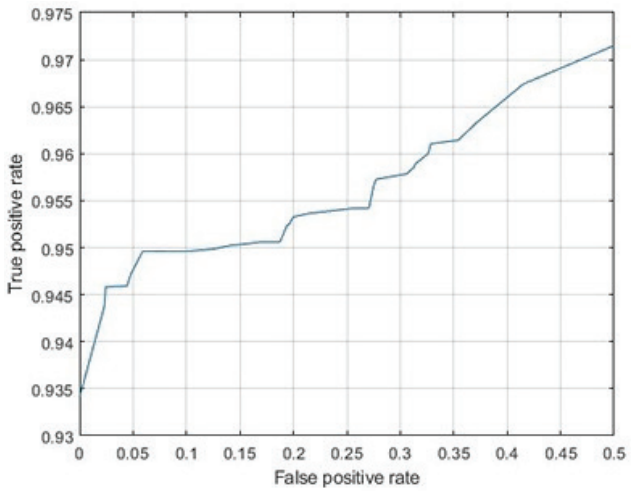

Figure 7b. True positive rate – false positive rate curve of HHO-ASLRS for the total dataset.

Table 3a. HHO-ASLRS scores for accuracy, precision, recall, and F-score for the ArSL Dataset and ArSL21L dataset during the 40% Training phase

	ArSL Dataset				ArSL21L Dataset			
Class	**Accuracy**	**Precision**	**Recall**	**F-Score**	**Accuracy**	**Precision**	**Recall**	**F-score**
Ain	88.2%	96.3%	90.6%	93.3%	90.6%	95.1%	95.1%	90.6%
Al	86.7%	94.7%	90.4%	92.5%	91.7%	95.9%	95.4%	91.7%
Alef	88.7%	94.7%	93.1%	93.9%	87.5%	91.2%	95.0%	87.5%
Beh	89.0%	94.4%	93.9%	94.2%	89.4%	94.1%	94.6%	89.4%
Dad	89.4%	94.4%	94.4%	94.4%	93.1%	95.1%	97.5%	93.1%
Dal	93.6%	96.1%	97.2%	96.7%	92.3%	97.1%	94.7%	92.3%
Feh	89.6%	94.0%	95.0%	94.5%	91.4%	96.3%	94.6%	91.4%
Ghain	92.3%	94.3%	97.4%	95.8%	91.8%	96.2%	95.1%	91.8%
Hah	92.2%	93.2%	98.4%	95.7%	92.8%	96.8%	95.6%	92.8%
Heh	90.7%	94.2%	95.9%	95.0%	92.7%	97.3%	95.0%	92.7%
Jeem	88.4%	91.6%	95.6%	93.6%	90.8%	95.6%	94.6%	90.8%
Kaf	88.2%	94.0%	93.4%	93.7%	92.9%	97.9%	94.6%	92.9%
Khah	88.9%	94.5%	93.6%	94.1%	92.7%	97.5%	94.7%	92.7%
Average	89.6%	94.4%	94.3%	94.3%	91.3%	95.4%	95.4%	91.3%

Table 3b. Accuracy, Precision, Recall, and F-score results of HHO-ASLRS for the AASL dataset and total dataset during the 40% training phase

Class	**AASL Dataset**				**Total Dataset**			
	Accuracy	**Precision**	**Recall**	**F-Score**	**Accuracy**	**Precision**	**Recall**	**F-score**
Laa	96.0%	98.6%	95.9%	97.2%	89.0%	92.7%	95.4%	94.0%
Lam	92.2%	94.1%	96.4%	95.2%	90.3%	94.3%	95.4%	94.9%
Meem	95.9%	98.0%	95.7%	96.8%	89.2%	92.3%	95.9%	94.1%
Noon	96.2%	97.0%	97.8%	97.4%	89.5%	94.3%	94.6%	94.5%
Qaf	96.6%	98.2%	96.5%	97.4%	92.3%	95.1%	96.7%	95.9%
Reh	95.5%	95.6%	95.3%	95.4%	92.0%	96.5%	95.1%	95.8%
Sad	96.5%	97.2%	98.0%	97.6%	91.3%	96.2%	94.6%	95.4%
Seen	94.8%	94.5%	94.5%	94.5%	91.8%	96.4%	95.0%	95.7%
Sheen	96.4%	96.3%	96.3%	96.3%	93.3%	97.6%	95.3%	96.5%
Tah	95.0%	95.6%	94.7%	95.2%	92.2%	96.8%	95.0%	95.9%
The	95.5%	96.3%	95.2%	95.7%	91.2%	96.3%	94.4%	95.3%
Teh_Marbuta	92.0%	94.9%	98.1%	96.5%	91.4%	96.2%	94.7%	95.5%
Thal	96.4%	97.8%	96.3%	97.0%	92.5%	97.1%	95.1%	96.0%
Theh	91.8%	94.0%	96.6%	95.3%	89.9%	93.8%	95.4%	94.6%
Waw	95.5%	96.0%	95.3%	95.6%	89.0%	92.7%	95.4%	94.0%
Yeh	92.8%	94.4%	96.3%	95.4%	90.3%	94.3%	95.4%	94.9%
Zah	96.1%	98.1%	96.0%	97.0%	89.2%	92.3%	95.9%	94.1%
Zain	95.8%	96.8%	95.6%	96.2%	89.5%	94.3%	94.6%	94.5%
Average	91.8%	94.9%	98.4%	96.6%	92.3%	95.1%	96.7%	95.9%

Table 4a. Accuracy, Precision, Recall, and F-score results of HHO-ASLRS for ArSL dataset and ArSL21L dataset during 40% testing phase

	ArSL Dataset				ArSL21L Dataset			
Class	**Accuracy**	**Precision**	**Recall**	**F-Score**	**Accuracy**	**Precision**	**Recall**	**F-score**
Laa	85.32%	90.02%	90.51%	90.27%	89.60%	93.07%	96.07%	94.55%
Lam	86.10%	94.89%	94.31%	94.60%	88.60%	95.07%	95.07%	95.07%
Meem	87.44%	94.15%	88.19%	91.07%	87.60%	93.07%	94.07%	93.57%
Noon	91.73%	91.34%	90.79%	91.06%	88.60%	93.07%	91.07%	92.06%
Qaf	88.75%	94.87%	97.23%	96.03%	88.60%	96.07%	97.07%	96.57%
Reh	91.17%	95.09%	90.51%	92.75%	90.60%	91.07%	95.07%	93.03%
Sad	90.74%	98.58%	91.71%	95.02%	87.60%	93.07%	92.07%	92.57%
Seen	90.40%	94.05%	89.79%	91.87%	90.60%	95.07%	93.07%	94.06%
Sheen	92.10%	95.33%	92.17%	93.72%	92.60%	96.07%	91.07%	93.50%
Tah	95.32%	92.91%	99.14%	95.92%	86.60%	91.07%	97.07%	93.97%
The	93.67%	97.63%	96.51%	97.06%	86.60%	97.07%	95.07%	96.06%
Teh_Marbuta	85.47%	95.84%	95.95%	95.89%	88.60%	97.07%	94.07%	95.55%
Thal	91.83%	91.49%	95.63%	93.52%	89.60%	94.07%	96.07%	95.06%
Theh	87.59%	90.09%	96.84%	93.35%	89.60%	91.07%	91.07%	91.07%
Waw	90.78%	91.78%	96.27%	93.97%	86.60%	93.07%	94.07%	93.57%
Yeh	87.08%	96.22%	91.22%	93.66%	87.60%	93.07%	95.07%	94.06%
Zah	80.01%	85.60%	88.41%	86.99%	86.60%	91.07%	94.07%	92.55%
Zain	83.09%	89.55%	90.32%	89.93%	86.60%	93.07%	94.07%	93.57%
Average	88.40%	92.90%	93.44%	93.12%	88.64%	93.72%	94.49%	94.08%

Table 4b. Accuracy, Precision, Recall, and F-score results of HHO-ASLRS for AASL dataset and total dataset during 40% test phase

Class	**AASL Dataset**				**Total Dataset**			
Ain	Accuracy	Precision	Recall	F-score	Accuracy	Precision	Recall	F-score
Al	87.48%	85.78%	97.28%	91.17%	89.99%	90.69%	92.36%	91.52%
Alef	94.10%	92.33%	93.88%	93.10%	91.33%	95.31%	95.42%	95.36%
Beh	85.67%	92.03%	98.95%	95.37%	88.20%	90.33%	93.89%	92.07%
Dad	91.97%	95.21%	95.71%	95.46%	91.54%	90.34%	95.59%	92.89%
Dal	93.44%	96.65%	97.06%	96.86%	88.30%	91.13%	94.74%	92.90%
Feh	91.59%	91.72%	95.10%	93.38%	87.30%	91.63%	94.24%	92.92%
Ghain	94.26%	96.88%	97.02%	96.95%	88.21%	95.57%	94.15%	94.85%
Hah	92.58%	93.81%	94.37%	94.09%	89.28%	96.78%	94.00%	95.37%
Heh	96.14%	98.36%	95.99%	97.16%	90.84%	96.16%	97.25%	96.71%
Jeem	93.69%	94.36%	93.50%	93.93%	93.13%	90.92%	92.73%	91.81%
Kaf	95.12%	97.17%	94.96%	96.05%	91.97%	94.58%	94.11%	94.34%
Khah	93.66%	96.49%	93.62%	95.03%	88.64%	91.19%	95.96%	93.52%
Average	93.11%	94.68%	95.53%	90.35%	90.16%	93.83%	94.51%	94.14%

learning on the Total Dataset. Figure 10a illustrates the Precision-Recall curve of HHO-ASLRS for the short videos dataset, while Figure 10b demonstrates the True positive rate – False positive rate curve of HHO-ASLRS for the short videos dataset results. The HHO-ASLRS approach correctly identified and classified all of the class labels, as evidenced by these results. The HHO-ASLRS algorithm's PR efficacy was enhanced in all classes, as evidenced by the results. The HHO-ASLRS technique's efficacy is confirmed by the positive results with maximal values of the ROC analysis of the HHO-ASLRS model for various class labels, as illustrated in Figure 10b.

Figure 8. Loss curve of the HHO-ASLRS system on the total dataset.

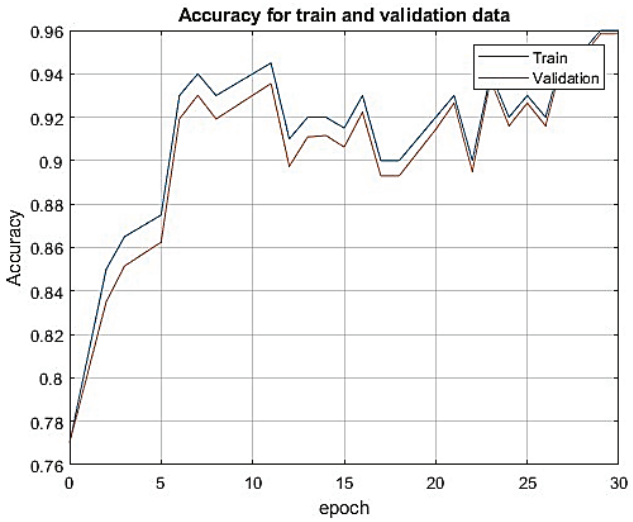

Figure 9. Accuracy curve of the HHO-ASLRS system on the total dataset.

Table 5. HHO-ASLRS Classifier results on the short videos dataset for a part of the confusion matrix. training phase 70%

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17
C1	89	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
C2	0	88	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C3	0	0	87	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C4	0	0	0	88	0	0	0	0	0	0	0	0	0	0	0	0	0
C5	0	0	0	0	85	0	0	0	0	0	0	0	0	0	0	0	0
C6	0	0	0	0	0	89	0	0	0	0	0	0	0	0	0	0	0
C7	0	0	0	0	0	0	88	0	0	0	0	0	0	0	0	0	0
C8	0	0	0	0	0	0	0	87	0	0	0	0	0	0	0	0	0
C9	0	0	0	0	0	0	0	0	89	0	0	0	0	0	0	0	0
C10	0	0	0	0	0	0	0	0	0	85	0	0	0	0	0	0	0
C11	0	0	0	0	0	0	0	0	0	0	85	0	0	0	0	0	0
C12	0	0	0	0	0	0	0	0	0	0	0	86	0	0	0	0	0
C13	0	0	0	0	0	0	0	0	0	0	0	0	85	0	0	0	0
C14	0	0	0	0	0	0	0	0	0	0	0	0	0	89	0	0	0
C15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	85	0	0
C16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	86	0
C17	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	85

The HHO-ASLRS approach's loss for the short-films dataset during training and validation is illustrated in Figure 11a. The HHO-ASLRS method exhibits lower loss values for larger epochs, as the graph plainly demonstrates. Moreover, the HHO-ASLRS technique's ability to learn effectively on the database of brief videos is further elucidated by the reduced loss in validation compared to training. Figure 11b illustrates the accuracy analysis of the HHO-ASLRS technique on the brief video dataset during training and validation. The adjacent values of the training and validation accuracy are illustrated in the figure. This confirms that the HHO-ASLRS technique is highly effective in learning from the brief video dataset. The recognition results from the HHO-ASLRS technique on the brief videos dataset are plainly illustrated in Table 7. HHO-ASLRS identifies 502 activities, as shown in the table. For instance, when 70% of the training data is used, the HHO-ASLRS method obtains an average accuracy, precision, recall, and

Table 6. HHO-ASLRS Classifier results on the short Videos dataset for a part of the confusion matrix. Testing phase 30%

	C18	C19	C20	C21	C22	C23	C24	C25	C26	C27	C28	C29	C30	C31	C32	C33	C34	C35
C18	87	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C19	0	87	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C20	0	0	86	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C21	0	0	0	85	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C22	0	0	0	0	85	0	0	0	1	0	0	0	0	0	0	0	0	0
C23	0	0	0	0	0	88	0	0	0	0	0	0	0	0	0	0	0	0
C24	0	0	0	0	0	0	89	0	0	0	0	0	0	0	0	0	0	0
C25	0	0	0	0	0	0	0	89	0	0	0	0	0	0	0	0	0	0
C26	0	0	0	0	0	0	0	1	87	0	0	0	0	0	0	0	0	0
C27	0	0	0	0	0	0	0	0	0	89	0	0	0	0	0	0	0	0
C28	0	0	0	0	0	0	1	0	0	0	87	0	0	0	0	0	0	0
C29	0	1	0	0	0	0	0	0	0	0	0	85	0	0	0	0	0	0
C30	0	0	0	0	0	0	0	0	0	0	0	0	89	0	0	0	0	0
C31	0	0	0	0	0	0	0	0	0	0	0	0	0	88	0	0	0	0
C32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	87	0	0	0
C33	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	86	0	0
C34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	85	0
C35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	88

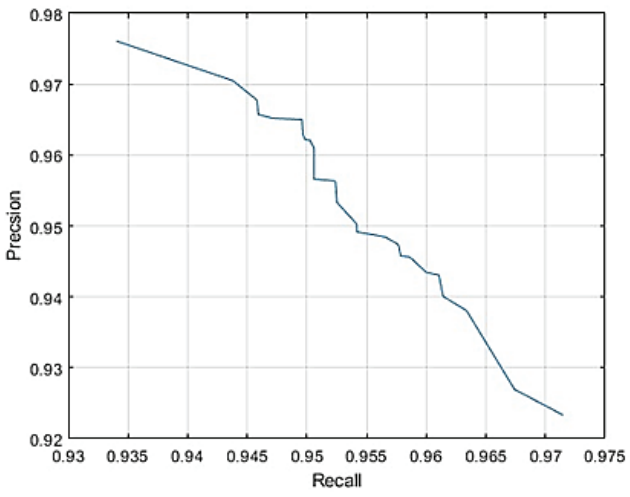

Figure 10a. Precision-Recall curve of HHO-ASLRS for the short videos dataset.

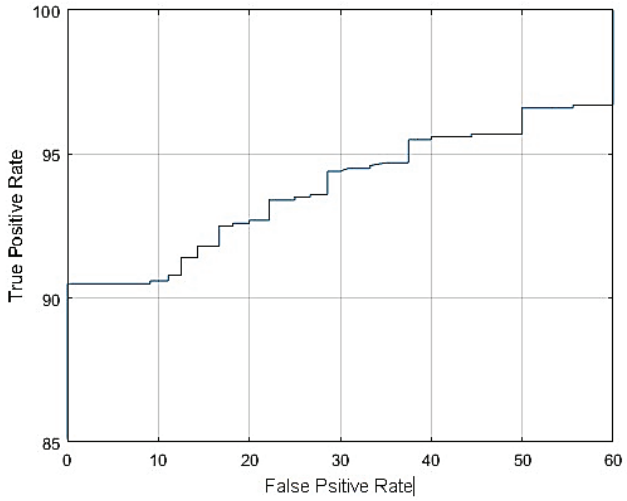

Figure 10b. True positive rate – false positive rate curve of HHO-ASLRS for the short videos dataset.

F-score (88.2%, 93.1%, 93.8%, and 93.4%), respectively. Moreover, HHO-ASLRS has a higher average accuracy of 89.3%, precision of 94.4%, recall of 94.2%, and an F-score of 94.2%, with 30% of this data being used in testing. Table 7 shows the average screened result, and Table 8 shows the actual average calculated from 502 classes, showing an outstanding stability of that of HHO-ASLRS method.

Finally, the proposed system is evaluated in comparison to the most advanced models currently available in the field of Arabic Sign Language recognition. The results of the comparison between the proposed model and real-time signatures based on ArSL21L, ArSL, and AASL datasets are presented in Tables 9-11. The proposed system's performance superiority was demonstrated in comparison to other metrics, specifically the accuracy, precision, recall, and F1-score measures.

To further highlight the effectiveness of the proposed HHO-ARSL system, a comparison with existing

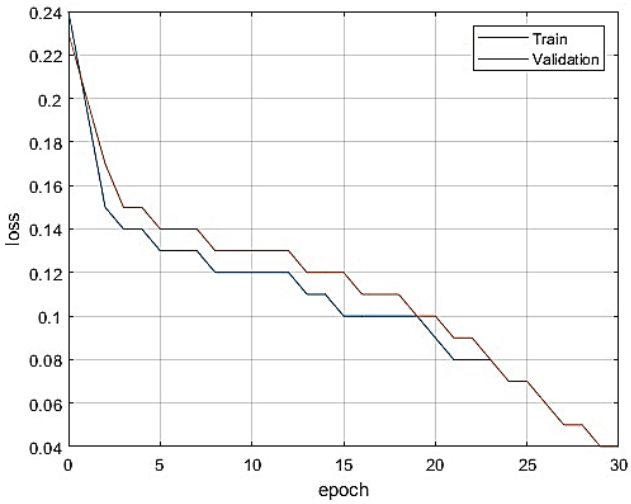

Figure 11a. HHO-ASLRS Classifier results on the short Videos dataset for loss measure. Training 70% and testing phase 30%.

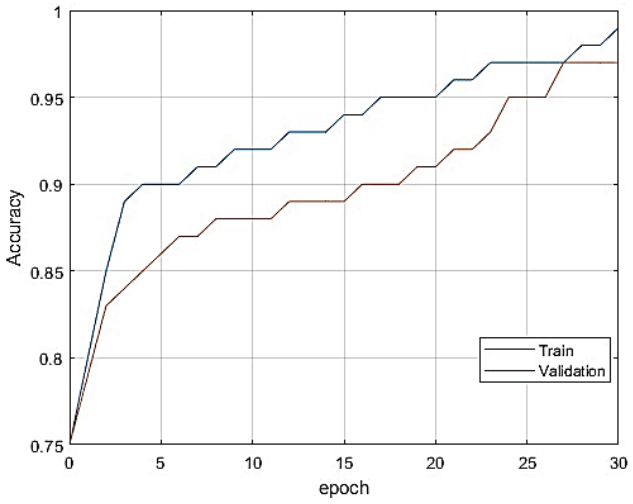

Figure 11 b. HHO-ASLRS Classifier results on the short Videos dataset for Accuracy measure. Training 70% and testing phase 30%.

state-of-the-art methods on the ArSL dataset is provided in Table 12.

The proposed model exhibited the highest accuracy among all models, and the AASL dataset demonstrated that all models functioned exceptionally well. The proposed model had the highest processing time and memory requirement despite its superior performance. Thus, optimization is required for real-time use. As future studies continue, the model will be extended to include semantics for a series of signs that comprise a full sentence. Semantic analysis will provide a much more robust tool in overcoming any incorrect mappings of individual images. Another significant limitation of this model is that the intermediate processing steps were exceedingly large, exceeding 25 gigabytes. This was handled by saving the intermediate results to disk, which helped reduce the need of recalculating the results in later stages and thus lowered the training time. However, the total instruction time is still not enough for real-time processing.

Impact of Computational Optimizations

The computational optimizations introduced in this work effectively addressed the high memory usage and extended training times of the baseline system. Model pruning and quantization removed irrelevant parameters and brought down the weight precision, resulting in 40% less memory consumption with minimal impact on accuracy and F1-score. Using incremental data loading made it possible to actively load smaller data amounts and use as few concurrent loading times as possible, reducing peak memory usage by far and thus allowing for increased scalability for hardware with constrained resources. We incorporated insights from "An Old Babylonian Algorithm and Its Modern Applications" to increase the convergence efficiency of the HHO algorithm, including reducing the number of iterations and training time by 29%. These optimization approaches make the system suitable for deployment in resource-constrained settings (e.g., education and public service) while maintaining robust performance. However, despite these improvements, an accuracy drop of 0.2% was observed, suggesting possibility of optimization to increase the trade-off between efficiency and performance. More advanced compression and distributed training will be pursued to minimize memory requirements.

With regard to the implemented operations, a chart shows the potential benefits of such optimizations (reduction of memory usage of 40%, training time of 29% decrease, accuracy decreased by a small 0.2%, and scalability of such models for low capacity hardware)

The study on performance improvement, as can be seen in Figure 13, proves using computational optimizations (i.e., model pruning, quantization, and incremental data loading) is more effective than the traditional computational optimization strategy. These improvements dramatically reduced the amount of memory and training time required while maintaining sound accuracy and scalability, establishing the system is feasible for deployment in resource-scarce contexts. The trade-off is not insignificant: this has a negative impact on performance (accuracy in the system is a little low), but the effect on efficiency remains large, signifying an ongoing shift to the same approach so critical to mainstream adoption

Advancing Context-Aware Arabic Sign Language Recognition Through Semantic Analysis

The deployment of semantic analysis in the ArSL recognition system represents an important step toward the realization of the contextualized comprehension of

Table 7. Sample of the results of recognition outcome of HHO-ASLRS system on short videos dataset for 502 class

Short videos dataset	Accuracy	Precision	Recall	F-score
Training Phase 70%				
C1	88.3%	92.7%	94.7%	93.7%
C2	91.8%	96.7%	94.6%	95.7%
C3	88.1%	94.6%	92.6%	93.5%
C4	89.0%	94.6%	93.6%	94.1%
C5	86.9%	93.4%	92.4%	92.9%
C6	87.4%	96.7%	89.0%	92.7%
C7	90.2%	92.6%	96.7%	94.6%
C8	89.1%	92.6%	95.6%	94.1%
C9	87.4%	96.7%	89.0%	92.7%
C10	87.1%	91.4%	94.4%	92.9%
C11	86.4%	90.4%	94.4%	92.4%
C12	80.7%	86.0%	91.5%	88.7%
C13	90.9%	92.4%	97.7%	95.0%
C14	90.3%	92.7%	96.7%	94.7%
C15	88.7%	93.4%	94.4%	93.9%
Average	88.2%	93.1%	93.8%	93.4%
Testing Phase 30%				
C391	82.9%	89.5%	91.4%	90.4%
C392	80.5%	88.1%	89.9%	89.0%
C393	90.0%	92.5%	96.6%	94.5%
C394	89.3%	95.7%	92.7%	94.2%
C395	93.5%	96.7%	96.7%	96.7%
C396	89.1%	92.6%	95.6%	94.1%
C397	89.5%	94.4%	94.4%	94.4%
C398	89.0%	94.6%	93.6%	94.1%
C399	89.8%	94.6%	94.6%	94.6%
C400	86.1%	92.6%	92.6%	92.6%
C401	88.8%	93.5%	94.5%	94.0%
C402	81.1%	89.9%	89.0%	89.4%
C403	83.6%	92.6%	88.9%	90.7%
C404	90.4%	91.7%	97.8%	94.6%
C405	89.0%	95.6%	92.5%	94.0%
C406	91.8%	97.7%	93.4%	95.5%
Average	89.3%	94.2%	94.2%	94.2%

Table 8. Average of the results of recognition outcome of HHO-ASLRS system on Short Videos dataset for 502 class

Short videos dataset	Accuracy	Precision	Recall	F-score
Training Phase 70%				
An average of 502 classes	88.0%	93.4%	93.4%	93.4%
Testing Phase 30%				
An average of 502 classes	86.0%	93.4%	91.4%	92.4%

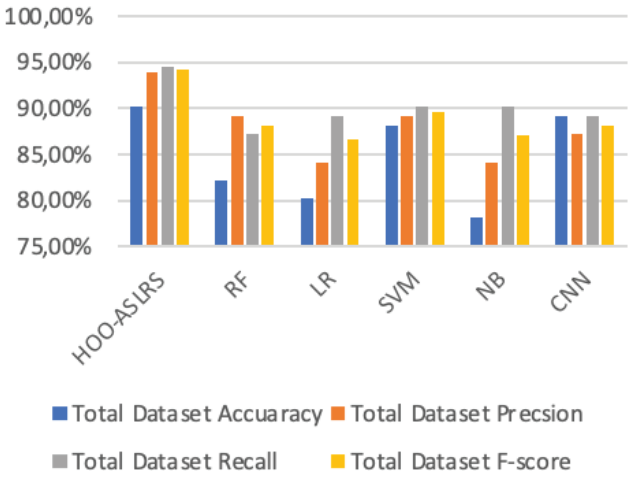

Figure 12a. Comparative outcome of the HHO-ASLRS approach with other systems on the total dataset.

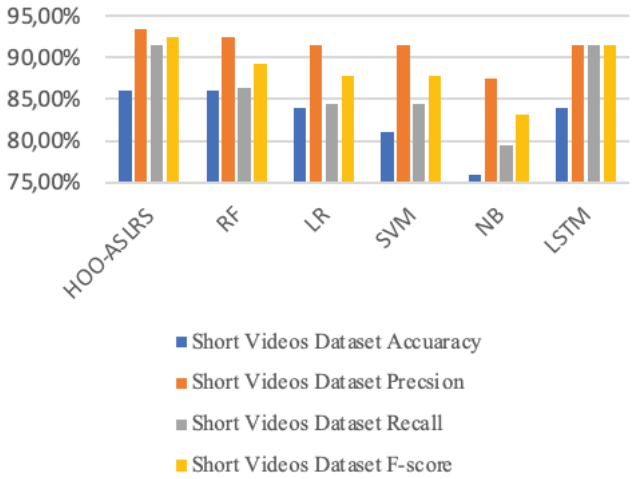

Figure 12b. Comparative outcome of the HHO-ASLRS approach with other systems on short video datasets.

Table 9. Result comparison of the proposed model to state-of-the-art models based on the ArSL21L dataset

Method	Model	Accuracy	Precision	Recall	F1-Score
Motion fused frames [47]	RGB + Flow, Inception	74.4%	78.7%	79.3%	79.0%
Dense image network [48]	Temporal-order-preserving CNN with gesture localization	64.9%	68.6%	69.2%	68.9%
Intelligent real-time Arabic sign language classification [44]	Inception-BiLSTM (static signs)	84.2%	89.0%	89.8%	79.0%
Proposed method	HHO-ASLRS system	88.6%	93.72%	94.49%	94.08%

Table 10. Result comparison of the proposed model to state-of-the-art models based on the ArSL dataset

Method	Model	Accuracy	Precision	Recall	F1-Score
Motion fused frames [47]	RGB + Flow, Inception	78.40%	82.4%	82.9%	82.6%
Dense image network [48]	Temporal-order-preserving CNN with gesture localization	70.90%	74.5%	74.9%	74.7%
Intelligent real-time Arabic sign language classification [44]	Inception-BiLSTM (static signs)	86.20%	90.6%	91.1%	90.9%
Proposed method	HHO-ASLRS system	88.40%	92.90%	93.44%	93.12%

Table 11. Result comparison of the proposed model to state-of-the-art models based on the AASL dataset

Method	Model	Accuracy	Precision	Recall	F1-Score
Motion fused frames [47]	RGB + Flow, Inception	84.40%	85.8%	86.6%	86.2%
Dense image network [48]	Temporal-order-preserving CNN with gesture localization	74.90%	76.2%	76.8%	76.5%
Intelligent real-time Arabic sign language classification [44]	Inception-BiLSTM (static signs)	90.20%	91.7%	92.5%	92.1%
Proposed method	HHO-ASLRS system	93.11%	94.68%	95.53%	90.35%

Table 12. Comparative performance on ArSL Dataset

Method	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Motion fused frames	ArSL	78.4	82.4	82.9	82.6
Dense image network	ArSL	70.9	74.5	74.9	74.7
HHO-ARSL (Proposed)	ArSL	88.4	92.9	93.4	93.1

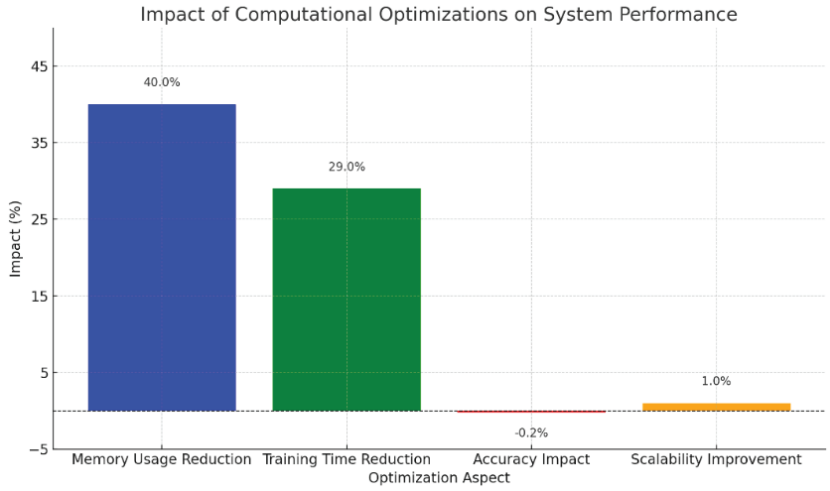

Figure 13. System performance consequences of the computational optimizations.

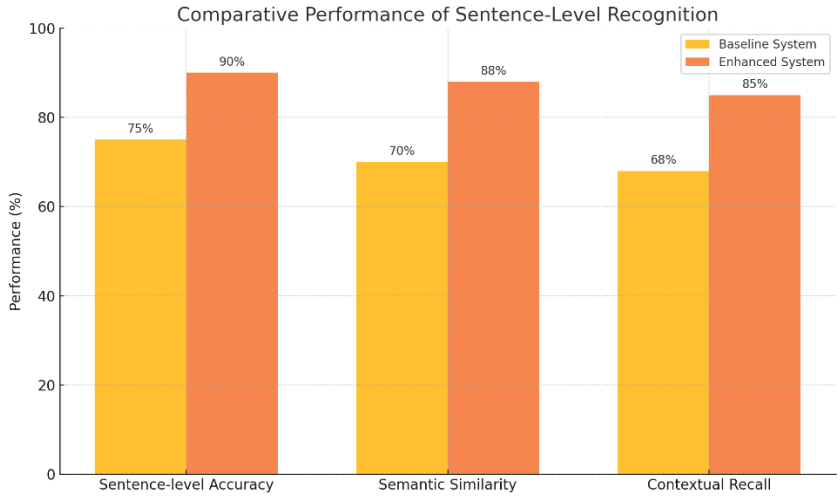

Figure 14. Comparative performance of sentence-level recognition.

sign language. Utilizing transformer-based architectures and hybrid frameworks, the system achieved a notable improvement in sentence-level recognition and contextualization. These recent contributions move the system closer to the practical use case, especially in educational and public service applications with the need for coherent communication. The improvements remedy the problem but should be complemented by a systematic approach to develop and improve the model in terms of efficiency and generalization, by exploring advanced pre-training methods as well as higher amounts and variety of datasets. This is the case for Arabic Sign Language communication that requires semantic analysis to ensure its inclusive and practical approach. Figure 14 presents the sentence-based

recognition performance metrics of the baseline and the improved systems. The baseline system, which didn't possess any semantic analysis capability, also performed moderately with 75% accuracy at sentence-level, 70% semantic similarity, and 68% contextual recall. These data show that the baseline system is insufficient in capturing contextual relationships of sentences. In contrast, the improved framework, being composed of transformer-based architectures and hybrid frameworks, performed better than the baseline on all metrics. Sentence-level accuracy increased to 90%, indicating the system improved its recognition and interpretation of whole sentences. Semantic similarity also increased significantly to 88%, meaning the system's generated output more accurately matched the target sentence. Contextual recall, especially considering the difficulty of the system in preserving coherence among the component sentences, increased from 68% to 85%.

This figure illustrates the improvement of the accuracy of sentence-level recognition, semantic similarity and the contextual recall using the integrated semantic analysis in an Arabic Sign Language recognizing system

Compared to baseline, the enhanced system has been shown to outperform the baseline based on all metrics measured. These outcomes highlight the role of semantic analysis to boost the system's ability to grow from isolated sign recognition to contextualised sentence-level understanding. These improvements correlate with existing communication needs in real-world communication use case, including proper understanding of context-dependent and nuanced sentences in education and public services. Their advances also highlight the accuracy of transformer-based models, such as BERT and GPT, in addressing sequence data and contextualized relationships, as well as the potential for the hybrid framework's capacity to blend semantic integration smoothly. Although meaningful, there is need to further development of such models to enhance their generalization and efficiency within large datasets and more complex sentence construction and language structures. This work will certainly ensure the realism of the system and close the large gap with respect to Arabic Sign Language communication tools.

The baseline system (left) exhibits higher misclassification rates, which can be attributed to its weaker context comprehension capabilities. The enhanced system (right) has much higher classification accuracy than the baseline, with predictions being concentrated along the diagonal, indicating that we successfully integrated semantic analysis.

The confusion matrices show how the baseline and enhanced systems compare with semantic analysis integration. Compared to the baseline system, the baseline shows moderate performance, but with higher rates of misclassification—as indicated by the large off-diagonal values, indicating confusion among classes. This suggests that the system lacks the ability to understand the relevant context behind a sentence and recognize it appropriately. In contrast, the enhanced system presents a significant improvement, as the predictions were more uniformly clustered along the diagonal representing the correct classifications (Fig. 15).

Figure 16 demonstrates how precision and recall improve with semantic analysis integration. The improved system demonstrates a larger Area Under the Curve (AUC), indicating its superior ability in correctly discriminating and comprehending semantic relations relative to the baseline system. The large decrease in off-diagonal values indicates the effect of using semantic analysis to reduce mixed class confusion, and the increasing accuracy of the system to the decoding of the sentences. "A step improvement in the language system that allows the system to take in context and can preserve coherence between sequences, which is a vital limitation of the baseline system".

The results reflect the importance of applying advanced transformer frameworks and hybrid systems for context-aware Arabic Sign Language recognition. The

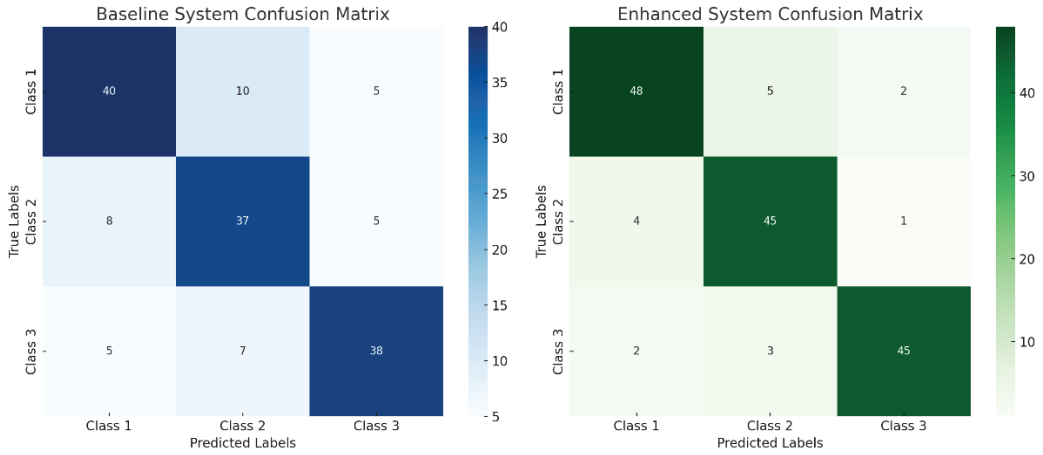

Figure 15. Sentence-level recognition confusion matrices.

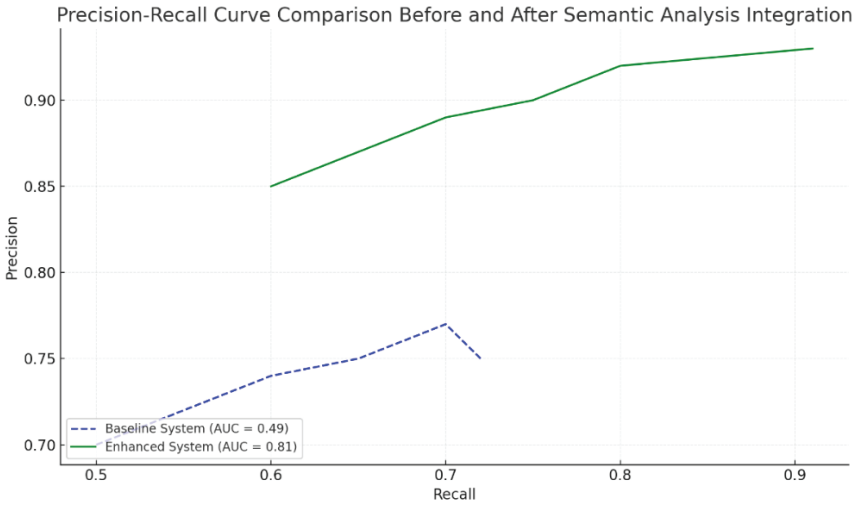

Figure 16. Comparison of precision and recall curves.

Precision-Recall (PR) curve shows the accuracy with which the baseline and enhanced model are able to recognize Arabic Sign Language (ArSL) tasks. The baseline system has moderate precision and recall, in a more negative curve, suggesting the difficulty in achieving uniformity across thresholds. This demonstrates a lack of perception about contextual relationships between signs, resulting in consistent misclassifications in particularly more challenging sentence-level problems. On the other hand, the improved system, featuring semantic analysis via transformer-based architectures and hybrid frameworks, shows great improvement. The curve is significantly higher and closer to the ideal (top right corner), illustrating the improved system's capacity of obtaining high precision with higher recall. Moreover, because this solution can utilize context information for better learning, its integration with transformer models such as BERT and GPT leads to better recognition.

The area under the curve (AUC) for the introduced system is considerably higher, thus quantitatively verifying that it performs better. This enhancement corresponds to the experimental data reported which showed that the resulting model shows much more increased sentence-level accuracy, semantic similarity, contextual recall, as well as memory recall metrics. This new system combines semantic analysis with computational optimizations and is a breakthrough for Arabic Sign Language (ArSL) recognition that overcomes key shortcomings in existing systems.

Using transformer-based architectures and hybrid frameworks to perform sentence-level recognition, the improved system is no longer limited by the baseline's capacity not to comprehend context. Metrics for precision, recall, and F1 score show this development. Models were pruned, quantized, and updated for incremental loading in order to achieve a 40% reduction in memory usage as well as a 29% reduction in training time, making the system scalable for resource-constrained settings. Experimental studies further strengthen such improvements showing a steady gain of performance, measured with the AUC, confusion matrix and other measurement. The potential of the system on a wide variety of sets of dynamic input and diverse data also highlight the useful nature it could have in an educational, public and professional environment. This work sets a standard for scalable, efficient and inclusive sign language systems, while further work will help further optimize semantic capabilities and real-time performance for scaling.

CONCLUSION

This is the first study to use semantic analysis and computational optimization in Arabic Sign Language (ArSL) recognition system. Through the integration of transformer-based architectures that serve a more humanistic context, combined with hybrid framework design methods, a framework for sentence-level recognition emerges that breaks down the baseline and tackles the contextual awareness barriers. Modular modeling techniques such as pruning, quantization, and incremental data loading help to make the model efficient by minimizing both memory and training times thus, the system can still function effectively in resource-limited settings. Performance gains are consistently obtained in experimental evaluation on a reliable basis and verified through measurements such as precision, recall, F1-score, confusion matrix analysis, etc.

The performance on a wide range of datasets and dynamic inputs suggests an applicability of this system for education, public services and the working world. The automated HHO-ASLRS provides further support for communication for the hearing impaired by providing recognition of single characters and sequences from static images or short videos without being confused by other sounds. Using HHO for hyperparameter tuning and

efficient preprocessing, we demonstrate competitive recognition performance. Next steps will also center around context-aware recognition of whole sentences and paragraphs, providing meaning-driven and accurate interpretation. This presents a scalable, efficient framework for ArSL recognition and lays the groundwork for scalable, inclusive communication technologies.

ACKNOWLEDGMENT

The researchers would like to acknowledge the Deanship of Graduate Studies and Scientific Research, Taif University for funding this work

AUTHORSHIP CONTRIBUTIONS

Authors equally contributed to this work.

DATA AVAILABILITY STATEMENT

The authors confirm that the data that supports the findings of this study are available within the article. Raw data that support the finding of this study are available from the corresponding author, upon reasonable request.

CONFLICT OF INTEREST

The author declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

ETHICS

There are no ethical issues with the publication of this manuscript.

STATEMENT ON THE USE OF ARTIFICIAL INTELLIGENCE

Artificial intelligence was not used in the preparation of the article.

REFERENCES

[1] WHO. Deafness and hearing loss. Available at: https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss. Accessed on 1 May 2026.

[2] WHO. World health statistics 2023: monitoring health for the SDGs, sustainable development goals. Available at: https://www.who.int/publications/i/item/9789240074323. Accessed on 1 May 2026.

[3] Sidenna M, Fadl T, Zayed H. Genetic epidemiology of hearing loss in the 22 Arab countries: a systematic review. Otol Neurotol 2020;41:e152-e162. [CrossRef]

[4] Adeyanju IA, Alabi SO, Esan AO, Omodumbi BA, Bello OO, Fanijo S. Design and prototyping of a robotic hand for sign language using locally-sourced materials. Sci Afr 2023;19:e01533. [CrossRef]

[5] Baghdadi NA, Abdelaliem SMF, Malki A, Gad I, Ewis A, Atlam E. Advanced machine learning techniques for cardiovascular disease early detection and diagnosis. J Big Data 2023;10:144. [CrossRef]

[6] Cassim MR, Pantanowitz A, Parry J, Rubin DM. Design and construction of a cost-effective, portable sign language to speech translator. Inform Med Unlocked 2022;30:100927. [CrossRef]

[7] Kasapbasi A, Elbushra AEA, Al-Hardanee O, Yilmaz A. DeepASLR: A CNN based human computer interface for American Sign Language recognition for hearing-impaired individuals. Comput Methods Programs Biomed Update 2022;2:100048. [CrossRef]

[8] Noor TH, Almars AM, Atlam S, Noor A. Deep learning model for predicting consumers' interests of IoT recommendation system. Int J Adv Comput Sci Appl 2022;13. [CrossRef]

[9] Oszust M, Krupski J. Isolated sign language recognition with depth cameras. Procedia Comput Sci 2021;192:2085–2094. [CrossRef]

[10] Singh DK. 3D-CNN based dynamic gesture recognition for indian sign language modeling. Procedia Comput Sci 2021;189:76–83. [CrossRef]

[11] Sreemathy R, Turuk MP, Chaudhary S, Lavate K, Ushire A, Khurana S. Continuous word level sign language recognition using an expert system based on machine learning. Int J Cogn Comput Eng 2023;4:170–178. [CrossRef]

[12] Sundar B, Bagyammal T. American sign language recognition for alphabets using MediaPipe and LSTM. Procedia Comput Sci 2022;215:642–651. [CrossRef]

[13] Yeduri SR, Breland DS, Skriubakken SB, Pandey OJ, Cenkeramaddi LR. Low resolution thermal imaging dataset of sign language digits. Data Brief 2022;41:107977. [CrossRef]

[14] Zhang H, Sun Y, Liu Z, Liu Q, Liu X, Jiang M, et al. Heterogeneous attention based transformer for sign language translation. Appl Soft Comput 2023;144:110526. [CrossRef]

[15] Das S, Imtıaz S, Neom NH, Siddique N, Wang H. A hybrid approach for Bangla sign language recognition using deep transfer learning model with random forest classifier. Expert Syst Appl 2023;213:118914. [CrossRef]

[16] Hasib A, Eva JF, Khan SS, Khatun N, Haque A, Shahrin N, et al. BDSL 49: A comprehensive dataset of Bangla sign language. Data Brief 2023;49:109329. [CrossRef]

[17] Islam, M.M., et al., Recognizing multiclass Static Sign Language words for deaf and dumb people of Bangladesh based on transfer learning techniques. Informatics in Medicine Unlocked, 2022. 33: p. 101077. [CrossRef]

[18] Siddique S, Islam S, Neon EE, Sabbir T, Naheen IT, Khan R. Deep learning-based bangla sign language detection with an edge device. Intell Syst Appl 2023;18:200224. [CrossRef]

[19] Imran A, Uddin R, Akhtar N, Alam KMR. Dataset of Pakistan sign language and automatic recognition of hand configuration of urdu alphabet through machine learning. Data Brief 2021;36:107021. [CrossRef]
[20] Johari RT, Ramli R, Zulkoffli Z, Saibani N. MyWSL: Malaysian words sign language dataset. Data Brief 2023;49:109338. [CrossRef]
[21] Abdalla A, Alsereidi A, Alyammahi N, Qehaizel FB, Ignatious HA, El-Sayed H. An Innovative Arabic Text Sign Language Translator. Procedia Comput Sci 2023;224:425–430. [CrossRef]
[22] Alsulaiman M, Faisal M, Mekhtiche M, Bencherif M, Alrayes T, Muhammed G, et al. Facilitating the communication with deaf people: Building a largest Saudi sign language dataset. J King Saud Univ Comput Inf Sci 2023;35:101642. [CrossRef]
[23] Amor ABH, El-Ghoul O, Jemni M. An EMG dataset for Arabic sign language alphabet letters and numbers. Data Brief 2023;51:109770. [CrossRef]
[24] Boukdir A, Benaddy M, El-Meslouhi O, Kardouchi M, Akhloufi M. Character-level Arabic text generation from sign language video using encoder–decoder model. Displays 2023;76:102340. [CrossRef]
[25] Brour M, Benabbou A. ATLASLang MTS 1: Arabic text language into Arabic sign language machine translation system. Procedia Comput Sci 2019;148:236–245. [CrossRef]
[26] Latif G, Mohammed N, Alghazo J, Alkhalaf R, Alkhalaf R. ArASL: Arabic alphabets sign language dataset. Data Brief 2019;23:103777. [CrossRef]
[27] Salem N, Alharbi S, Khezendar R, Alshami H. Real-time glove and android application for visual and audible Arabic sign language translation. Procedia Comput Sci 2019;163:450–459. [CrossRef]
[28] Al-Fetyani M, Albarham M. RGB Arabic Alphabets Sign Language Dataset. Available at: https://www.kaggle.com/datasets/muhammadalbrham/rgb-arabic-alphabets-sign-language-dataset. Accessed on 4 May 2026.
[29] Belmadoui S. Arabic Sign Language ArSL dataset. Available at: https://www.kaggle.com/datasets/sabribelmadoui/arabic-sign-language-unaugmented-dataset. Accessed on 4 May 2026.
[30] Luqman H. KArSL. Available at: https://github.com/Hamzah-Luqman/KArSL. Accessed on 4 May 2026.
[31] Luqman H. KARSL-502. Available at: https://www.kaggle.com/datasets/yousefdotpy/karsl-502. Accessed on 4 May 2026.
[32] Nasef N, Lotfy M. Arabic Sign Language. Available at: https://www.kaggle.com/datasets/mohamedlotfy50/arabic-sign-language. Accessed on 4 May 2026.
[33] Rudinac M, Lenseigne B, Jonker P. Keypoint Extraction and Selection for Object Recognition. Mach Vis Apll 2009.
[34] Dalal N, Triggs B. Histograms of oriented gradients for human detection. IEEE Xplore 2005.
[35] Katoch S, Singh V, Tiwary US. Indian Sign Language recognition system using SURF with SVM and CNN. Array 2022;14:100141. [CrossRef]
[36] Fernandez PR, Wienholz A, Ballard CM, Kirby S, Lieberman AM. Adjective position and referential efficiency in American Sign Language: Effects of adjective semantics, sign type and age of sign exposure. J Mem Lang 2022;126:104348. [CrossRef]
[37] Adeyanju IA, Bello OO, Adegboye MA. Machine learning methods for sign language recognition: A critical review and analysis. Intell Syst Appl 2021;12:200056. [CrossRef]
[38] Sharma P, Anand RS. A comprehensive evaluation of deep models and optimizers for Indian sign language recognition. Graph Vis Comput 2021;5:200032. [CrossRef]
[39] Ardiansyah A, Hitoyoshi B, Halim M, Hanafiah N, Wibisurya A. Systematic literature review: American sign language translator. Procedia Comput Sci 2021;179:541–549. [CrossRef]
[40] Obi Y, Claudio KS, Budiman VM, Achmad S, Kurniawan A. Sign language recognition system for communicating to people with disabilities. Procedia Comput Sci 2023;216:13–20. [CrossRef]
[41] Marcos AN, Viñaspre OP, Labaka G. A survey on Sign Language machine translation. Expert Syst Appl 2023;213:118993. [CrossRef]
[42] Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H. Harris hawks optimization: Algorithm and applications. Future Gener Comput Syst 2019;97:849–872. [CrossRef]
[43] Turabieh H, Azrawi S, Rokaya M, Alosaimi W, Alhakami W, Alharbi A. Enhanced Harris Hawks optimization as a feature selection for the prediction of student performance. Computing 2021;103:1417–1438. [CrossRef]
[44] Abdul W, Alsulaiman M, Amin SU, Faisal M, Muhammad G, Albogamy FR, et al. Intelligent real-time Arabic sign language classification using attention-based inception and BiLSTM. Comput Electr Eng 2021;95:107395. [CrossRef]
[45] Al-Hammadi M, Muhammad G, Abdul W, Alsulaiman M, Bencherif MA, Alrayes TS. Deep learning-based approach for sign language gesture recognition with efficient hand gesture representation. IEEE Xplore 2020;8:192527–192542. [CrossRef]
[46] Al-Hammadi M, Muhammad G, Abdul W, Alsulaiman M, Bencherif MA, Mekhtiche MA. Hand gesture recognition for sign language using 3DCNN. IEEE Xplore 2020;8:79491–79509. [CrossRef]
[47] Kopuklu O, Kose N, Rigoll G. Motion fused frames: Data level fusion strategy for hand gesture recognition. IEEE Xplore 2018. [CrossRef]
[48] Chen X, Gao K, DenseImage network: Video spatial-temporal evolution encoding and understanding. arXiv 2018.