



Article

Smart Home Automation-Based Hand Gesture Recognition Using Feature Fusion and Recurrent Neural Network

Bayan Ibrahim Alabdullah ¹, Hira Ansar ², Naif Al Mudawi ^{3,*} , Abdulwahab Alazeb ³, Abdullah Alshahrani ⁴, Saud S. Alotaibi ⁵  and Ahmad Jalal ^{2,*}

- ¹ Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia; bialabdullah@pnu.edu.sa
- ² Department of Computer Science, Air University, E-9, Islamabad 44000, Pakistan; hiraansar53@gmail.com
- ³ Department of Computer Science, College of Computer Science and Information System, Najran University, Najran 55461, Saudi Arabia; afalazeb@nu.edu.sa
- ⁴ Department of Computer Science and Artificial Intelligence, College of Computer Science and Engineering, University of Jeddah, Jeddah 21589, Saudi Arabia; asalshahrani2@uj.edu.sa
- ⁵ Information Systems Department, Umm Al-Qura University, Makkah 24382, Saudi Arabia; ssotaibi@uqu.edu.sa
- * Correspondence: naalmudawi@nu.edu.sa (N.A.M.); ahmadjalal@mail.au.edu.pk (A.J.)

Abstract: Gestures have been used for nonverbal communication for a long time, but human–computer interaction (HCI) via gestures is becoming more common in the modern era. To obtain a greater recognition rate, the traditional interface comprises various devices, such as gloves, physical controllers, and markers. This study provides a new markerless technique for obtaining gestures without the need for any barriers or pricey hardware. In this paper, dynamic gestures are first converted into frames. The noise is removed, and intensity is adjusted for feature extraction. The hand gesture is first detected through the images, and the skeleton is computed through mathematical computations. From the skeleton, the features are extracted; these features include joint color cloud, neural gas, and directional active model. After that, the features are optimized, and a selective feature set is passed through the classifier recurrent neural network (RNN) to obtain the classification results with higher accuracy. The proposed model is experimentally assessed and trained over three datasets: HaGRI, Egogesture, and Jester. The experimental results for the three datasets provided improved results based on classification, and the proposed system achieved an accuracy of 92.57% over HaGRI, 91.86% over Egogesture, and 91.57% over the Jester dataset, respectively. Also, to check the model liability, the proposed method was tested on the WLASL dataset, attaining 90.43% accuracy. This paper also includes a comparison with other-state-of-the-art methods to compare our model with the standard methods of recognition. Our model presented a higher accuracy rate with a markerless approach to save money and time for classifying the gestures for better interaction.

Keywords: feature fusion; filter; home automation; adaptive median filter; hand detection; deep learning; noise reduction; gesture recognition



Citation: Alabdullah, B.I.; Ansar, H.; Mudawi, N.A.; Alazeb, A.; Alshahrani, A.; Alotaibi, S.S.; Jalal, A. Smart Home Automation-Based Hand Gesture Recognition Using Feature Fusion and Recurrent Neural Network. *Sensors* **2023**, *23*, 7523. <https://doi.org/10.3390/s23177523>

Academic Editor: Henry Leung

Received: 19 June 2023

Revised: 27 July 2023

Accepted: 16 August 2023

Published: 30 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, home automation has emerged as a research topic. Many researchers have started investigating the demand criteria for home automation in different environments. Human–computer interaction (HCI) [1] is considered a more interactive and resourceful method of engaging with different appliances to make the system work. In the conventional approach, different devices like a mouse, keyboard, touch screen, and remote devices are used to fulfill requirements so that users can interact by only using their hands with different home appliances, home healthcare, and home monitoring systems. Usually, changing channels and controlling light on/off switches are more demanding research areas for HCI [2]. Earlier systems were divided into two approaches for interacting

with computers. The first approach is inertial sensor-based and the second approach is vision-based. In the first approach, sensors are built with one or more arrays. They track the position of the hand, the velocity, and acceleration. Then, these motion features are trained and tested for hand gesture recognition. They are used to control home appliances like TV, radio, and lights [3–7]. Despite its high sensitivity, this approach makes it difficult to obtain higher accuracy. This approach demands a proper setup with high-quality sensors. The use of high-quality sensors can attain better results, but they make the system more expensive, and durability issues arise. With the advancement of technology, new sensors are continually being launched in the market [8], the purpose of which is to minimize sensitivity, making them more expensive.

The second approach is vision-based, which reduces the limitations arising from the sensor-based approach [9]. With the help of this sensor, hand gestures are recognized using images. The images consist of RGB and depth. The RGB images are collected using cameras. The cameras are less expensive and easy to set up properly. The RGB image color, shape, orientation, contours, and positions are calculated for hand gesture recognition. The vision-based sensors with depth images gain more dimensions than RGB [10]. For depth, thresholding techniques are either empirical or automated. Empirical techniques include the trial-and-error method, in which the search space is excluded, and the computation cost is a priority for hand localization. In automated solutions, the hand is considered the main focus area for data acquisition [11]. The hand is localized as the closest object in front of the camera's in-depth image.

Vision-based sensors also pose some challenges for researchers, such as light intensity, clutter sensitivity, and skin tone color [12]. Hand localization is a crucial step. For this, the conventional systems are divided into different steps to obtain better accuracy while keeping the challenges in view. First, data acquisition is performed, followed by hand detection. For hand detection, multiple methods are used, including segmentation, tracking, and color-based extractions. The features are extracted using different algorithms. After that, the gesture is recognized. For the given approach, both images and videos are collected [13]. The still images provide static gestures, whereas videos provide dynamic hand gestures, as changes in hand gestures from one frame to another are noticed. Static gestures are still images and require less computation cost [14–17], whereas dynamic gestures contain three-dimensional motion. The movement in dynamic hand gestures becomes a challenging task as the speed varies, and gesture acquisition is difficult due to speed issues. In the literature, static and dynamic gesture recognition has been performed using two different methods: supervised and unsupervised learning. Supervised learning methods include decision trees, random forests, and SVM, whereas unsupervised learning methods include k-means, hidden Markov model, and PCA [18].

In our proposed model, we have used dynamic gestures to challenge our limitations. Our system proved its compatibility. In this paper, the videos are first converted into frames. An adaptive median filter and gamma correction are applied to the images to reduce noise and adjust the light intensity, respectively. Then, the hand is detected using saliency maps. The extracted hand is then available for feature extraction. We have extracted different features while keeping the issues hindered in classification. For this feature, we have chosen three different state-of-the-art algorithms. These features are named the joint color cloud, neural gas, and directional active model. The features are then optimized using an active bee colony algorithm. The optimized features are passed through the RNN. Our accuracies are shown to be better for model designs. The main contributions of our system are as follows:

- The system approach is different from previous systems; it recognizes dynamic gestures with complex backgrounds.
- Hands are detected from both images using two-way detection: first, the skin tone pixels are extracted, and then the saliency map is applied for greater precision.
- Features are collected using different algorithms, like fast marching, neural gas, and the 8-freeman chain model. All the features are extracted with modifications to the

algorithms listed. The features are collected and fused to make a feature fusion for recognition.

- The proposed system uses a deep learning algorithm such as RNN to achieve higher accuracy.

The rest of the sections presented in this article are as follows: Section 2 includes a related study of the existing methods. Section 3 presents the architecture of the proposed system. Section 4 shows the experimental section with system performance evaluations. Section 5 describes the strengths and weaknesses of our proposed system. Section 6 presents the conclusion of the system and future work directions.

2. Literature Review

Multiple methods have been introduced to acquire hand gestures. This section presents the most useful and popular methods. A literature review was conducted to study the research work carried out in particular areas.

2.1. Hand Gesture Recognition via RGB Sensors

In hand gesture recognition systems, many researchers use sensors and cameras to recognize gestures. The RGB videos can be collected using different cameras. Table 1 presents the methods used by researchers for hand gesture recognition using RGB videos.

Table 1. Related studies on hand gesture recognition using RGB sensors.

Authors	Methodology
S. Nagarajan et al. [19]	The proposed system captures the American sign language and filters the images using Canny edge detection. An Edge Orientation Histogram (EOH) for feature extraction was used, and these feature sets were classified by a multiclass SVM classifier; however, some signs were not detected due to hand orientation and gesture similarity.
Mandeep et al. [20]	The hand gesture system used the skin color model and thresholding; the YCbCr segmented the hand region, skin color segmentation was used to extract the skin pixels, and Otsu thresholding removed the image's background. In the last PCA, the template-matching method was used to recognize a total of twenty images per gesture from five different poses from four gesture captures. On the other hand, this system has some limitations in that skin color varies due to light colors, and the background contains skin color pixels.
Thanh et al. [21]	Multimodal streams are used to increase the performance of hand recognition by combining depth, RGB, and optical flow. A deep learning model is used for feature extraction from each stream; afterward, these features are combined with different fusion methods for the final classification. This system outperforms the results with multi-modal streams of different viewpoints collected from twelve gestures.
Noorkholis et al. [22]	In dynamic hand gesture recognition, the dataset of RGB and depth images is preprocessed from the Region of Interest (ROI) to extract the original pixel value of the hand instead of other unnecessary points. To extract the feature set, a three-dimensional convolutional neural network (3DNN) and long short-term memory (LSTM) combination of deep learning is used to extract the spatio-temporal features that are further classified by finite state machine (FSM) model classification to solve the problem of different gestures used in different applications for ease. This proposed system is designed for a smart TV environment, and for this purpose, eight gestures perform robustly in real-time testing out of 24 gestures.
K. Irie et al. [23]	In this paper, the hand gesture is detected by the motion of the hand in front of the camera. The hand motion is detected to control the electronic appliances in intelligent rooms with complete control of hand gestures. The cameras have the ability to zoom in and focus on the user to detect the hand gesture. The hand is detected via color information and motion direction using fingers.

Table 1. *Cont.*

Authors	Methodology
Chen-Chiung Hsieh et al. [24]	This research was conducted to reduce issues like hand gesture detection from complex backgrounds and light intensity issues. The hand gesture was detected with the help of the body skin detection method. The gestures were classified with the help of a new hand gesture recognition model called the motion history image-based method. A total of six hand gestures at different distances from the camera were used as the dataset. The images were trained using a haar-like structure with up, down, right, and left movements. The home automation-based system generated 94.1% accuracy using the proposed method.
Zhou Ren et al. [25]	A new study was conducted on hand gesture recognition using the finger earth mover distance (FEMD) approach. They noticed the speed and accuracy of the FEMD, shape context, and shape-matching algorithm. The dataset was collected from the Kinect camera, so it contained both depth and RGB images.
Jaya Prakash Saho [26]	Currently, convolutional neural networks (CNNs) exhibit good recognition rates for image classification problems. It is difficult to train deep CNN networks such as AlexNet, VGG-16, and ResNet from scratch due to the lack of big, labelled picture examples in static hand gesture images. To recognize hand gestures in a dataset with a low number of gesture images, they used an end-to-end fine-tuning strategy for a pre-trained CNN model with score-level fusion. They used two benchmark datasets, and the efficacy of the proposed approach was assessed using leave-one-subject-out cross-validation (LOO CV) and conventional CV tests. They proposed a real-time American Sign Language (ASL) recognition system and also evaluated it.
Ing-Jr Ding [27]	In the proposed system, the suggested method consists of two sequential computation steps: phase 1 and phase 2. The deep learning model, a visual geometry group (VGG)-type convolutional neural network (CNN), also known as the VGG-CNN, is used to assess the recognition rate. The experiments proved that image extraction efficiently eliminates the undesirable shadow region in hand gesture depth pictures and greatly improves the identification accuracy.
Jun Li [28]	They proposed MFFCNN-LSTM for forearm sEMG signal recognition using time-domain and time-frequency spectrum features. They first extracted hand movements from the NinaPro db8 dataset, and then images were denoised via empirical Fourier decomposition. The images were passed through the different channels using CNN to collect the time-domain and time-frequency-spectrum features. The features were fused and passed to the LSTM. They achieved 98.5% accuracy with the proposed system.

2.2. Hand Gesture Recognition via Marker Sensors

Many researchers worked on marker sensors with proper equipment setup. Gloves were attached to the hands to note down the locations and movements. Table 2 presents the researchers' methods for hand gesture recognition using marker videos.

Table 2. Related work for hand gesture recognition using marker sensors.

Authors	Methodology
Safa et al. [29]	Currently, the hand gesture system deploys many recognition systems with sensors to locate the correct motion and gesture of the hand without any distortion. Combining machine learning and sensors increases the potential in the field of digital entertainment by using touchless and touch-dynamic hand motion. In a recent study, a leap motion device was used to detect the dynamic motion of the hand without touching it, analyse the sequential time series data using long short-term memory (LSTM) for recognition, and separate unidirectional and bidirectional LSTM. The novel model, named Hybrid Bidirectional Unidirectional LSTM (HBU-LSTM), improves performance by considering spatial and temporal features between leap motion data and neural network layers.
Xiaoliang et al. [30]	The hand gesture system, with a novel approach, combines a wearable armband and customized pressure sensor smart gloves for sequential hand motion. The data collected from the inertial measurement unit (IMU), fingers, palm pressure, and electromyography was computed using deep learning. Long and short-term memory models (LSTM) for testing and training were applied. The experimental work showed outstanding results with dynamic and air gestures collected from ten different participants.

Table 2. Cont.

Authors	Methodology
Muhammad et al. [31]	In a smart home, the automatic system developed for the elder’s care deployed a home automation system with the gesture to control the appliances of daily use by using embedded hand gloves to detect the motion of the hand. For hand movements, wearable sensors such as an accelerometer and gyroscope were used to collect the combined feature set, and a random forest classifier was used to recognize the nine different gestures.
Dong-Luong-Dinh et al. [32]	In hand gesture recognition for home appliances, a novel approach towards detection is provided in this paper. They controlled home appliances using hand gestures by detecting hands and generating control commands. They created a database for hand gestures via labelling part maps and then classifying them using random forests. They generated a system for TV, lights, doors, changing channels, fans, temperature, and volume using hand gestures.
Muhammad Muneeb et al. [33]	Smart homes for the elderly and disabled people need special attention, as awareness of geriatric problems is necessary to resolve these issues. Researchers have developed many gesture recognition systems in various domains, but the authors of this paper presented a way to deal with elderly issues in particular. They used gloves to record the movements of the rotation, tilting of the hand, and acceleration. The nine gestures were classified using random forest, attaining an accuracy of 94% over the benchmark dataset.
Chi-Huang Hung et al. [34]	They proposed a system for an array lamp that performed ON/OFF actions and dimmed the light. They used a gyroscope and an accelerometer for hand detection. The noise was removed using a Kalman filter, and signals were decoded after receiving them from the devices to convert them into the desired gestures.
Marvin S. Verdadero et al. [35]	Remote control devices are common, but the setup is very expensive. The static hand gestures are taken from an Android mobile, and the signals are passed to the electronic devices. The distance should be 6 m from the device to pass the signals accurately for gesture recognition.
Zhiwen Deng [36]	Sign language recognition (SLR) is an efficient way to bridge communication gaps. SLR can additionally be used for human–computer interaction (HCI), virtual reality (VR), and augmented reality (AR). To enhance the research study, they proposed a skeleton-based self-distillation multi-feature learning method (SML). They constructed a multi-feature aggregation module (MFA) for the fusion of the features. For feature extraction and recognition, a self-distillation-guided adaptive residual graph convolutional network (SGA-ResGCN) was used. They tested the system on two benchmark datasets, WLASL and AUTSL, attaining accuracies of 55.85% and 96.85%, respectively.
Elahe Rahimian [37]	For the reduction in computation costs in complex architectures while training larger datasets, they proposed a temporal convolution-based hand gesture recognition system (TC-HGR). The 17 gestures were trained using attention mechanisms and temporal convolutions. They attained 81.65% and 80.72% classification accuracy for window sizes of 300 ms and 200 ms, respectively.

3. Materials and Methods

3.1. System Methodology

The proposed architecture detects hand gestures in a dynamic environment. Primarily, for a dynamic image, the images are first converted into frames. The acquired images are passed through an adaptive mean filter for noise reduction, and then gamma correction is applied to the images to adjust the image intensity for better detection. On the filtered images, skin color is detected, and a saliency map is applied over it for hand extraction. The extracted hand is trained over a pre-defined model for the hand skeleton. After that, the detected hand and skeleton are used for feature extraction. The features include a joint color cloud, neural gas, and a directionally active model. The features are optimized to reduce complexity via graph mining. Finally, for the gestures, an RNN is implemented for classification. The architecture of the proposed system is presented in Figure 1.

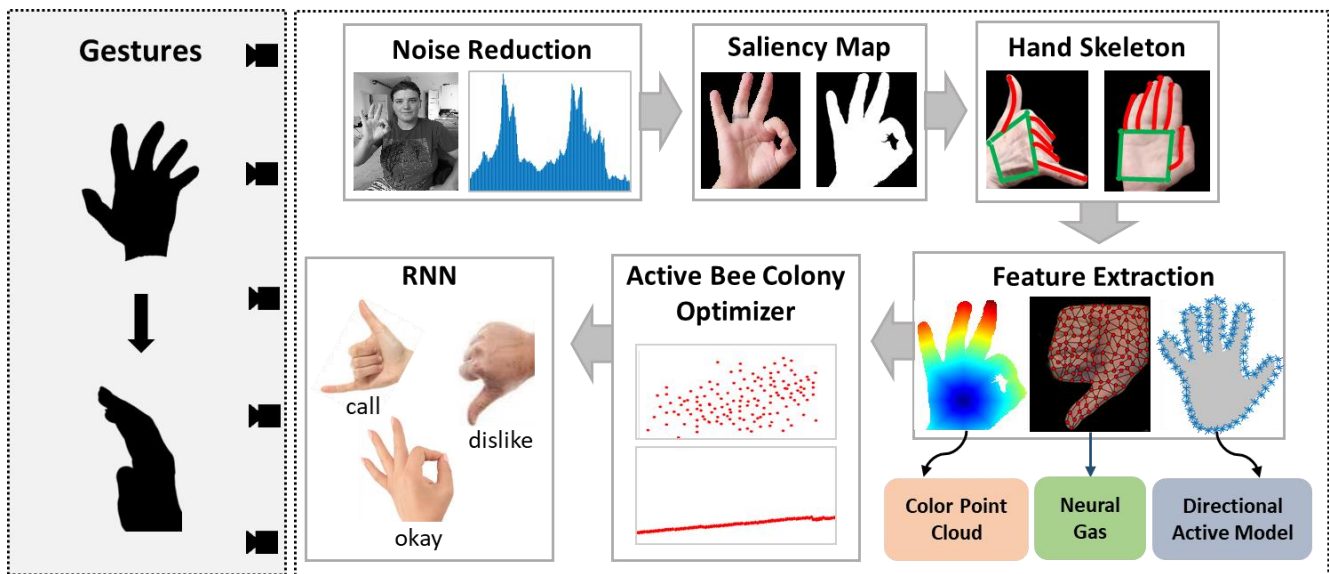


Figure 1. Architecture of the proposed system for hand gesture recognition.

3.2. Images Pre-Processing

In the acquired image, noise reduction is necessary to remove extra pixel information, as extra pixels hinder detection [38–41]. An adaptive median filter is used to detect the pixels affected by noise. This filter maintains the image quality, and the image blurring effect is negated. The pixels in the noised image are compared with the values of their neighboring pixels. A pixel showing a dissimilar value is labelled as a noisy pixel and a filter is applied over it. The pixel value is adjusted and replaced with the value of its neighboring pixels. For every pixel, the local region statistical estimate is calculated, resulting in \hat{a} ; a is the uncorrupted image, and \hat{a} is obtained from this image. The mean square error (MSE) is minimized between these two images, \hat{a} and a . The MSE is presented as follows:

$$m^2 = Ea - \hat{a}^2 \quad (1)$$

Conventional filters change all pixel values to denoise the image, but adaptive median filters work in two ways to change only the dissimilar pixels. Between level A and level B, level A is presented as follows:

$$\begin{aligned} A1 &= Q_{med} - Q_{min} \\ A2 &= Q_{med} - Q_{max} \end{aligned} \quad (2)$$

where Q_{med} represents the median of the gray level in the original image I_{xy} ; Q_{min} is the minimum gray level in I_{xy} ; Q_{max} is the maximum gray level in I_{xy} . If $A1 > 0$ and $A2 < 0$, there is a shift to level B. Otherwise, the window size is increased if the window size is less than or equal to I_{max} repeat level A, whereas I_{max} represents the maximum size of I_{xy} . Otherwise, the gray level coordinates Q_{xy} are shown. Level B is presented as follows:

$$\begin{aligned} B1 &= Q_{xy} - Q_{min} \\ B2 &= Q_{xy} - Q_{max} \end{aligned} \quad (3)$$

If $B1 > 0$ and $B2 < 0$ then Q_{xy} is shown, otherwise Q_{med} is shown. Figure 2 shows a flowchart of the algorithm implemented for the filter.

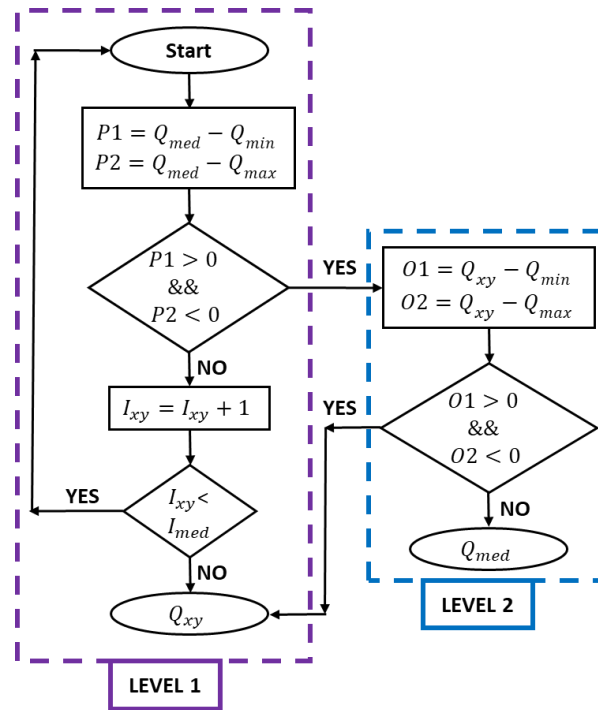


Figure 2. Sequential model representation for adaptive median filter algorithm.

The denoised image intensity is adjusted via gamma correction, as brightness plays a key role in the detection of a region of interest [42]. The power law for gamma correction is defined as follows:

$$W_o = GW_I^\gamma \quad (4)$$

where W_I is the input non-negative value with power γ and G is the constant usually equal to 1, and the range can lie between 0 and 1. W_o is the output value [43–45]. The denoised intensity-adjusted image, including the plot, is shown in Figure 3.

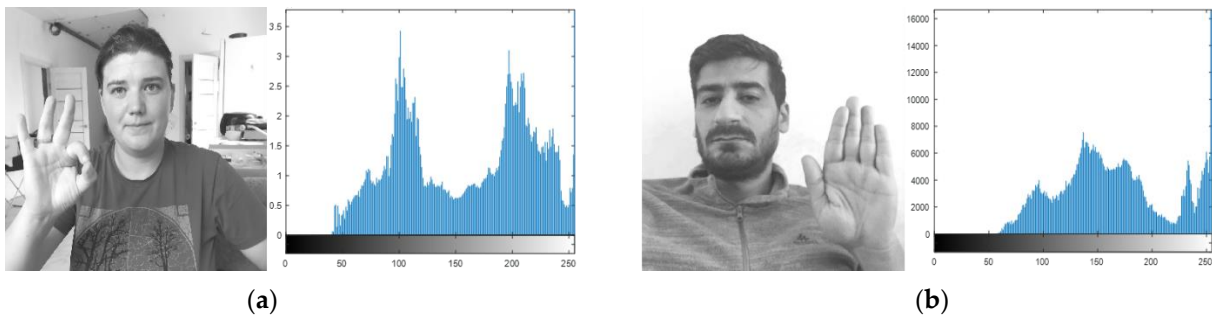


Figure 3. Pre-processed images with histograms of HaGRI dataset gestures: (a) ok; (b) stop.

3.3. Hand Detection

In this section, the hand is detected from the images using a two-way model. First, the skin tone from pixels is detected using hand gestures to localize the region of interest [46–50]. Then, a saliency map is applied over the image to obtain a better view of the desired gesture. The saliency map goal is to find the appropriate localization map, which is computed as follows:

$$M_s^h \in \mathbb{R}^{u*v} = \text{ReLU} \left(\sum_i \alpha_i^h H^i \right) \quad (5)$$

$$\alpha_i^h = \frac{1}{R} \left(\sum_i \sum_j \right)$$

where M_s^h is the localization map for the region of interest; $u * v$ represents the width and height of the image; i is the region of interest; α_i^h represents the global average pooling; R is the gradient via backpropagation. The average of the feature map is calculated using the weights assigned to the pixel gradient. Then, the *ReLU* is applied over the feature map. The image view range is set between 0 and 1, and the image is upscaled and overlay on the original image, resulting in a saliency map [51–53]. Figure 4 presents the saliency map for the HaGRI dataset “stop” and “ok” gestures.

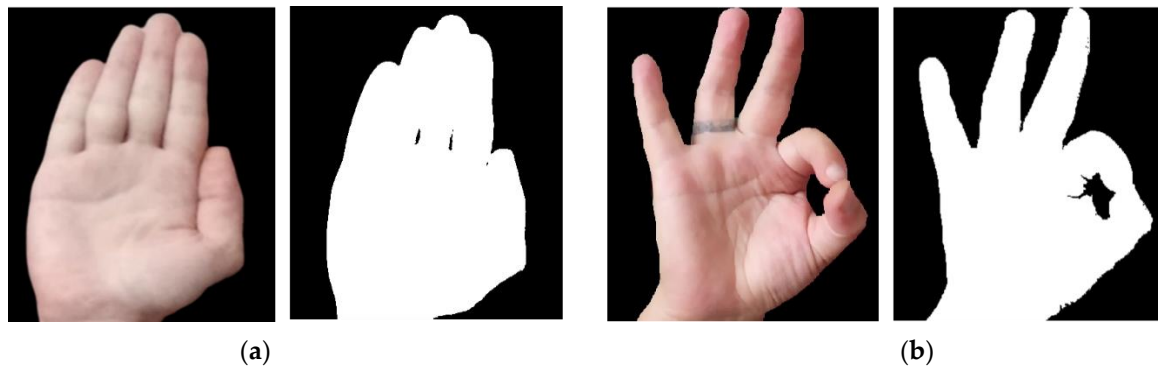


Figure 4. Hand detection using saliency map on the gestures (a) stop and (b) ok.

3.4. Hand Skeleton

For hand skeleton mapping, hand localization is the foremost step [54]. In our research, we first separated the palm and fingers for an accurate classification of the skeleton points. For palm extraction, a single-shot multibox detector (SSMD) is used; it excludes the fingers, and only the palm is bound by the blob. Then, the palm is first converted into binary, and a four-phase sliding window is moved across the whole area for the detection of the four extreme left, right, top, and bottom points. The second phase of the system includes finger identification; again, SSMD is used to detect the fingers. The palm is excluded, and the four-phase sliding window is moved to the extracted fingers again. It identified the extreme top, bottom, left, and right points [55]. From the extreme tops, the curves of the pixels are noted and marked. As a result, five points on the fingers and four points on the palm are obtained. Figure 5 shows the hand skeleton results for the HaGRI dataset.

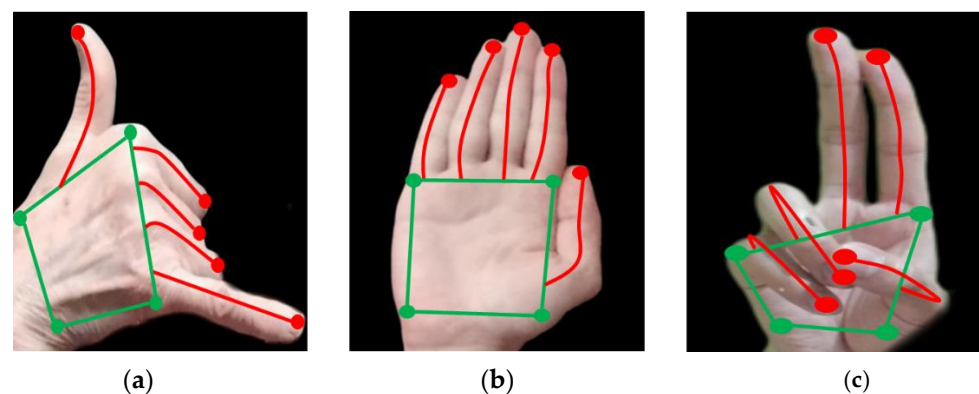


Figure 5. Hand skeleton mapping presenting palm and finger extreme points over gestures: (a) call; (b) stop; (c) two up.

3.5. Fusion Features Extraction

In this section, we illustrate how to extract various features from the acquired hand gestures. In hand gesture recognition systems, feature extraction contains two types of features: full-hand and point-based [56]. The full-hand feature set is made up of two techniques: a joint colour cloud and neural gas. A directionally active model is included in

the point-based feature. Both the extracted features are fused together to generate a feature set for recognition.

3.5.1. Joint Color Cloud

For this feature, the algorithm used to generate the cloud with different colors, which helps to obtain the skeleton point accuracy, and the geodesic distance for all fingers, including the palm, is extracted for the feature set. The color cloud is generated using a fast-marching algorithm [56–58]. This algorithm is defined as follows:

- (1) Suppose we are interested in the region of interest function value $f(i, j)$. This leads to two types of spatial derivative operators.

$$\begin{aligned} S^{+i}f &= \frac{f(i+t, j) - f(i, j)}{t} \\ S^{-i}f &= \frac{f(i, j) - f(i+t, j)}{t} \end{aligned} \quad (6)$$

where $S^{+i}f$ is the forward operator, as it uses the $f(i+t, j)$ to propagate from right to left by finding the value of $f(i, j)$. On the other hand, $S^{-i}f$ represents the backward operator, propagating from left to right.

- (2) For the difference operator, a discrete function is used to calculate $f_{i,j}$. For this purpose, at a specific point, the speed function $P_{i,j}$ is defined as follows:

$$\left[\begin{aligned} &\max(S_{i,j}^{-i}f, -S_{i,j}^{+i}f, 0)^2 \\ &+\max(S_{i,j}^{-j}f, -S_{i,j}^{+j}f, 0)^2 \end{aligned} \right]^{\frac{1}{2}} = \frac{1}{P_{i,j}} \quad (7)$$

The above equation is interpreted as follows, where (i, j) is the arrival time of $f_{i,j}$.

$$\left[\begin{aligned} &\text{Max}(f_{i,j} - f_{i-1,j}, f_{i,j} - f_{i+1,j}, 0)^2 \\ &+\max(f_{i,j} - f_{i,j-1}, f_{i,j} - f_{i,j+1}, 0)^2 \end{aligned} \right]^{\frac{1}{2}} = \frac{1}{P_{i,j}} \quad (8)$$

- (3) For the neighbor pixel value calculation, only $f_{i,j}$ point included in the set point (i, j) can be used. The $f_{i,j}$ value computation is defined as follows:

$$\begin{aligned} p &= \min(f_{i-1,j}, f_{i+1,j}) \\ q &= \min(f_{i,j-1}, f_{i,j+1}) \end{aligned} \quad (9)$$

- (4) The quadratic equation is formulated for $f_{i,j}$: if $\frac{1}{f_{i,j}} > |p - q|$, which leads to the following:

$$f_{i,j} = \frac{p + q + \sqrt{2\left(\frac{1}{f_{i,j}}\right)^2 - (a - b)^2}}{2} \quad (10)$$

$$\text{otherwise, } f_{i,j} = \left(\frac{1}{f_{i,j}}\right)^2 + \min(a, b)$$

These computations have only been performed on the neighbors of the new points added. If the neighboring value and calculated point (i, j) are equal then the values are compared, and the smaller value calculated before is added. In every iteration, a smaller value is found and stored. To save time, the min heap is used in the fast-marching algorithm to store the minimum values quickly with less time consumption. These iterations continue until the endpoint is achieved [59,60]. Figure 6 shows the results for the point-colored cloud.

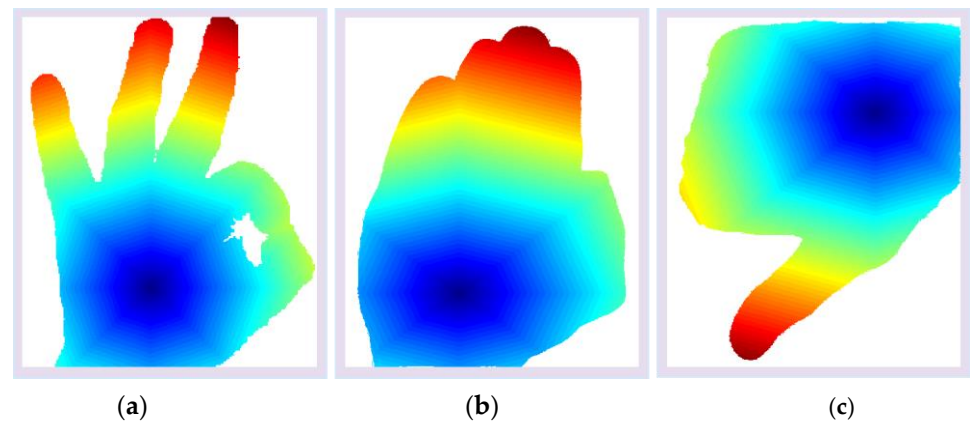


Figure 6. Wave propagation of point-colored cloud over HaGRI datasets gesture. (a) ok; (b) stop; (c) dislike.

3.5.2. Neural Gas

Neural maps organize themselves and form neural gas; it shows the ability to rank neighborhood vectors, which determine the neighborhood data space [61,62]. The neural gas is composed of multiple neurons, n , comprising weight vectors $W(r)$ that result in forming clusters. During training, every single neuron presents a change in position with an abrupt movement. Randomly, a feature vector is assigned to every single neuron. From the formed neural gas network, random data r is selected from the feature vector. With the help of this data vector r , the Euclidean distance is calculated from all the weight vectors. The distance values computed determine the center adjustment with the selected data vector [63–65]. The feature vector itself is defined as follows:

$$W_{f_n}^{m+1} = W_{f_n}^m + \varepsilon \cdot e^{-n/\lambda} \cdot (r - W_{f_n}^m), \quad (11)$$

$$n = 0, \dots, N - 1$$

where the probability distribution $W(r)$ of the data vector n with a finite number of sets s_f , $f = 1, \dots, N$. A data vector n for probability distribution $W(r)$ is presented at each time step m . The distance order is determined from the feature vector of the given data r . If n_0 is the index of the closed feature vector, n_1 is the second, and n_{N-1} is distant to the data vector n , then ε represents adaptation step size and λ represents neighborhood range. After most of the adaptation steps, the data space is covered with a feature vector with minimum errors. Algorithm 1 defines the pseudocode for neural gas formation, and Figure 7 presents the structure of the neural gas over the HaGRI dataset gesture.

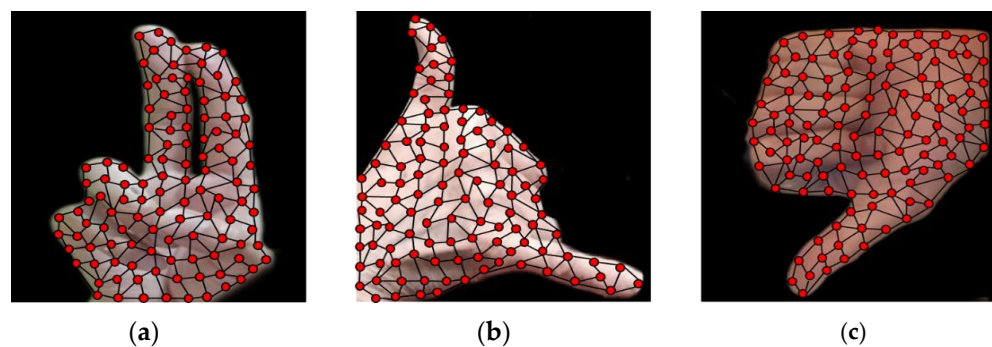


Figure 7. Neural gas formation with self-organized shape for gestures:(a) two up; (b) call; (c) dislike.

Algorithm 1: Pseudocode for Neural Gas Formation

Input: I : Input space;
Output: $G = (n_0, n_1, \dots, n_N)$: the map;
 $I \leftarrow []$
Method:
 $I \leftarrow N(n_0, n_1)$, where n_0 represents the first node and n_1 represents the second node
 $n_0 \leftarrow 0$;
 $n_N \leftarrow 100$;
Whereas, the input signal Φ is as follows:
 $I \leftarrow [\Phi]$
 Calculate winning nodes nearest Φ
 $p_1 \leftarrow \operatorname{argmin}_{o \in O} \|\Phi - w_n\|$
 $p_2 \leftarrow \operatorname{argmin}_{o \in \{p_1\}} \|\Phi - w_n\|$
 Adjust p_1 and p_2
 $edge \leftarrow edge \cup (p_1, p_2)$
 $edge \leftarrow 0$
 $\text{error}_{p_1} \leftarrow \text{error}_{p_1} + \|\Phi - w_{p_1}\|$
 Adjust $edge$
 $I \leftarrow [n_{i+1}]$
repeat
until $n_N \leftarrow 100$
end while
return $G = (n_0, n_1, \dots, n_N)$

3.5.3. Directional Active Model

The next feature is extracted using an 8-Freeman chain code algorithm, which measures the change in the directions of the curves at the boundary of the hand gesture [66]. Eight Freeman chain codes are shape descriptors, and they change structural schemes with a contour-dependent scheme. A shape description possesses a set of lines oriented in a particular manner. The oriented vectors are in eight and four directions, and the chain code vectors have integer numbers represented in a possible direction, as shown in Figure 8.

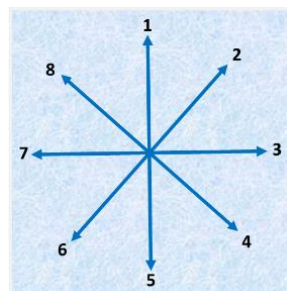


Figure 8. Direction representation of eight Freeman chain codes.

First, the boundary of the hand is identified to obtain the curves. Suppose the points on the curve are denoted by c on the boundary d . The starting point t on the top-right side of the thumb orientation is checked for its vector position. The curve points on the boundary P_b are calculated for all points, so it becomes $P_b = \{t_0, t_1, \dots, t_{n-1}\}$. After attaining the vector position of t_0 and t_1 , both of the curve point directions are compared; if they both have the same values, the value of t_1 is not considered and the next point t_2 vector position is checked; otherwise, both of the curve point values are added to the list. Hence, this whole procedure continues until t_{n-1} is reached. Figure 9 depicts the flow of the point extraction in a directionally active model.

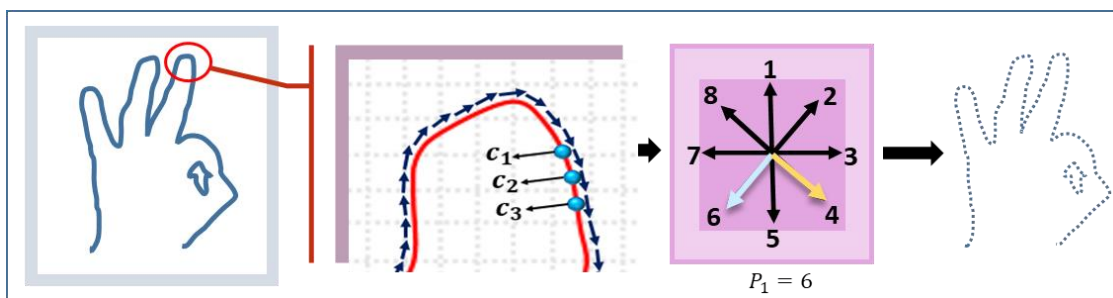


Figure 9. Flow sheet of point extraction in a directionally active model.

For our proposed system feature vector, we considered only 12 positions: 8 with an angle of 45° and 5 with an angle of 90°. The angle description is shown in Figure 10, which illustrates a better demonstration of the feature vector [67].

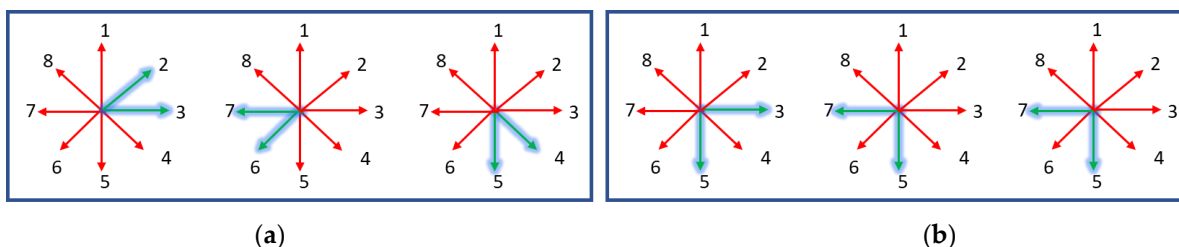


Figure 10. Angle demonstration for active directional model feature sets: (a) 45°; (b) 90°.

3.6. Feature Analysis and Optimization

After feature extraction from all datasets, the extracted features are passed through an artificial bee colony algorithm (ABCA) for optimization [68]. This helps reduce the computation time and also the complexity of the data. ABCA consists of two groups: one is known as the employer bee and the other is the onlooker bee. Both groups of bees have the same number, which is similar to the solutions in the group of honey bees, known as a swarm. The swarm size generates a randomly distributed initial population. Suppose the number of j -th solutions in the swarm is denoted as $X_j = (x_{j,1}, x_{j,2}, \dots, x_{j,n})$. Employed bees find their food sources as follows:

$$a_{j,i} = x_{j,i} + \varnothing_{j,i} \cdot (x_{j,i} - x_{j,l}) \tag{12}$$

where X_l represents the candidate solution and is randomly selected when $j \neq l$. $\varnothing_{j,i}$ represents a random number from the range $[-1, 1]$. l is the dimension index from $\{1,2,3,\dots,N\}$. When the food search by employee bees is completed, they share all the information between the onlookers and nectar. Then, they choose the food amount equal to the nectar amount. The fitness function of the new candidate solution is defined as follows:

$$Pn_j = \frac{fit(j)}{\sum_{j=1}^N fit(j)} \tag{13}$$

where Pn_j is the probability of the food source, which is higher if the solution better than j is achieved. fit represents the fitness value in the j -th swarm size. With predefined function iterations, if the position is not changed, then the value of the food source X_j is replaced with $X_{j,i}$ found by scout bees:

$$X_{j,i} = kb_i + rand(0,1) \cdot (ob_i - kb_i) \tag{14}$$

where ob_i and kb_i are the lower and upper boundaries of the i -th dimension; $rand(0,1)$ represents the random values between 0 and 1, respectively. Figure 11 presents the overall

flowchart of the artificial bee colony to determine the decision steps, while Figure 12 presents the best fitness result over the “call” gesture in the HaGRI dataset.

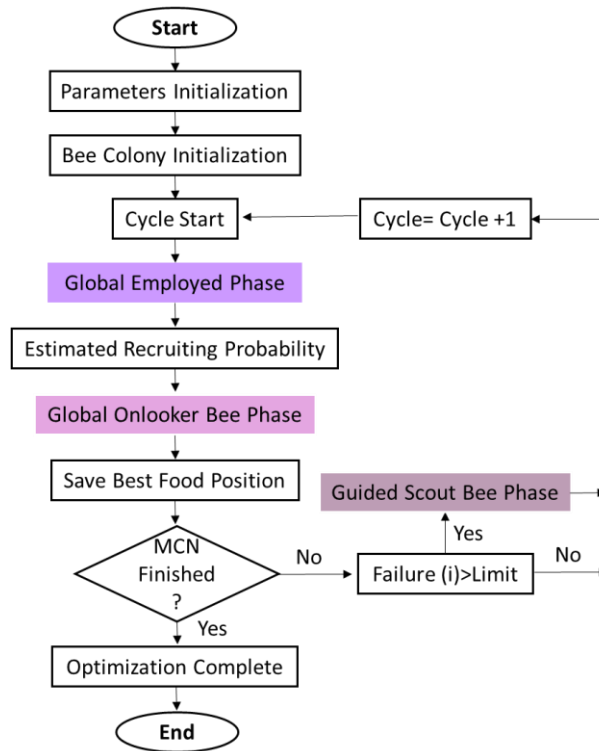


Figure 11. Flowchart of a working model for ABCA.

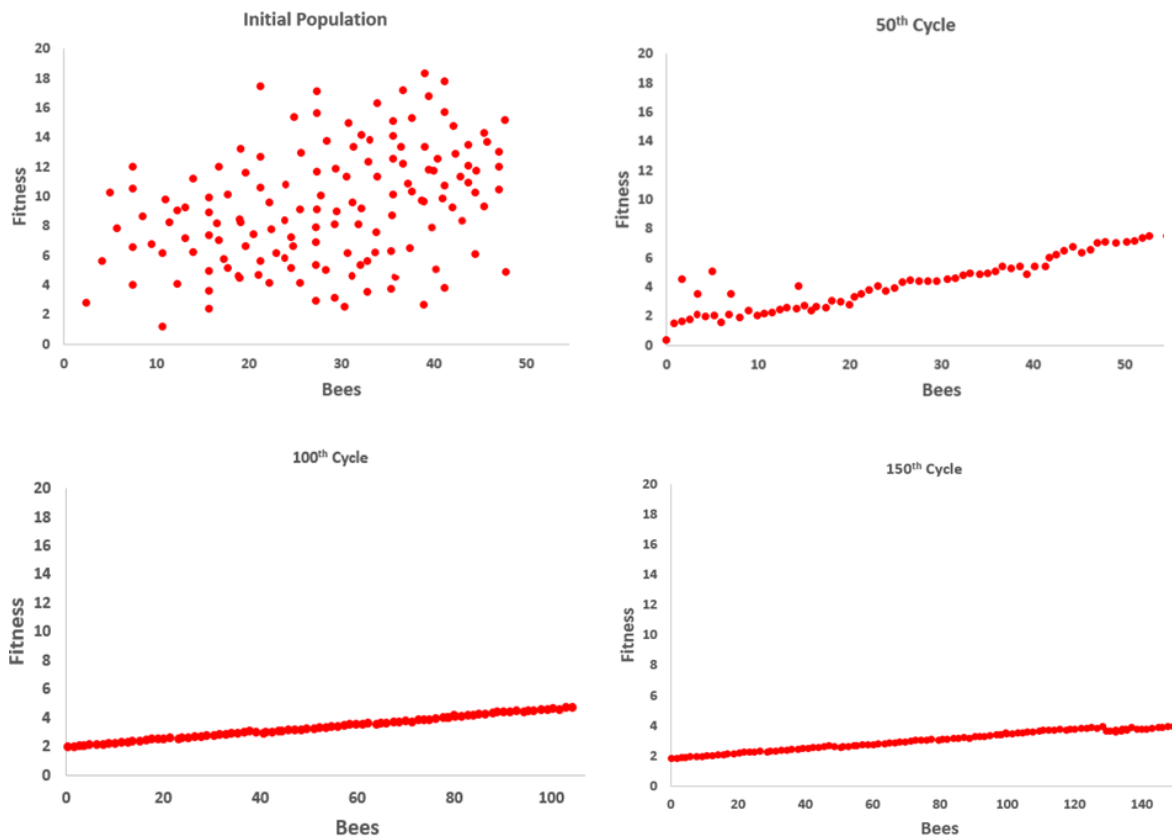


Figure 12. Angle fitness along the number of iterations over the “call” gesture.

3.7. Gesture Classification Using RNN

We used a recursive neural network (RNN) on our optimized feature vectors to classify gestures [69]. An RNN is a deep neural network that has the ability to learn distributive and structured data. Therefore, it is ideal for our proposed system of classification. In an RNN, the last output is typically used as the input for the next layer with hidden states. For each timestamp ts , the activation function $d^{(ts)}$ and the output $o^{(ts)}$ defined are as follows:

$$d^{(ts)} = k_1 \left(U_{dd}d^{(ts-1)} + U_{db}b^{(ts)} + g_d \right) \quad (15)$$

$$o^{(ts)} = k_2 \left(U_{od}d^{(ts)} + g_y \right) \quad (16)$$

where U_{dd} , U_{db} , U_{od} , g_d , g_y are the coefficients shared temporarily. k_1 , k_2 are activation functions. Figure 13 presents the overall flow of the RNN architecture.

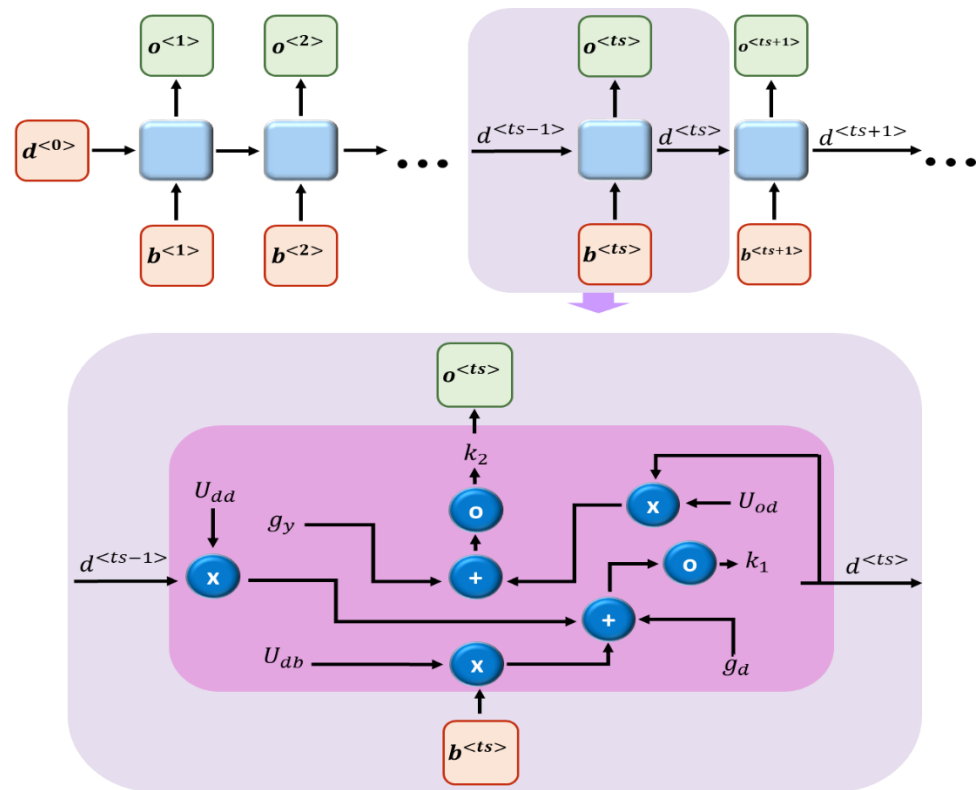


Figure 13. Overall flow of RNN architecture.

4. Experimental Setup and Evaluation

Experiments were performed on a system with the specifications of an Intel Core i7-9750H with 2.60GHz processing power, and 16GB RAM with $\times 64$ based Windows 10. The MATLAB tool and Google Colab were used for attaining the results. The system accessed the performance of the proposed architecture on four benchmark datasets: HaGRI, Geogesture, Jester, and WLASL. The k-fold cross-validation technique was applied to all three datasets to verify the reliability of our proposed system. This section includes a dataset description, the experiments performed, and a system comparison with other state-of-the-art systems.

4.1. Dataset Descriptions

4.1.1. HaGRI Dataset

The HaGRI Dataset [70] is specially designed for home automatic, automatic sector, and video conferencing. It consists of 552,992 RGB frames with 18 different gestures. The

dataset includes 34,730 subjects who performed gestures with different backgrounds. The subjects were aged between 18 and 65 years old. The gestures were performed indoors with different light intensities. The gestures used in our experiments were call, dislike, like, mute, ok, stop, and two up. Figure 14 presents the gestures from the HaGRI dataset.

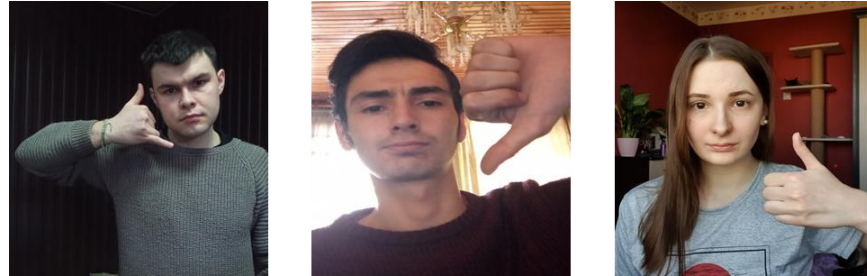


Figure 14. Example gesture frames from the HaGRI dataset.

4.1.2. Egogesture Dataset

The Egogesture [71] contains 2081 RGB videos and 2,953,224 frames with 83 different static and dynamic gestures. The gestures contain indoor and outdoor scenes. For our system training and testing, we selected seven dynamic gesture classes: scroll hand towards the right, scroll hand downward, scroll hand backward, zoom in with fists, zoom out with fists, rotate finger clockwise, and zoom in with fingers. The dataset samples with different gestures and different backgrounds are presented in Figure 15.

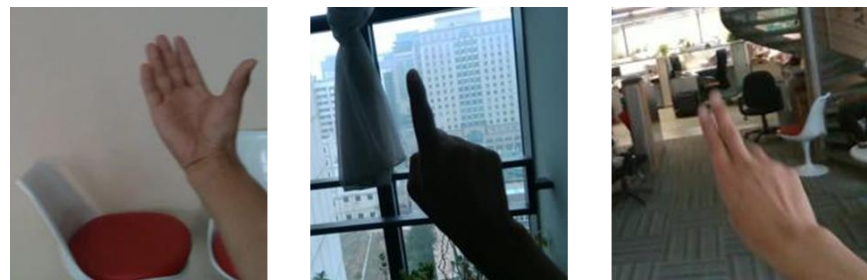


Figure 15. Example gesture frames from the Egogesture dataset.

4.1.3. Jester Dataset

The Jester dataset [72] contains 148,092 video clips of pre-defined human hand gestures collected in front of cameras; it comprises 27 gestures. The video quality of the gestures is set to 100 pixels at 12 fps. Seven hand gestures are selected for system training and testing for the following: sliding two fingers down, stop sign, swiping left, swiping right, turning the hand clockwise, turning the hand counterclockwise, and zooming in with two fingers. The example gestures of the Jester dataset are shown in Figure 16.



Figure 16. Example gesture frames from the Jester dataset.

4.1.4. WLASL Dataset

The WLASL dataset has the largest number of videos of American Sign Language hand gestures [73]. It has a total of 2000 hand gesture classes. The dataset was created specifically for communication between the deaf and hearing communities. We used seven classes to test the validity of our proposed model on hand gesture recognition datasets: hungry, wish, scream, forgive, attention, appreciate, and abuse. The WLASL dataset sample images are shown in Figure 17.



Figure 17. Example gesture frames from the WLASL dataset.

4.2. Evaluation via Experimental Results

We evaluated the performance of our proposed system on all three datasets, and the experiments proved the system's efficiency. Tables 3–6 illustrate the confusion matrices for the HaGRI, Egogesture, Jester, and WLASL datasets, achieving accuracy of 92.57%, 91.86%, 91.57%, and 90.43%, respectively. The experiments were repeated many times to evaluate the efficiency of the results. The HaGRI dataset presented the highest accuracy over the other datasets because of the higher resolution, and the hand extraction showed better results than the other datasets. Tables 7–10 depict the gesture evaluation matrices for the HaGRI, Egogesture, Jester, and WLASL datasets. This presents the gesture class accuracy, precision, recall, and f1 score for all the benchmark datasets used. This section also compares the selected classifier's accuracies to those of other conventional methods to demonstrate why they are preferred over other algorithms. Figure 18 demonstrates the comparison of the accuracy of RNN with other-state-of-the-art algorithms. Table 11 presents a comparison of our system with other conventional systems in the literature.

Table 3. Confusion matrix for gesture classification by the proposed approach using the HaGRI dataset.

Gesture Classes	Call	Dislike	Like	Mute	Ok	Stop	Two Up
call	0.93	0	0	0.03	0	0	0.04
dislike	0	0.92	0	0	0.05	0	0.03
like	0.05	0	0.95	0	0	0	0
mute	0	0.04	0	0.94	0	0.02	0
ok	0	0	0.07	0	0.93	0	0
stop	0	0.05	0	0.05	0	0.90	0
two up	0	0	0.04	0	0	0.05	0.91
Mean Accuracy = 92.57%							

Table 4. Confusion matrix for gesture classification by the proposed approach using the Egogesture dataset.

Gesture Classes	Scroll Hand towards Right	Scroll Hand Downward	Scroll Hand Backward	Zoom in with Fists	Zoom Out with Fists	Rotate Finger Clockwise	Zoom in with Fingers
scroll hand towards the right	0.90	0	0	0.03	0	0.07	0
scroll hand downward	0	0.93	0.07	0	0	0	0
scroll hand backward	0	0	0.92	0	0.05	0	0.03
zoom in with fists	0	0.03	0	0.93	0	0.04	0
zoom out with fists	0.04	0	0	0	0.94	0	0.02
rotate finger clockwise	0	0.07	0	0	0.02	0.91	0
zoom in with fingers	0.04	0	0	0.06	0	0	0.90
Mean Accuracy = 91.86%							

Table 5. Confusion matrix for gesture classification by the proposed approach using the Jester dataset.

Gesture Classes	Sliding Two Fingers Down	Stop Sign	Swiping Left	Swiping Right	Turning Hand Clockwise	Turning Hand Counterclockwise	Zoom in with Two Fingers
Sliding two fingers down	0.91	0	0	0	0	0.09	0
stop sign	0	0.92	0	0.05	0	0	0.03
swiping left	0.01	0	0.93	0	0.06	0	0
swiping right	0.06	0	0	0.92	0	0.02	0
turning hand clockwise	0	0.04	0	0	0.92	0	0.04
turning hand counterclockwise	0	0	0.08	0	0	0.92	0
zoom in with two fingers	0.06	0	0	0	0.05	0	0.89
Mean Accuracy = 91.57%							

Table 6. Confusion matrix for gesture classification by the proposed approach using the WLASL dataset.

Gesture Classes	Hungry	Wish	Scream	Forgive	Attention	Appreciate	Abuse
hungry	0.91	0	0	0.08	0	0	0.01
wish	0	0.90	0.09	0	0.01	0	0
scream	0.02	0.06	0.92	0	0	0	0
forgive	0	0	0	0.90	0.07	0.03	0
attention	0	0	0.04	0	0.89	0	0.07
appreciate	0	0.09	0	0.01	0	0.90	0
abuse	0.01	0	0	0	0	0.08	0.91
Mean Accuracy = 90.43%							

Table 7. Performance evaluation of the proposed approach using the HaGRI dataset.

Gesture Classes	Accuracy	Precision	Recall	F1 Score
call	0.98	0.93	0.95	0.94
dislike	0.97	0.92	0.91	0.92
like	0.97	0.95	0.90	0.92
mute	0.98	0.94	0.92	0.93
ok	0.98	0.93	0.95	0.94
stop	0.97	0.90	0.93	0.91
two up	0.97	0.91	0.93	0.92

Table 8. Performance evaluation of the proposed approach using the Egogesture dataset.

Gesture Classes	Accuracy	Precision	Recall	F1 Score
scroll hand towards the right	0.97	0.90	0.92	0.91
scroll hand downward	0.97	0.93	0.90	0.92
scroll hand backward	0.98	0.92	0.93	0.92
zoom in with fists	0.98	0.93	0.91	0.94
zoom out with fists	0.98	0.94	0.93	0.94
rotate finger clockwise	0.97	0.91	0.89	0.90
zoom in with fingers	0.98	0.90	0.95	0.92

Table 9. Performance evaluation of the proposed approach using the Jester dataset.

Gesture Classes	Accuracy	Precision	Recall	F1 Score
Sliding two fingers down	0.96	0.91	0.88	0.89
stop sign	0.98	0.92	0.96	0.94
swiping left	0.97	0.93	0.92	0.93
swiping right	0.98	0.92	0.95	0.93
turning hand clockwise	0.97	0.92	0.89	0.91
turning hand counterclockwise	0.97	0.92	0.89	0.91
zoom in with two fingers	0.97	0.89	0.93	0.91

Table 10. Performance evaluation of the proposed approach using the WLASL dataset.

Gesture Classes	Accuracy	Precision	Recall	F1 Score
hungry	0.98	0.91	0.97	0.94
wish	0.96	0.90	0.86	0.88
scream	0.97	0.92	0.88	0.90
forgive	0.97	0.90	0.91	0.90
attention	0.97	0.89	0.92	0.90
appreciate	0.97	0.90	0.89	0.90
abuse	0.97	0.91	0.92	0.91

Table 11. Comparison of the proposed method using conventional systems.

Methods	HaGRID	Egogesture	Jester
P. Molchanov et al. [74]	-	0.78	-
D. Tran et al. [75]	-	0.86	-
R. Cutura et al. [76]	0.89	-	-
P. Padhi [77]	0.90	-	-
J. Yang et al. [78]	-	-	0.67
S. Li et al. [79]	-	-	0.73
Proposed Method	0.92	0.91	0.91

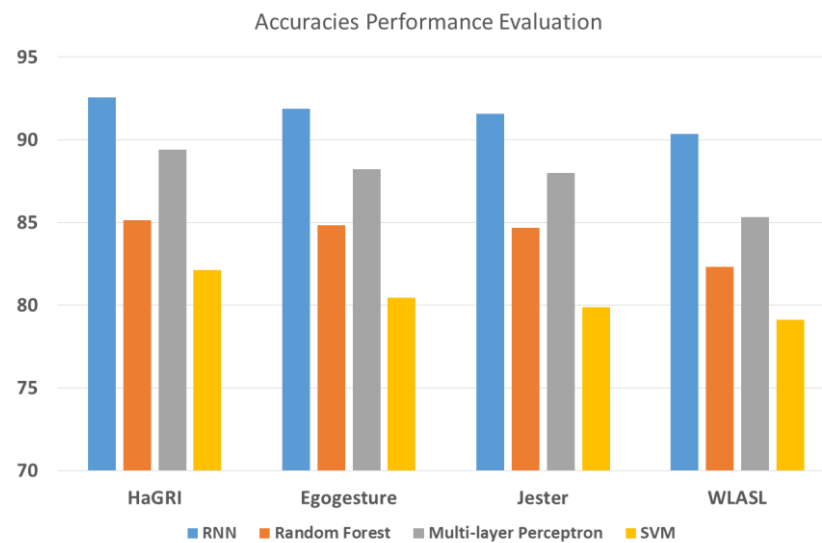


Figure 18. Accuracy comparison of RNN with other state-of-the-art algorithms.

5. Discussion

The proposed hand gesture recognition system model is designed to achieve state-of-the-art performance over RGB images. Initially, images with a variety of gestures and complex backgrounds are used as inputs from benchmark datasets, such as HaGRI, Egogesture, and Jester. Our suggested two-way method is used to process the images provided for hand extraction. There were also some shortcomings in the proposed approach that prevented concealed information from being accurately obtained from the hand skeletons. Frames with no suitable camera angle made it difficult to acquire the exact key points at hand. As presented in Figure 5a, the extreme key points are localized on the knuckles of the fingers due to the absence of the fingertips in the frame. The suggested system performed well on frames that initially presented the entire hand, followed by the movement of the hand. After the hand and skeleton extractions, the region of interest was passed through the fusion of features. The full-hand and one-point-based features were optimized and passed through RNN for recognition. The accuracy attained over the four datasets via RNN produced better results, with an accuracy of 92.57% using the HaGRI dataset; for Egogesture, it was 91.86%; for Jester, it was 91.57%; and for WLASL, it was 90.43%.

6. Conclusions

This paper provides a novel way of recognizing gestures in a home automation system. Home appliances like TVs, washing machines, lights, cleaning robots, printers, stoves, etc. can be controlled using hand gestures. Our system proposed a way to fulfill the requirement of detecting hands from a complex background via six steps, namely noise removal, hand detection, hand skeleton, feature extraction, optimization, and classification. The hand gestures were trained by preprocessing them first using the adaptive median algorithm. Then, the hand detection was performed using the two-way method, and after that, the hand skeleton was extracted using SSMD. From the extracted hand and skeleton points, fusion features were extracted, namely joint colour cloud, neural gas, and directional active model. The features were optimized using the active bee colony algorithm, which provided promising results for all four datasets. The accuracies attained using the HaGRI dataset was 92.57%; for Egogesture, it was 91.86%; Jester provided 91.57%; and WLASL showed 90.43%. The proposed system is for smart home automation, which was designed using different techniques. It provides a set of features for recognition, rather than conventional features, using only deep learning methods.

The proposed system needs to be trained with more gestures, and various experiments can be performed in different environments like healthcare, robotics, sports, and industries. The computation time needs to be considered to remove the complexity of the system.

The computational cost of the system can be managed by considering the architecture. In the future, we plan to work under different circumstances using computational cost management. Also, we will add more features and robust algorithms to make our system more efficient and standard for all environments.

Author Contributions: Conceptualization: H.A., N.A.M. and B.I.A.; methodology: H.A. and B.I.A.; software: H.A. and S.S.A.; validation: S.S.A., B.I.A. and A.A. (Abdulwahab Alazeb); formal analysis: A.A. (Abdullah Alshahrani) and N.A.M.; resources: A.J., S.S.A., B.I.A. and A.A. (Abdulwahab Alazeb); writing—review and editing: N.A.M. and B.I.A.; funding acquisition: N.A.M., A.A. (Abdullah Alshahrani), S.S.A., B.I.A. and A.A. (Abdulwahab Alazeb). All authors have read and agreed to the published version of the manuscript.

Funding: Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R440), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors are thankful to the Deanship of Scientific Research at Najran University for funding this study under the Research Group Funding program grant code (NU/RG/SERC/12/6).

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

References

1. Panwar, M.; Mehra, P.S. Hand gesture recognition for human computer interaction. In Proceedings of the IEEE 2011 International Conference on Image Information Processing, Shimla, India, 3–5 November 2011.
2. Khan, R.Z.; Ibraheem, N.A. Hand gesture recognition: A literature review. *Int. J. Artif. Intell. Appl.* **2012**, *3*, 161. [[CrossRef](#)]
3. Wu, C.H.; Lin, C.H. Depth-based hand gesture recognition for home appliance control. In Proceedings of the 2013 IEEE International Symposium on Consumer Electronics (ISCE), Hsinchu, Taiwan, 3–6 June 2013.
4. Solanki, U.V.; Desai, N.H. Hand gesture based remote control for home appliances: Handmote. In Proceedings of the 2011 IEEE World Congress on Information and Communication Technologies, Mumbai, India, 11–14 December 2011.
5. Hsieh, C.C.; Liou, D.H.; Lee, D. A real time hand gesture recognition system using motion history image. In Proceedings of the IEEE 2010 2nd International Conference on Signal Processing Systems, Dalian, China, 5–7 July 2010.
6. Chung, H.Y.; Chung, Y.L.; Tsai, W.F. An efficient hand gesture recognition system based on deep CNN. In Proceedings of the 2019 IEEE International Conference on Industrial Technology (ICIT), Melbourne, VIC, Australia, 13–15 February 2019.
7. Wang, M.; Yan, Z.; Wang, T.; Cai, P.; Gao, S.; Zeng, Y.; Wan, C.; Wang, H.; Pan, L.; Yu, J.; et al. Gesture recognition using a bioinspired learning architecture that integrates visual data with somatosensory data from stretchable sensors. *Nat. Electron.* **2020**, *3*, 563–570. [[CrossRef](#)]
8. Moin, A.; Zhou, A.; Rahimi, A.; Menon, A.; Benatti, S.; Alexandrov, G.; Tamakloe, S.; Ting, J.; Yamamoto, N.; Khan, Y.; et al. A wearable biosensing system with in-sensor adaptive machine learning for hand gesture recognition. *Nat. Electron.* **2021**, *4*, 54–63. [[CrossRef](#)]
9. Dang, L.M.; Min, K.; Wang, H.; Piran, M.J.; Lee, C.H.; Moon, H. Sensor-based and vision-based human activity recognition: A comprehensive survey. *Pattern Recognit.* **2020**, *108*, 107561. [[CrossRef](#)]
10. Mujahid, A.; Awan, M.J.; Yasin, A.; Mohammed, M.A.; Damaševičius, R.; Maskeliūnas, R.; Abdulkareem, K.H. Real-time hand gesture recognition based on deep learning YOLOv3 model. *Appl. Sci.* **2021**, *11*, 4164. [[CrossRef](#)]
11. Al-Hammadi, M.; Muhammad, G.; Abdul, W.; Alsulaiman, M.; Bencherif, M.A.; Alrayes, T.S.; Mathkour, H.; Mekhtiche, M.A. Deep learning-based approach for sign language gesture recognition with efficient hand gesture representation. *IEEE Access* **2020**, *8*, 192527–192542. [[CrossRef](#)]
12. Pinto, R.F.; Borges, C.D.; Almeida, A.M.; Paula, I.C. Static hand gesture recognition based on convolutional neural networks. *J. Electr. Comput. Eng.* **2019**, *2019*, 4167890. [[CrossRef](#)]
13. Tolentino, L.K.S.; Juan, R.O.S.; Thio-ac, A.C.; Pamahoy, M.A.B.; Forteza, J.R.R.; Garcia, X.J.O. Static sign language recognition using deep learning. *Int. J. Mach. Learn. Comput.* **2019**, *9*, 821–827. [[CrossRef](#)]
14. Beddiar, D.R.; Nini, B.; Sabokrou, M.; Hadid, A. Vision-based human activity recognition: A survey. *Multimed. Tools Appl.* **2020**, *79*, 30509–30555. [[CrossRef](#)]
15. Skaria, S.; Al-Hourani, A.; Lech, M.; Evans, R.J. Hand-gesture recognition using two-antenna Doppler radar with deep convolutional neural networks. *IEEE Sens. J.* **2019**, *19*, 3041–3048. [[CrossRef](#)]
16. Tabernik, D.; Skočaj, D. Deep learning for large-scale traffic-sign detection and recognition. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 1427–1440. [[CrossRef](#)]
17. Zheng, Y.; Lv, X.; Qian, L.; Liu, X. An Optimal BP Neural Network Track Prediction Method Based on a GA-ACO Hybrid Algorithm. *J. Mar. Sci. Eng.* **2022**, *10*, 1399. [[CrossRef](#)]
18. Qian, L.; Zheng, Y.; Li, L.; Ma, Y.; Zhou, C.; Zhang, D. A New Method of Inland Water Ship Trajectory Prediction Based on Long Short-Term Memory Network Optimized by Genetic Algorithm. *Appl. Sci.* **2022**, *12*, 4073. [[CrossRef](#)]

19. Dinh, D.L.; Kim, J.T.; Kim, T.S. Hand gesture recognition and interface via a depth imaging sensor for smart home appliances. *Energy Procedia* **2014**, *62*, 576–582. [[CrossRef](#)]
20. Kim, M.; Cho, J.; Lee, S.; Jung, Y. IMU sensor-based hand gesture recognition for human-machine interfaces. *Sensors* **2019**, *19*, 3827. [[CrossRef](#)] [[PubMed](#)]
21. Rautaray, S.S.; Agrawal, A. Vision based hand gesture recognition for human computer interaction: A survey. *Artif. Intell. Rev.* **2015**, *43*, 1–54. [[CrossRef](#)]
22. Pisharady, P.K.; Saerbeck, M. Recent methods and databases in vision-based hand gesture recognition: A review. *Comput. Vis. Image Underst.* **2015**, *141*, 152–165. [[CrossRef](#)]
23. Irie, K.; Wakamura, N.; Umeda, K. Construction of an intelligent room based on gesture recognition: Operation of electric appliances with hand gestures. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan, 28 September–2 October 2004.
24. Lone, M.R.; Khan, E. A good neighbor is a great blessing: Nearest neighbor filtering method to remove impulse noise. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 9942–9952. [[CrossRef](#)]
25. Ren, Z.; Meng, J.; Yuan, J. Depth camera based hand gesture recognition and its applications in human-computer-interaction. In Proceedings of the 2011 8th International Conference on Information, Communications & Signal Processing, Singapore, 13–16 December 2011.
26. Sahoo, J.P.; Prakash, A.J.; Plawiak, P.; Samantray, S. Real-time hand gesture recognition using fine-tuned convolutional neural network. *Sensors* **2022**, *22*, 706. [[CrossRef](#)] [[PubMed](#)]
27. Ding, J.; Zheng, N.W. RGB-D Depth-sensor-based Hand Gesture Recognition Using Deep Learning of Depth Images with Shadow Effect Removal for Smart Gesture Communication. *Sens. Mater.* **2022**, *34*, 203–216. [[CrossRef](#)]
28. Li, J.; Wei, L.; Wen, Y.; Liu, X.; Wang, H. An approach to continuous hand movement recognition using SEMG based on features fusion. *Vis. Comput.* **2023**, *39*, 2065–2079. [[CrossRef](#)]
29. Alam, M.M.; Islam, M.T.; Rahman, S.M. *A Unified Learning Approach for Hand Gesture Recognition and Fingertip Detection*; UMBC Student Collection; University of Maryland: Baltimore, MD, USA, 2021.
30. Ameer, S.; Khalifa, A.B.; Bouhlel, M.S. A novel hybrid bidirectional unidirectional LSTM network for dynamic hand gesture recognition with leap motion. *Entertain. Comput.* **2020**, *35*, 100373. [[CrossRef](#)]
31. Zhang, X.; Yang, Z.; Chen, T.; Chen, D.; Huang, M.C. Cooperative sensing and wearable computing for sequential hand gesture recognition. *IEEE Sens. J.* **2019**, *19*, 5775–5783. [[CrossRef](#)]
32. Hakim, N.L.; Shih, T.K.; Arachchi, S.P.K.; Aditya, W.; Chen, Y.-C.; Lin, C.-Y. Dynamic hand gesture recognition using 3DCNN and LSTM with FSM context-aware model. *Sensors* **2019**, *19*, 5429. [[CrossRef](#)]
33. Dong, B.; Shi, Q.; Yang, Y.; Wen, F.; Zhang, Z.; Lee, C. Technology evolution from self-powered sensors to AIoT enabled smart homes. *Nano Energy* **2021**, *79*, 105414. [[CrossRef](#)]
34. Muneeb, M.; Rustam, H.; Jalal, A. Automate appliances via gestures recognition for elderly living assistance. In Proceedings of the IEEE 2023 4th International Conference on Advancements in Computational Sciences (ICACS), Lahore, Pakistan, 20–22 February 2023.
35. Hung, C.H.; Bai, Y.W.; Wu, H.Y. Home appliance control by a hand gesture recognition belt in LED array lamp case. In Proceedings of the 2015 IEEE 4th Global Conference on Consumer Electronics (GCCE), Osaka, Japan, 27–30 October 2015.
36. Deng, Z.; Gao, Q.; Ju, Z.; Leng, Y. A Self-Distillation Multi-Feature Learning Method for Skeleton-Based Sign Language Recognition. *Pattern Recognit.* **2023**, *144*, 1–33.
37. Rahimian, E.; Zabihi, S.; Asif, A.; Farina, D.; Atashzar, S.F.; Mohammadi, A. Hand gesture recognition using temporal convolutions and attention mechanism. In Proceedings of the ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 23–27 May 2022.
38. Zhang, X.; Huang, D.; Li, H.; Zhang, Y.; Xia, Y.; Liu, J. Self-training maximum classifier discrepancy for EEG emotion recognition. *CAAI Trans. Intell. Technol.* **2023**. [[CrossRef](#)]
39. Khandizod, A.G.; Deshmukh, R.R. Comparative analysis of image enhancement technique for hyperspectral palmprint images. *Int. J. Comput. Appl.* **2015**, *121*, 30–35.
40. Soni, H.; Sankhe, D. Image restoration using adaptive median filtering. *IEEE Int. Res. J. Eng. Technol.* **2019**, *6*, 841–844.
41. Balasamy, K.; Shamia, D. Feature extraction-based medical image watermarking using fuzzy-based median filter. *IETE J. Res.* **2023**, *69*, 83–91. [[CrossRef](#)]
42. Veluchamy, M.; Subramani, B. Image contrast and color enhancement using adaptive gamma correction and histogram equalization. *Optik* **2019**, *183*, 329–337. [[CrossRef](#)]
43. Veluchamy, M.; Subramani, B. Fuzzy dissimilarity color histogram equalization for contrast enhancement and color correction. *Appl. Soft Comput.* **2020**, *89*, 106077. [[CrossRef](#)]
44. Liu, Z.; Lin, W.; Li, X.; Rao, Q.; Jiang, T.; Han, M.; Fan, H.; Sun, J.; Liu, S. ADNet: Attention-guided deformable convolutional network for high dynamic range imaging. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20 June–25 June 2021.
45. Rahman, T.; Khandakar, A.; Qiblawey, Y.; Tahir, A.; Kiranyaz, S.; Kashem, S.B.A.; Islam, M.T.; Al Maadeed, S.; Zughaiyer, S.M.; Khan, M.S.; et al. Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images. *Comput. Biol. Med.* **2021**, *132*, 104319. [[CrossRef](#)]

46. Ghose, D.; Desai, S.M.; Bhattacharya, S.; Chakraborty, D.; Fiterau, M.; Rahman, T. Pedestrian detection in thermal images using saliency maps. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–17 June 2019.
47. Ye, D.; Chen, C.; Liu, C.; Wang, H.; Jiang, S. Detection defense against adversarial attacks with saliency map. *Int. J. Intell. Syst.* **2021**, *37*, 10193–10210. [[CrossRef](#)]
48. Etmann, C.; Lunz, S.; Maass, P.; Schönlieb, C.B. On the connection between adversarial robustness and saliency map interpretability. *arXiv* **2019**, arXiv:1905.04172.
49. Zhao, Y.; Po, L.-M.; Cheung, K.-W.; Yu, W.-Y.; Rehman, Y.A.U. SCGAN: Saliency map-guided colorization with generative adversarial network. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *31*, 3062–3077. [[CrossRef](#)]
50. Li, H.; Li, C.; Ding, Y. Fall detection based on fused saliency maps. *Multimed. Tools Appl.* **2020**, *80*, 1883–1900. [[CrossRef](#)]
51. Rastgoo, R.; Kiani, K.; Escalera, S. Video-based isolated hand sign language recognition using a deep cascaded model. *Multimed. Tools Appl.* **2020**, *79*, 22965–22987. [[CrossRef](#)]
52. Yang, L.; Qi, Z.; Liu, Z.; Liu, H.; Ling, M.; Shi, L.; Liu, X. An embedded implementation of CNN-based hand detection and orientation estimation algorithm. *Mach. Vis. Appl.* **2019**, *30*, 1071–1082. [[CrossRef](#)]
53. Gao, Q.; Liu, J.; Ju, Z. Robust real-time hand detection and localization for space human–robot interaction based on deep learning. *Neurocomputing* **2020**, *390*, 198–206. [[CrossRef](#)]
54. Tang, J.; Yao, X.; Kang, X.; Nishide, S.; Ren, F. Position-free hand gesture recognition using single shot multibox detector based neural network. In Proceedings of the 2019 IEEE International Conference on Mechatronics and Automation (ICMA), Tianjin, China, 4–7 August 2019.
55. Tang, H.; Liu, H.; Xiao, W.; Sebe, N. Fast and robust dynamic hand gesture recognition via key frames extraction and feature fusion. *Neurocomputing* **2018**, *331*, 424–433. [[CrossRef](#)]
56. Tan, G.; Zou, J.; Zhuang, J.; Wan, L.; Sun, H.; Sun, Z. Fast marching square method based intelligent navigation of the unmanned surface vehicle swarm in restricted waters. *Appl. Ocean Res.* **2020**, *95*, 102018. [[CrossRef](#)]
57. Xia, J.; Jiang, Z.; Zhang, H.; Zhu, R.; Tian, H. Dual fast marching tree algorithm for human-like motion planning of anthropomorphic arms with task constraints. *IEEE/ASME Trans. Mechatron.* **2020**, *26*, 2803–2813. [[CrossRef](#)]
58. Muñoz, J.; López, B.; Quevedo, F.; Barber, R.; Garrido, S.; Moreno, L. Geometrically constrained path planning for robotic grasping with Differential Evolution and Fast Marching Square. *Robotica* **2023**, *41*, 414–432. [[CrossRef](#)]
59. Liu, Y.; Nedo, A.; Seward, K.; Caplan, J.; Kambhamettu, C. Quantifying actin filaments in microscopic images using keypoint detection techniques and a fast marching algorithm. In Proceedings of the 2020 IEEE International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, 25–28 October 2020.
60. Gadekallu, T.R.; Alazab, M.; Kaluri, R.; Maddikunta, P.K.; Bhattacharya, S.; Lakshmana, K. Hand gesture classification using a novel CNN-crow search algorithm. *Complex Intell. Syst.* **2021**, *7*, 1855–1868. [[CrossRef](#)]
61. Qi, J.; Jiang, G.; Li, G.; Sun, Y.; Tao, B. Surface EMG hand gesture recognition system based on PCA and GRNN. *Neural Comput. Appl.* **2020**, *32*, 6343–6351. [[CrossRef](#)]
62. Todorov, H.; Cannoodt, R.; Saelens, W.; Saeys, Y. TinGa: Fast and flexible trajectory inference with Growing Neural Gas. *Bioinformatics* **2020**, *36*, i66–i74. [[CrossRef](#)]
63. Hahn, C.; Feld, S.; Zierl, M.; Linnhoff-Popien, C. Dynamic Path Planning with Stable Growing Neural Gas. *InICAART* **2019**, 138–145. [[CrossRef](#)]
64. Mirehi, N.; Tahmasbi, M.; Targhi, A.T. Hand gesture recognition using topological features. *Multimed. Tools Appl.* **2019**, *78*, 13361–13386. [[CrossRef](#)]
65. Ansar, H.; Jalal, A.; Gochoo, M.; Kim, K. Hand gesture recognition based on auto-landmark localization and reweighted genetic algorithm for healthcare muscle activities. *Sustainability* **2021**, *13*, 2961. [[CrossRef](#)]
66. Zaaraoui, H.; El Kaddouhi, S.; Abarkan, M. A novel approach to face recognition using freeman chain code and nearest neighbor classifier. In Proceedings of the 2019 International Conference on Intelligent Systems and Advanced Computing Sciences (ISACS), Taza, Morocco, 26–27 December 2019.
67. Jalal, A.; Khalid, N.; Kim, K. Automatic recognition of human interaction via hybrid descriptors and maximum entropy markov model using depth sensors. *Entropy* **2020**, *22*, 817. [[CrossRef](#)] [[PubMed](#)]
68. Zahra, A.K.A.; Abdalla, T.Y. Design of fuzzy super twisting sliding mode control scheme for unknown full vehicle active suspension systems using an artificial bee colony optimization algorithm. *Asian J. Control* **2020**, *23*, 1966–1981. [[CrossRef](#)]
69. Dhruv, P.; Naskar, S. Image classification using convolutional neural network (CNN) and recurrent neural network (RNN): A review. In *Machine Learning and Information Processing: Proceedings of ICMLIP*; Springer: Singapore, 2019.
70. Kapitanov, A.; Makhlyarchuk, A.; Kvanchiani, K. HaGRID-HAnd Gesture Recognition Image Dataset. *arXiv* **2022**, arXiv:2206.08219.
71. Chalasani, T.; Smolic, A. Simultaneous segmentation and recognition: Towards more accurate ego gesture recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Long Beach, CA, USA, 16–17 June 2019.
72. Materzynska, J.; Berger, G.; Bax, I.; Memisevic, R. The jester dataset: A large-scale video dataset of human gestures. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Long Beach, CA, USA, 16–17 June 2019.
73. Naz, N.; Sajid, H.; Ali, S.; Hasan, O.; Ehsan, M.K. Signgraph: An Efficient and Accurate Pose-Based Graph Convolution Approach Toward Sign Language Recognition. *IEEE Access* **2023**, *11*, 19135–19147. [[CrossRef](#)]

74. Molchanov, P.; Yang, X.; Gupta, S.; Kim, K.; Tyree, S.; Kautz, J. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016.
75. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015.
76. Cutura, R.; Morariu, C.; Cheng, Z.; Wang, Y.; Weiskopf, D.; Sedlmair, M. Hagrid—Gridify scatterplots with hilbert and gosper curves. In Proceedings of the 14th International Symposium on Visual Information Communication and Interaction, Potsdam, Germany, 6–7 September 2021.
77. Padhi, P.; Das, M. Hand Gesture Recognition using DenseNet201-Mediapipe Hybrid Modelling. In Proceedings of the 2022 International Conference on Automation, Computing and Renewable Systems (ICACRS), Pudukkottai, India, 13–15 December 2022.
78. Li, S.; Aich, A.; Zhu, S.; Asif, S.; Song, C.; Roy-Chowdhury, A. Krishnamurthy. Adversarial attacks on black box video classifiers: Leveraging the power of geometric transformations. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 2085–2096.
79. Zhao, B.; Hua, X.; Yu, K.; Xuan, W.; Chen, X.; Tao, W. Indoor Point Cloud Segmentation Using Iterative Gaussian Mapping and Improved Model Fitting. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 7890–7907. [[CrossRef](#)]

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