computers

MDPI

Article

Cognitive Classifier of Hand Gesture Images for Automated Sign Language Recognition: Soft Robot Assistance Based on Neutrosophic Markov Chain Paradigm

Muslem Al-Saidi [1,2,*], Áron Ballagi [2,*], Oday Ali Hassen [3] and Saad M. Saad [4,5]

[1] Doctoral School of Multidisciplinary Engineering Sciences, Széchenyi István University, Egyetem tér 1, 9026 Győr, Hungary
[2] Department of Automation, Széchenyi István University, Egyetem tér 1, 9026 Győr, Hungary
[3] Ministry of Education, Wasit Education Directorate, Kut 52001, Iraq; odayali@uowasit.edu.iq
[4] Department of Information Technology, Institute of Graduate Studies and Research, Alexandria University, Alexandria 21526, Egypt; saad.darwish.pt@pua.edu.eg
[5] Department of Artificial Intelligence, Faculty of Computer Sciences and Artificial Intelligence, Pharos University in Alexandria, Canal El Mahmoudia Street, Beside Green Plaza, Alexandria 21648, Egypt
* Correspondence: muslem@uowasit.edu.iq (M.A.-S.); ballagi@ga.sze.hu (Á.B.)

Abstract: In recent years, Sign Language Recognition (SLR) has become an additional topic of discussion in the human–computer interface (HCI) field. The most significant difficulty confronting SLR recognition is finding algorithms that will scale effectively with a growing vocabulary size and a limited supply of training data for signer-independent applications. Due to its sensitivity to shape information, automated SLR based on hidden Markov models (HMMs) cannot characterize the confusing distributions of the observations in gesture features with sufficiently precise parameters. In order to simulate uncertainty in hypothesis spaces, many scholars provide an extension of the HMMs, utilizing higher-order fuzzy sets to generate interval-type-2 fuzzy HMMs. This expansion is helpful because it brings the uncertainty and fuzziness of conventional HMM mapping under control. The neutrosophic sets are used in this work to deal with indeterminacy in a practical SLR setting. Existing interval-type-2 fuzzy HMMs cannot consider uncertain information that includes indeterminacy. However, the neutrosophic hidden Markov model successfully identifies the best route between states when there is vagueness. This expansion is helpful because it brings the uncertainty and fuzziness of conventional HMM mapping under control. The neutrosophic three membership functions (truth, indeterminate, and falsity grades) provide more layers of autonomy for assessing HMM's uncertainty. This approach could be helpful for an extensive vocabulary and hence seeks to solve the scalability issue. In addition, it may function independently of the signer, without needing data gloves or any other input devices. The experimental results demonstrate that the neutrosophic HMM is nearly as computationally difficult as the fuzzy HMM but has a similar performance and is more robust to gesture variations.

Keywords: sign language recognition; neutrosophic reasoning; high order fuzzy logic; decision making under uncertainty; context-based classification

check for updates

Citation: Al-Saidi, M.; Ballagi, Á.; Hassen, O.A.; Saad, S.M. Cognitive Classifier of Hand Gesture Images for Automated Sign Language Recognition: Soft Robot Assistance Based on Neutrosophic Markov Chain Paradigm. *Computers* **2024**, *13*, 106. https://doi.org/10.3390/computers13040106

Academic Editors: Jeremy Straub, Hussain Mohammed Dipu Kabir, Syed Bahauddin Alam and Subrota Kumar Mondal

Received: 28 February 2024
Revised: 9 April 2024
Accepted: 16 April 2024
Published: 22 April 2024

1. Introduction

The SLR aims to provide a reliable method for transcribing sign language into written or spoken form, facilitating communication between deaf people and those with normal hearing [1]. Automatic gesture recognition is very engaging since hand gestures are spatiotemporally changing. The primary difficulty in sign language recognition is finding a model that can grasp the language and scale up to significant terms [2,3]. It is possible to achieve gesture recognition in one of two ways [4]: either by wearing sensual gloves on one's hands or with the aid of computer vision.

Glove-based approaches use sensory gloves to measure the angles and spatial positions of a hand and fingers. Computer-vision-based approaches to hand gesture detection need either a single camera or many cameras to acquire images. Using computer-generated gloves or other resources as input devices produces difficulties in distinguishing gestures from a typical standpoint and it is particularly hard to distinguish them in real-time. Thus, in recent years, researchers have made several SLR systems using the available computer vision methods [5,6]. Regular Human–Computer Interaction (HCI) should be glove-free, rapid, reliable, and acceptable.

In the literature, the standard hand-communication systems may be broken down into three layers: detection, tracking, and recognition [7]. The task of the detection layer is to identify and extract visual features that may be linked to the presence of hands in the camera's view; for the system to know "what is where" at all times, it is up to the tracking layer to establish temporal data links between successive frames of images. The recognition layer is responsible for integrating the collected spatiotemporal data from the previous layers and communicating the resulting clusters, which are annotated with labels related to certain types of gestures/postures.

Building a hand gesture detection and recognition system involves several main steps. A high-level overview is provided as follows (see Figure 1) [1–4]: (1) Data Collection: Gather a diverse dataset of hand gesture images. These should cover different hand poses, lighting conditions, backgrounds, and skin tones. (2) Preprocessing: Preprocess the collected data to improve their quality and suitability for training. This may involve tasks such as resizing, normalization, noise reduction, and background subtraction. (3) Hand region detection: This involves isolating the region of an image that contains the hand(s). Several techniques can be used for hand region detection, such as skin color segmentation. (4) Feature Extraction: Features relevant to hand gestures are extracted from the preprocessed data. This may include extracting hand regions, a contour analysis, a histogram of oriented gradients (HOG), or other deep-learning-based feature extraction techniques. (4) Gesture Detection and Recognition: A machine learning model is employed to detect and recognize hand gestures based on the extracted features. This involves classifying the hand gestures into predefined categories. (5) Output/Feedback: The recognized gestures are conveyed as output, which can be used for various purposes, such as controlling devices, interacting with applications, or providing feedback to the user.

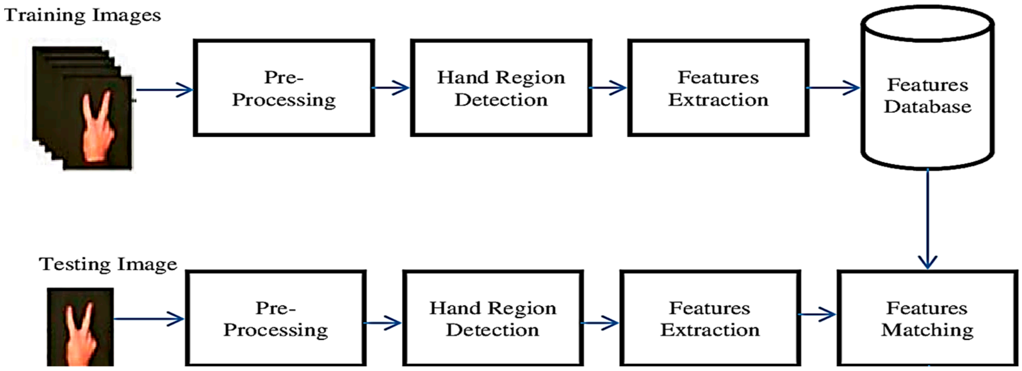

Figure 1. Schematic diagram of a hand gesture recognition system.

Different types of SLR may be broken down further into signer-related (dependent) and signer-free (independent) subcategories, depending on whether or not the signer is actively involved in the identification process. Signer-free SLR is crucial because it enables the direct application of a technique to the package and the development of the method for signers who were previously unrecognized [8,9]. There are two issues with signerless SLR [7]: There is a noticeable divergence in the signals (postures) of different people, which increases the model convergence complexity. In order to obtain a robust recognition paradigm, the training data must constitute a large number of signers. The second issue concerns the inadequate signatures (descriptions) extracted from several signers' data. The investigation into the drawing out of an SLR signature is only just beginning, as opposed to speech recognition, where each voice signature has been exhaustively searched. Properly eliciting public signatures from varied postures is a challenging issue that demands a solution [8].

Several methods have been employed for signer-independent SLR, with HMM being the most popular option because of its many benefits [8,9]. HMM provides the capacity to simulate a time-domain process when taking into consideration the positions and orientations of gestures throughout time. However, there are limitations to the standard HMM. The assumption that a mixture of Gaussian functions may adequately characterize the allocations of different control elements is one such limitation. The scalability of HMMs is another major limitation when using them for gesture recognition. There are hundreds of symbols in the SLR dictionary. It is impractical and time-consuming to train and test thousands of HMMs. As a result of these two issues with signer-free SLR, it is clear that conventional HMM cannot adequately create posture descriptions.

The incomplete information and non-specificity caused by gesture vagueness are two examples of the types of vagueness in gesture recognition, which include noisy or non-stationary data, an uncertain matching degree, and fuzzy similarity matches, in addition to fuzzy decision functions [10]. When the variables of the HMM are stated, they are fully identified. As a consequence of incomplete or noisy data in sign recognition, these factors might not precisely show the primary distribution of the examinations. Also, the use of hiring probabilities to evaluate uncertain HMMs might look complicated based on the observation. It is still possible to describe the ambiguous factors of the HMM to account for the uncertain likelihoods, even though this does not create a dangerous problem for different uses. Generalized HMM with fuzzy measure and a fuzzy integral (fuzzy hidden Markov model) is a useful method. At this point, many researchers have used Fuzzy HMM to recognize posture and obtained great results. Even so, these limits are still in place [11–13].

Recently, the neutrosophic hidden Markov model (NHMM) has been employed as a significant mathematical mode for uncertainty, redundancy, inconsistency, and ambiguity. The neutrosophic variable (NV) is subject to change due to randomness and indeterminacy, in contrast to the conventional stochastic (random) variable, which accounts only for the changes due to randomness [14]. The neutrosophic probability (NP) of an event E is defined as $NP(E) =$ (the chance that the event may occur, the indeterminate chance of the event occurring, and the chance that the event may not occur). NHMM explicitly quantifies indeterminacy. Truth, indeterminacy, and falsity are independent.

1.1. Motivation

Researchers in the field of gesture recognition are driven by the fact that these gestural communication and device control methods are more commonplace. Numerous applications now rely on camera-based gesture detection technologies, allowing for two-way communication between humans and computers. Both the feature and hypothesis spaces in gesture/posture categorization involve ambiguities, and HMM-based approaches can accurately characterize these data with sufficient training data. Practically, the HMM still performs insufficiently on the test data for the reasons of noise, insufficient training data, and incomplete information. Therefore, in HMM, ambiguity (uncertainties) must be materialized. In order to improve performance in terms of resilience with postural anomalies, generalization capacity, and recognition accuracy, we combine high-order fuzzy sets (a neutrosophic set in our case) with HMM classifiers. The realization of the neutrosophic set was essentially qualified through three membership functions (truth, indeterminacy, and falsity) to manage more vagueness in real-world applications.

1.2. Contribution and Novelty

The work presented in this paper proposes a novel Neutrosophic-based Hidden Markov Model (NHMM) to address the issues with signer-independent SLR. It is motivated by the desire to provide users with an instinctive gesture input system that performs well on a large vocabulary size with a small set of training data under conditions of uncertainty. The classification of hand gestures involves ambiguities in both the feature space and the hypothesis space. Typically, class-conditional probability density functions in the feature space allow for the extraction of arbitrary observations. The variables that make up the decision function in the hypothesis space are random variables whose prior distributions were obtained using training data. Due to their exclusive focus on whether an element is a member of a particular class or not, fuzzy logic controllers and related logics do not incorporate the idea of a range of neutralities. As a result, these logics are ill-equipped to handle any uncertainty that may occur during data collection. Consequently, when the obtained data may be uncertain or insufficient, a neutrosophic controller is proposed as a solution.

According to recent reviews, neutrosophic logic with HMM classifiers has not been adjusted for SLR tasks, despite its use in many decision functions' tasks; hence, the current study offers something new in this area. The system is able to simulate the different changes in hand postures by recognizing a posture based on the structure of hand features. Considered a valuable matrix decay approach, particularly for data exploration and dimensional-decreasing translations, the observation sequence utilized to characterize the states of the NHMM is derived from the features extracted from the segmented hand image using Singular Value Decomposition (SVD).

The remainder of this paper consists of the following sections: Section 2 provides a literature review of relevant publications regarding the SLR framework. The suggested approach is presented in Section 3. The assessment of the suggested technique, including results and discussion, is presented in Section 4. The study is concluded, and possible future directions are discussed, in Section 5.

2. State of the Art

Hand gesture recognition is often divided into two categories [4,15–23]: (1) Rule-based techniques, which include a set of rules that are manually configured among feature inputs. The rule that balances the input is output as the gesture after a set of features are retrieved and matched to the encoded rules. The fundamental flaw of rule-based systems is their reliance on human encoding expertise. (2) Machine-learning-based techniques that manage a gesture as the result of a stochastic process. Among these methods, HMMs have been extensively studied for their ability to classify gestures. On the other hand, accurate and real-time SLR with an extensive vocabulary still faces several obstacles. If the number of expressions increases, the hybrid HMM model will not be able to scale finely enough to handle them. Still, the vast majority of such systems are SLRs that rely on signers.

A deep-learning-based method that can recognize and classify words from a person's gestures is proposed in ref. [24]. Finding a suitable model to deal with the problems of continuous signer-independent signs is the tricky part. Building a continuous and accurate model is problematic for various reasons, including the fact that signers' rates and durations vary. Two distinct deep-learning-based methods are used in this work for the purpose of effective and accurate sign identification. The authors generate a single hybrid algorithm after training both methods independently for static and continuous signs. To obtain a hand's feature data, the Leap Motion sensor is used in ref. [25]. A hybrid HMM-DTW model is suggested for dynamic gesture recognition after evaluating the recognition effects of HMM and dynamic time warping (DTW) algorithms. It is clear that the model is significant based on the Kappa coefficient.

In ref. [26], the authors used a well-defined stochastic mathematical technique to build a sophisticated hand motion detection system for use in VR applications. First, the user and system do not make physical contact; second, geometric factors may dictate the hand gesture's rotation; and third, to increase measurement performance, the model parameter must be adjusted. The suggested study hybridizes a bio-inspired metaheuristic approach—specifically, the cuckoo search algorithm—to reduce the complicated trajectory in the HMM model, and this is compared to existing classification strategies used in hand gesture detection. The experimental results show how to increase system performance metrics by validating the HMM model using the optimizer's cost value.

The use of hidden Markov models as classifiers was confirmed in the method for identifying chosen human–computer interaction (HCI) gestures that was published in ref. [27]. Using a system developed for recognizing VR gaming gestures, the studies looked at the possibility of applying it to the classification of HCI gestures. In this study, the authors evaluated several approaches to discretizing forearm orientation data and compared their classification accuracy.

The study in Ref. [28] presented an autonomous method for simultaneous and delay-free hand gesture detection and prediction by combining Hidden Markov Models (HMMs) with Deep Neural Networks (DNNs). A hidden Markov model (HMM) may be used for feature extraction, a forward spotting mechanism with variable sliding window sizes can be utilized to identify the meaning of gestures, and finally, deep neural networks can be employed for recognition. Hence, to reliably detect a significant number of gestures (0–9), a stochastic approach to building a non-gesture model using HMMs and no training data was presented. The authors use the confidence measure from the non-gesture model as an adaptive threshold to find the beginning and end of the input video stream, where meaningful gestures start and stop. The experimental data show that their system has a 94.70% success rate in detecting and predicting significant movements.

For the purpose of hand gesture recognition, a multistage spatial-attention-based neural network has been suggested in ref. [29]. The suggested paradigm has three levels, with each step being passed down from the CNN. Before highlighting the dataset's practical features, the authors apply a feature extractor and a spatial attention module, utilizing the original dataset's self-attention. Then, they multiply the feature vector with the attention map. After that, they look at features that might be combined with the initial dataset to achieve modality feature embedding. They used the same approach in the second stage to build an attention map and feature vector using the feature extraction architecture and self-attention method. In the third step, a classification module is used to predict the label of the corresponding hand motion.

For skeleton-based hand gesture identification, ref. [30] research recommended a deep ensemble architecture known as the multi-model ensemble gesture recognition network (MMEGRN). For the purpose of extracting and categorizing skeleton sequences, the authors offered an architecture that makes use of four sub-networks and three spatiotemporal feature classifiers. Late feature fusion creates a new fusion classifier by merging the features obtained from each sub-network's feature extractor. The usefulness and superiority of the proposed framework compared to previous models in terms of recognition accuracy were shown via extensive testing on three skeleton-based hand gesture recognition datasets.

In order to assess the temporal hand trajectories derived from the identified hand poses, the segmentation process in ref. [31] employs the adaptive-region-based active contour (ARAC) method, which is based on the meta-heuristic foundation of the opposition strategic velocity updated beetle swarm optimization (OSV-BSO). To reduce dimensionality, principal component analysis (PCA) is used after feature extraction. Standard features used in this process include oriented fast and rotated brief (ORB), region props, and the histogram of oriented gradients (HOG). An optimized probabilistic neural network (PNN) uses these essential features to complete the recognition phase. The authors adjust the training weights based on the meta-heuristic basis of the OSV-BSO to enhance the present performance of PNN as a deep learning model.

To recognize hand gestures by extracting all conceivable types of skeleton-based gestures, the authors in ref. [32] presented a multi-branch attention graph and a generic deep-learning model. Using a generic deep neural network module, the final branch collects features based on general deep learning. Concatenating the spatial, temporal, and general features and putting them into the fully connected layer resulted in the final feature vector. The suggested technique surpassed the current state-of-the-art methods due to its cheap computing cost and excellent performance accuracy.

A lightweight deep neural network with enhanced processing was proposed for real-time dynamic sign language recognition (DSLR) in ref. [33]. There are three primary parts of the solution approach. Dataset preprocessing involves first standardizing the number of frames in the incoming dataset. Then, in order to find and identify them, the MediaPipe framework takes hand and position landmarks (features). In order to correctly identify the DSL, the features of the models are passed on after processing the body's depth and position unification. The study results demonstrate that this solution approach can identify dynamic signs in real time with remarkable speed and accuracy.

To recognize the static gestures of the hands in sign language, a novel architecture of deep learning neural networks is presented in ref. [34]. Classical, non-intelligent feature extraction and convolutional neural networks (CNNs) are part of the suggested architecture. Following background removal and preprocessing, the suggested structure uses three streams of feature extraction to extract useful features and assign a class to the gesture image. Three commonly used approaches in hand gesture classification—CNN, Gabor filter, and the Oriented Fast and Rotated Brief (ORB) feature descriptor—make up these three streams, each of which extracts its own distinctive features. The final feature vector is formed by merging these features. Not only does the suggested structure become more resistant to uncertainties and ambiguity in the hand gestures, it also achieves very high accuracy in hand gesture classification by combining these efficient approaches.

In ref. [35], the authors introduced a portable convolutional neural network (CNN), called a hybrid feature attention network (HyFiNet), for accurate hand gesture identification. The architecture of HyFiNet consists of four REEMs or multi-scale refined edge extraction modules. The REEM module incorporates a hybrid feature attention (HyAttention) block to collect the finer edge information of hand motions. In order to learn the discriminable semantic structure of hand postures, the HyAttention block is designed to zero in on efficient key features from multi-receptive fields. When compared to the current state-of-the-art networks, the results of the study and visual representation show a significant improvement in accuracy.

In ref. [36], the authors detail their work on a computer vision system that uses Hidden Markov Models (HMMs) to identify hand gestures. After receiving video input of hand gestures, the system performs morphological operations, segmentation of the hands, tracking of the hands, trajectory smoothing, and skin-color-based segmentation. The HMM learning package is used to implement the HMM with Gaussian emissions. The Viterbi technique is employed to decode observation sequences and ascertain the most probable state sequences, together with their corresponding probabilities. A validation set is used to examine the maximum likelihood classifier's sign recognition accuracy, while a test set of observation sequences is used to evaluate the system's performance. This system's ability to correctly recognize hand gestures and their matching letters is confirmed by the results. The reader can refer to refs. [37–39] for a comprehensive study on the challenges and solutions in hand gesture recognition.

The Need to Extend the Related Work

Using a large dataset to identify a few isolated signs under controlled illumination, many scholars presented an automated SLR system based on HMM. Backgrounds and visuals were severely constrained throughout the data collection phase. Given the behavior complexity and uncertainty of hand movement changes in SLR, this paper's primary objective is to investigate the efficacy of neutrosophic HMM in signer-independent SLR with the aim of improving the scalability of SLR systems, with the ultimate goal of processing more hand signs and predicting more correct models with different types of uncertainty, unlike current methods, which do not deal with all cases of uncertain data. Not all uncertain (indeterminate) data can be represented by high-order fuzzy sets, while neutrosophic sets deal with all types of indeterminacy.

3. Methodology

With a large enough vocabulary size, the suggested posture recognition system may be used in real-world scenarios. It is a vision-based method for identifying isolated signs in posture images without any constraints on the signer or the background. By using neutrosophic logic, this technique enhances the classifier ability of HMMs concerning uncertainty. Neutrosophic sets, one of the fuzzy set extensions, have the ability to deliver more successful results when modeling uncertainty since they contain the membership functions of truth, indeterminacy, and falsity definitions, rather than only a membership function, thus providing a fair estimate of the reliability of the information. The independence of truth-membership (TM) and indeterminacy-membership (IM) functions from each other and expression of the fact that an individual does not have full control of the issue with the IM function have an essential role in modeling uncertainty problems [40]. The left–right non-skip HMM topology is often used to simulate posture signals. In this topology, states with marks lower than the current state cannot be entered via switches. The neutrosophic hidden Markov model's (NHMM) parameters are defined using a neutrosophic probability, which is determined by the training samples. Based on neutrosophic numbers, the training is carried out using the Baum–Welch method. To determine the most likely expression sequence, the recognition phase employs the Viterbi technique.

The many advantages of the proposed scheme include the following: the features are unaffected by the scaling or revolution of gestures under different conditions inside an image, making the system more adaptive, and Singular Value Decomposition (SVD) minimizes the feature vector required for a particular posture image. In addition, the feature vector that is unique to each motion is formed by features that are constructed using this approach. Real-time online SLR posture detection and interactive training are both possible with this technology. It is also capable of learning new postures intelligently, which may be used for recognition purposes in the future. The proposed SLR system is detailed below, and Figure 2 shows how it works.

3.1. Hand Detection

Due to limitations in vision-based methods, such as changing illumination conditions and composite image backgrounds, distinguishing hands is essential to success in any gesture classification effort. Two primary methods exist for skin recognition: color-based and region-based division [41,42]. Due to the difficulties encountered while recognizing skin colors and the unpredictability of illumination conditions, skin-color identification could be more reliable. There must be a sufficient contrast between the object and background for methods that use shape outlines (region-based) to work. Considering the trade-off between computational effort and identification accuracy, the proposed system employs a YCbCr color space pixel-based, non-parametric (histogram) segmentation approach [43]. The main steps include computing the histogram, selecting thresholds, applying the thresholds to segment the image, and optionally performing post-processing for refinement (morphological operations). In order to classify skin pixels and provide robust parameters under different lighting situations, the color space transformation is intended to reduce the overlap between skin and non-skin pixels [44].

Let I be a digital posture image, represented as a matrix of pixel intensities $I(x,y)$ for $x = 1,\ 2, \ldots M$ and $y = 1,\ 2, \ldots N$, where M and N are the dimensions of the image. The histogram H of image I is defined as a discrete function:

$$H(i) = \sum_{x=1}^{M} \sum_{y=1}^{N} \delta(I(x,y) - i) \tag{1}$$

where δ is the Dirac delta function and i is the intensity level. The histogram represents the frequency distribution of pixel intensities in the image. Then, the image is segmented into K + 1 regions (two region in our case) $R_0, R_1, \ldots\ldots, R_k$, where R_i consists of pixels with intensities in the ith interval, where T_{i-1} and, T_i are the lower and upper bounds (inclusive–exclusive) of intensity level i, respectively.

$$R_i = \{(x,y)|I(x,y) \in [T_{i-1},\ T_i)\} \tag{2}$$

The input in this approach may be an image or a video frame obtained from any camera, including webcams. The input RGB color image is first downsized to save processing time and then converted to a CbCr chrominance image (chrominance vector). In order to mitigate the effects of lighting variation, the system uses only the chrominance matrices to fully describe the color data, ignoring the Y matrix [44]. Building a decision process that can differentiate between pixels that contain skin and those that do not is the last objective of skin color detection. One way to create a skin classifier is to define the skin cluster's borders in the CbCr color space using a set of rules. Refer to [45,46] for a comprehensive overview of skin clusters.

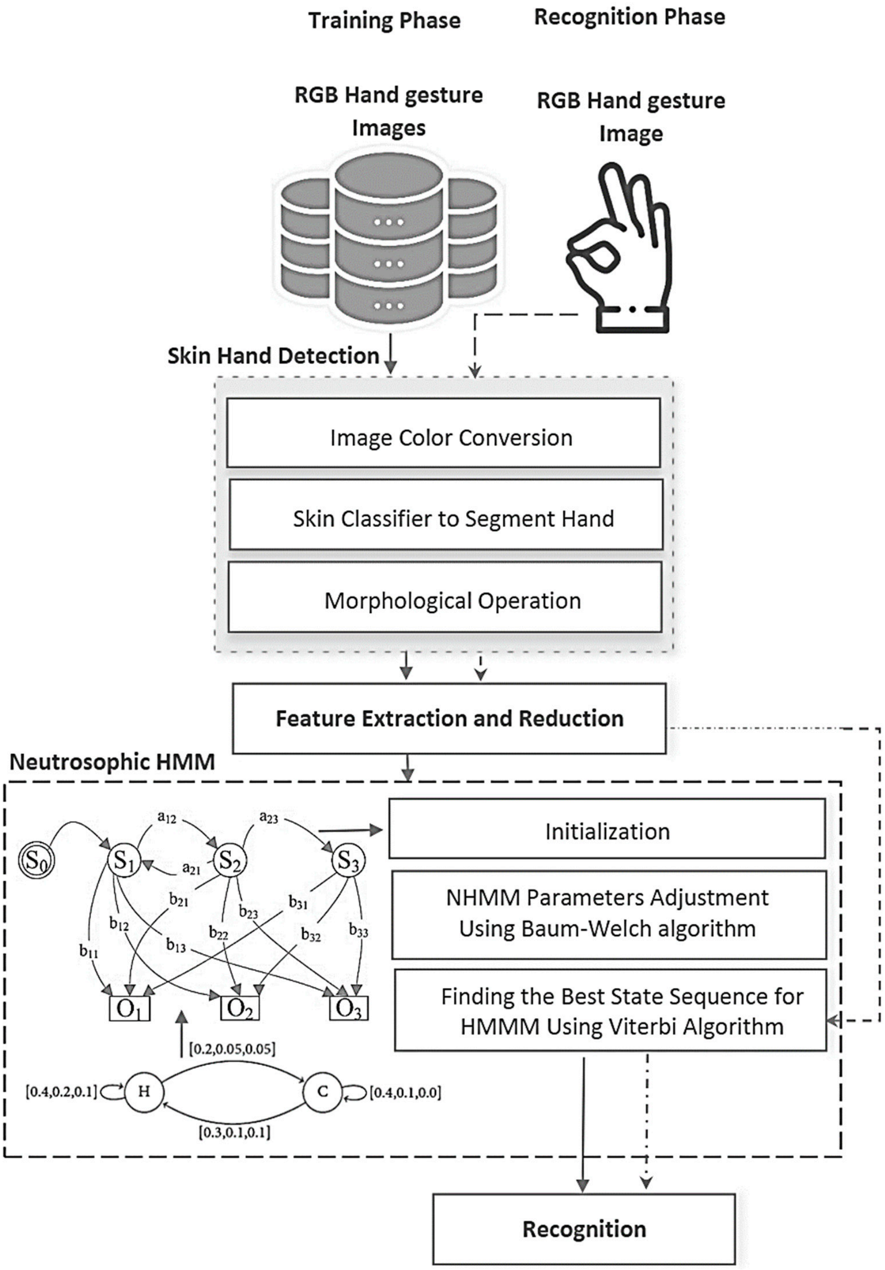

Figure 2. Schematic diagram of proposed neutrosophic HMM-based hand gesture recognition system.

3.2. Feature Extraction

Because hand motions are so varied in form, movement, and texture, choosing features is critical to gesture detection. The following three methods are the most common ones that may produce features: viewpoint, model (Kinematic Model), and appearance-based low-level feature approaches. Refer to [47–50] for a comprehensive overview of the use of gesture feature extraction for static gesture recognition. In order to improve the accuracy of the hand posture identification scheme and to circumvent issues caused by multiple modifications like rotating, resizing, and reallocating, the SVD feature extraction method is applied to the skeleton of the hand shape using the fewest number of pixels without losing the contour data in order to identify unique and distinguishable features [50]. This is achieved using the binary segmented hand image from the previous step.

SVD helps to reduce the feature space's dimensionality, which can improve computational efficiency and reduce the risk of overfitting. The singular vectors obtained from SVD can provide insights into the principal components of the data, aiding in understanding the most influential features of gesture recognition [51–53]. For any $m \times n$ array A, there exist orthogonal arrays, as follows:

$$A = U\sum V^T \tag{3}$$

$$U = [u_1, u_2, \ldots, u_m] \in \mathcal{R}^{m\times m} \tag{4}$$

$$V = [v_1, v_2, \ldots, v_n] \ \in \mathcal{R}^{n\times n} \tag{5}$$

where

$$\sum = diag\left(\sigma_1, \sigma_2, \ldots, \sigma_{\min(m,n)}\right) \in \mathcal{R}^{m\times n},$$
$$\sigma_1 \geq \sigma_2, \geq \ldots \geq \sigma_{\min(m,n)} \geq 0 \tag{6}$$

σ_i is the ith unique quantity of A in a non-increasing sequence; u_i and v_i are the ith left and right unique vectors of A for $i = min(m,n)$, in that order. The ith largest unique quantity σ_i of A is really the Euclidean distance of the ith major estimated vector Ax in the x direction, which is perpendicular to all the $i-1$ superior perpendicular vectors:

$$\sigma_i = \max_{U} \min_{x\in U, \|x\|_2=1} \|Ax\|_2 \tag{7}$$

$$\sigma_1 = \max_{\|x\|_2=1} \|Ax\|_2 \tag{8}$$

The highest is engaged over all i-dimensional areas $U \subseteq \mathcal{R}^n$. In this case, the HMM observable vectors consist of $\sigma_1, \sigma_2, \ldots, \sigma_{\min(m,n)}$ coefficients stemmed by the operating SVD transform for each training sign. The number, and particularly the order, of the coefficients play a significant role in producing an acceptable recognition model [54].

3.3. Neutrosophic Hidden Markov Model

Hidden Markov Models (HMMs) are a powerful tool for modeling sequential data, making them well-suited for gesture recognition tasks where the temporal aspect of the data is crucial [55–57]. Handling uncertainty in Hidden Markov Models (HMMs) is crucial for many real-world applications, as it allows the model to account for the inherent ambiguity and variability in the observed data. Combining neutrosophic logic with HMMs introduces a framework for modeling uncertainty, indeterminacy, and imprecision in the context of sequential data analysis. This combination is referred to as Neutrosophic Hidden Markov Models (NHMMs) [58,59].

A Markov chain is a sequence of random variables $X = \{X_0, X_1, \ldots\ldots\}$ with the following properties: for $n = \{0, 1, 2, \ldots\}$, X_n is defined on the sample space $\mathcal{F}$ and takes values from the finite set S. Thus, $X_n : \mathcal{F} \longrightarrow S$. Also, for $n = \{0,\ 1,\ 2, \ldots\ldots\}$ and $\{i, j,\ i_{n-1},\ i_{n-2}, \ldots\ldots,\ i_0\} \subseteq S$,

$$P\{X_{n+1} = j|\ X_n = i, X_{n-1} = i-1, X_{n-2} = i-2,\ \ldots\ldots X_0 = i_0\} = P\{X_{n+1} = j|\ X_n = i\} \quad (9)$$

The transition probabilities $P\{X_{n+1} = j|X_n = i\} = P_{ij}$ are independent of n [58]. A Neutrosophic Markov Chain is a sequence of random neutrosophic variables $X_1,\ X_2,\ X_3, \ldots\ldots$ where the next neutrosophic state depends only on the current state.

$$NP\{X_{n+1} = j|X_n = i, X_{n-1} = i-1, X_{n-2} = i-2,\ \ldots, X_0 = i_0\} = P\{X_{n+1} = j|\ X_n = i\} \quad (10)$$

NP is the neutrosophic probability in which each event is associated with three values: T (truth), F (falsehood), and I (indeterminacy). These values represent the degrees of truth, falsehood, and indeterminacy associated with an event.

An interval neutrosophic stochastic process $\{X(n) : n \in N\}$ is said to be an interval neutrosophic Markov chain if it satisfies the Markov property.

$$\psi\{X_{n+1} = j|X_{n-1} = i, X_n = k,\ \ldots, X_0 = m\} = \psi\{X_{n+1} = j|\ X_{n-1} = i\} \quad (11)$$

where i, j, and k establish the state space S of the process. Here, $\tilde{P}_{i,j} = \psi(X_{n+1} = j|X_n = i)$ is called the interval-valued neutrosophic probabilities of moving from state i to state j in one step.

$$\tilde{P}_{i,j} = \left(\left[T^L_{\underset{\sim}{P}_{i,j}}, T^U_{\underset{\sim}{P}_{i,j}}\right], \left[I^L_{\underset{\sim}{P}_{i,j}}, I^U_{\underset{\sim}{P}_{i,j}}\right], \left[F^L_{\underset{\sim}{P}_{i,j}}, F^U_{\underset{\sim}{P}_{i,j}}\right]\right) \quad (12)$$

where $T^L_{\underset{\sim}{P}_{i,j}}, T^U_{\underset{\sim}{P}_{i,j}}$ are the lower and upper truth-membership of the transition from state i to state j, respectively, $I^L_{\underset{\sim}{P}_{i,j}}, I^U_{\underset{\sim}{P}_{i,j}}$ are the lower and upper indeterminate membership of the transition from state i to state j, respectively, and $F^L_{\underset{\sim}{P}_{i,j}}, F^U_{\underset{\sim}{P}_{i,j}}$ are the lower and upper falsity-membership of the transition from state i to state j. Matrix $P = \tilde{P}_{i,j}$ is called the interval-valued neutrosophic transition probability matrix.

The Neutrosophic Hidden Markov Model (NHMM) is a neutrosophic Markov chain $\{P_n\}_{1 \le n \le N}$ whose states are unobservable directly but can be observed through a sequence of observations $\{O_n, n \ge 1\}$, generated by the state X_n, which is only conditional on X_n. The NHMM is similar to a fuzzy hidden Markov chain [60,61], where the arithmetic operations are neutrosophic. The NHMM, like any other HMM, consists of the initial NP distribution, neutrosophic transition probability matrix, and the observation matrix providing the conditional NP:

$$NP(v_k\ at\ n|X_n = j) \quad (13)$$

where $v_k \in V$ is the set of observations $V = \{v_1, v_2, \ldots, v_m\}$. The main problem of any hidden Markov chain involves the decoding part, which finds the best state sequence $\{X_n\}_{1 \le n \le N}$ given the model $\overset{\sim}{\lambda}$ and the observation sequence of length N. Traditionally, the Viterbi algorithm is used to solve this problem with the aim of finding $\overset{\sim}{\gamma}_n(i)$, which provides the highest possibility along the single path at time step n. See [62] for more details.

$$\overset{\sim}{\gamma}_n(i) = \frac{max}{X_1, X_2, \ldots, X_{n-1}}\ \sigma\left(X_1, X_2, \ldots, X_n = i, O_1, O_2, \ldots, O_n \middle| \overset{\sim}{\lambda}\right), 1 \le i \le s, 1 \le n \le N. \quad (14)$$

Here is how the NHMM could be applied to gesture recognition [14,58,59]:

- Neutrosophic Representation of Gestures: Represent different possible gesture states using neutrosophic sets. Each state could have associated truth-membership, indeterminacy-membership, and false-membership values, capturing the uncertainty in recognizing a particular gesture. For observations, capture the uncertainty in detecting and interpreting individual features or components of gestures using neutrosophic sets. This accounts for variations in the observed data due to noise or imprecision.

In hand gesture recognition, neutrosophic membership functions can be used to model the uncertainty associated with recognizing gestures. Let us consider a simple example, where we have three possible gestures: "open hand", "closed fist", and "pointing finger". We can define neutrosophic membership functions for each gesture based on the degrees of truth, indeterminacy, and falsity and represent these membership functions using triangular functions, where the parameters defining the functions (e.g., the peaks, widths, and slopes) are determined based on the specific characteristics of each gesture and the available data. For example, for the "open hand" gesture, the membership functions could be defined as follows: Truth (T): The hand is fully open, so the truth membership function peaks at one and decreases as the observed hand gesture deviates from the fully open hand. Indeterminacy (I): There might be some ambiguity if the hand is partially open or partially closed, so the indeterminacy membership function has a peak around the intermediate state. Falsity (F): The hand is not in a closed fist or pointing gesture, so the falsity membership function peaks at 0 and decreases as the observed hand gesture becomes more similar to a closed fist or pointing finger.

Similarly, membership functions for other gestures can be defined based on their characteristic features and the degree of ambiguity associated with their recognition. Here is a simple triangular representation for each membership function [14]:

$$For\ Truth\ (T) : T(x) = \begin{cases} 0 & x \le a \\ \frac{x-a}{b-a} & a < x < b \\ \frac{c-x}{c-b} & b \le x \le c \\ 0 & x \ge c \end{cases} \tag{15}$$

$$For\ Indeteminacy\ (I) : I(x) = \begin{cases} 0 & x \le d \\ \frac{x-d}{e-d} & d < x < e \\ \frac{f-x}{f-e} & e \le x \le f \\ 0 & x \ge f \end{cases} \tag{16}$$

$$For\ Falsity\ (F) : F(x) = \begin{cases} 0 & x \le g \\ \frac{x-g}{h-g} & g < x < h \\ \frac{i-x}{i-h} & h \le x \le i \\ 0 & x \ge i \end{cases} \tag{17}$$

where $a < b < c$ are the parameters defining the truth membership function: $d < e < f$ for indeterminacy and $g < h < i$ for falsity. More complex functions can be used depending on the complexity of the available hand gesture samples.

The inference engine processes these data to make decisions or inferences, considering the uncertainties inherent in the data. Let A be a set representing a piece of information. We denote $\tilde{A}$ as a neutrosophic set $\tilde{A} = \{(x, T(x), I(x), F(x)|x \in U)\}$ where x represents an element in the universe of discourse U, and $T(x), I(x),$ and $F(x)$ represent the truth, indeterminacy, and falsity degrees associated with x, respectively. The inference engine operates based on a set of rules R, where each rule r_i is defined as a triplet (A_i, B_i, C_i), where A_i, B_i, and C_i are neutrosophic sets representing the antecedent, consequent, and indeterminate parts of the rule, respectively. A_i is a neutrosophic set defining the conditions or criteria that must be satisfied for the rule to apply. B_i is a neutrosophic set defining the outcomes or conclusions that follow if the antecedent is satisfied. C_i is a neutrosophic set capturing the degree of indeterminacy associated with the rule, indicating the uncertainty or ambiguity in the inference process.

Given an input neutrosophic set X, the inference engine evaluates each rule against X to derive the degree of support for each consequence. This process involves determining the intersection between the input set X and the antecedent A_i of each rule r_i, resulting in a new set Y_i. The degree of support for the consequent B_i is then determined based on the truth, indeterminacy, and falsity degrees of the elements in Y_i. If multiple rules provide conclusions, the engine aggregates these conclusions to derive a final decision or inference. This aggregation process may involve combining the degrees of support for each consequent using aggregation operators such as max, min, or weighted sum. Finally, based on the aggregated conclusions, the engine generates an output classification, decision, or inference, along with associated measures of confidence or uncertainty.

- Neutrosophic Transition Probabilities: Define the transition probabilities between neutrosophic gesture states. These transition probabilities should reflect the uncertainty in transitioning from one gesture state to another. For example, transitioning from a "hand raised" state to a "hand lowered" state may have some indeterminacy due to variations in the gesture execution.
- Neutrosophic Emission Probabilities: Associate neutrosophic emission probabilities with each gesture state. These probabilities represent the likelihood of observing specific features or components given the current gesture state. Again, this accounts for uncertainty in the observed data.
- Learning and Inference: Train the NHMM using a dataset of neutrosophic represented gestures. Learning involves estimating the neutrosophic parameters, such as transition and emission probabilities, from the training data. During inference, the NHMM can be used to recognize and classify new gestures. The model considers the uncertainty associated with each aspect of the gesture and provides a more nuanced understanding of the gesture recognition process.

A. Training phase

Given observable sequences for each training set, NHMM can be prepared with the neutrosophic form of the Baum–Welch algorithm. The Baum–Welch algorithm, also known as the forward–backward algorithm, is an expectation-maximization (EM) algorithm used for training hidden Markov models HMMs (tuning the parameters of the HMM). Adapting the Baum–Welch algorithm for an NHMM involves handling neutrosophic sets and updating the parameters accordingly [63,64]. Here is a high-level overview of the adaptation of the Baum–Welch algorithm for neutrosophic HMMs:

1. Initialization: initialize the parameters of the NHMM, including initial state neutrosophic measures π, transition neutrosophic measures A, and emission neutrosophic measures B.
2. Forward Pass (Forward Algorithm): Compute the forward neutrosophic probabilities using the current model parameters. This involves propagating the neutrosophic measures through the states and time steps of the NHMM.
3. Backward Pass (Backward Algorithm): Compute the backward neutrosophic probabilities, representing the probability of observing the remaining sequence given the current state of neutrosophic measures at a specific time.
4. Expectation Step: Use the forward and backward neutrosophic probabilities to compute the expected number of transitions and emissions for each state. This step involves calculating the expected neutrosophic counts based on the current model parameters.
5. Maximization Step: Update the model parameters (transition neutrosophic measures, emission neutrosophic measures) using the expected counts obtained in the expectation step. This step involves maximizing the likelihood of the neutrosophic observed data given the model.
6. Iterative Refinement: Repeat steps 2–5 until convergence or until a specified number of iterations is reached. Convergence can be determined by monitoring the change in the neutrosophic log-likelihood between iterations.
7. Final Model: The final model parameters obtained after convergence represent the trained neutrosophic HMM.

Given the model $\lambda = (A, B, \pi)$, the Baum–Welch algorithm trains the NHMM on a fixed set of observations $O = (O_1, O_2, \ldots\ldots, O_T)$. By adjusting its parameters, A, B, and π, the Baum–Welch algorithm maximizes $P(O; \lambda)$. The parameters of the Baum–Welch algorithm are updated iteratively (for $i, j = 1, 2, \ldots, N$ and $t = 1,\ 2, \ldots\ldots, T - 1$), as follows:

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O; \lambda)} \tag{18}$$

$$\gamma_t(i) = \sum\nolimits_{j=1}^{N} \xi_t(i,j) \tag{19}$$

$$\alpha_t(i) = \left[\sum\nolimits_{j=1}^{N} \alpha_{t-1}(j) a_{ji}\right] b_i(O_t) \tag{20}$$

$$\beta_t(i) = \sum\nolimits_{j=1}^{N} a_{ij} b_i(O_{t+1}) \beta_{t+1}(j) \tag{21}$$

$\xi_t(i,j)$ is the probability of a path reaching state q_i at time t and transitioning to state q_{ij} at time t + 1. Summing a set of observations, we can obtain values for the expected number of transitions from or to any arbitrary state. Thus, it is straightforward to update the NHMM parameters:

$$\acute{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{22}$$

$$\acute{b}_j = \frac{\sum_{t=1, O_{t=k}}^{T} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)};\ \acute{\pi}_i = \gamma_1(i) \tag{23}$$

the model parameters are re-estimated iteratively using $\acute{\lambda} = \left(\acute{A}, \acute{B}, \acute{\pi}\right)$, where $\acute{A} = \{\acute{a}_{ij}\}$, $B = \left\{ \acute{\ }_{j}(k) \right\} b$ and $\acute{\pi} = \{\acute{\pi}_i\}$. In this case, $\alpha_{t-1}(j) a_{ji}$ is the probability of the joint event that $O_1, O_2, \ldots, O_{T-1}$ are observed, given by $\alpha_{t-1}(j)$, and there is a transition from state q_j at time $t - 1$ to state q_i at time t, given by a_{ji}; $b_i(O_t)$ is the probability that O_t is observed from state q_i. Similarly, we can define the backward variable $\beta_t(i)$ as the probability of the observation sequence from time t + 1 to the end, given state q_i at time t and the model λ, $\beta_t(i) = P(O_{t+1}, O_{t+2}, \ldots, O_T; S_t = q_i, \lambda)$ [63,64].

It is important to note that dealing with neutrosophic measures introduces additional complexities, and the mathematical operations involved in the computations are adapted to handle neutrosophic sets. The pseudocode for the neutrosophic Baum–Welch algorithm might look similar to the standard Baum–Welch algorithm but with additional considerations for neutrosophic measures and computations. The implementation may require specialized libraries or tools capable of handling neutrosophic sets and operations. See [65] for more details.

B. Recognition phase

Recognizers primarily function to classify testing gestures into their respective training classes. In order for the recognizer to find the corresponding trained gesture class, it must first rank the classes according to the available testing gesture [17,66,67]. The NHMM-based gesture recognition paradigm's final parameters are derived using the neutrosophic Baum–Welch recursive approach. In order to maximize the probability of the state order for the given observation order, the neutrosophic Viterbi algorithm selects the highest state series [68,69]. The key steps for adapting the Viterbi algorithm to an NHMM are as follows:

1. Initialization: Initialize the Viterbi matrix to store the partial probabilities of the most likely sequence up to each state for each time step. Initialize the back pointer matrix to keep track of the path leading to the most likely sequence. Initialize the first column of the Viterbi matrix based on the neutrosophic emission probabilities and the initial probabilities of each neutrosophic state.

2. Recursion: For each subsequent time step, calculate the partial probabilities in the Viterbi matrix based on the neutrosophic transition probabilities and the neutrosophic emission probabilities. Use the recursive formula that incorporates the truth, indeterminacy, and false memberships in the calculation.
3. Backtracking: Update the back pointer matrix as the algorithm progresses, keeping track of the most likely path leading to each state.
4. Termination: Once the entire sequence has been processed, find the final state with the highest probability in the last column of the Viterbi matrix. This state represents the most likely ending state of the sequence. Use the back pointer matrix to backtrack from the final state to the initial state, reconstructing the most likely sequence of neutrosophic states. The formal definition of the Viterbi recursion is as follows:
   - Initialization:

$$v_1(j) = \pi_j b_j(O_1) \qquad 1 \le j \le N \tag{24}$$

$$b_t(j) = 0 \qquad 1 \le j \le N \tag{25}$$

   - Recursion:

$$v_t(j) = \max_{1 \le i \le N} v_{t-1}(i) a_{ij} b_j(O_t); 1 \le j \le N,\ 1 \le t \le T \tag{26}$$

$$b_t(j) = \underset{1 \le i \le N}{\text{argmax}} v_{t-1}(i) a_{ij} b_j(O_t); 1 \le j \le N,\ 1 \le t \le T \tag{27}$$

   - Termination:

$$\text{The best path}: \ p = \max_{1 \le i \le N} v_T(i) \tag{28}$$

$$\text{The start of the back trace}: \ q_T = \underset{1 \le i \le N}{\text{argmax}} v_T(i) \tag{29}$$

$v_{t-1}(i)$ is the previous Viterbi path probability from the previous time step, a_{ij} is the transition probability from the previous state q_i to the current state q_j, and $b_j(O_t)$ is the state observation likelihood of the observation symbol O_t given the current state j [68,69]. The pseudocode of the suggested model is summarized in Algorithm 1.

In conclusion, the system provides the option of combining hidden Markov chains with neutrosophic logic. This blending entails exchanging the basic HMM mathematical procedures with neutrosophic sets and numbers. Here are some potential advantages of using neutrosophic HMMs for gesture recognition: (1) Handling Uncertainty and Ambiguity: Neutrosophic logic allows for the representation of indeterminacy, uncertainty, and ambiguity in a more explicit manner. In gesture recognition, where the input data may be noisy or ambiguous, NHMMs can better handle uncertain information. (2) Flexibility in Modeling: Neutrosophic sets provide a flexible framework for modeling different degrees of truth, falsehood, and indeterminacy. This flexibility is particularly beneficial when dealing with complex and diverse gesture patterns. (3) Adaptability to Dynamic Environments: NHMMs can adapt to changes in gesture patterns over time. In dynamic environments, where gestures may evolve, or new gestures may emerge, the flexibility of neutrosophic logic can be advantageous.

While NHMMs offer several advantages for gesture recognition, they also come with certain disadvantages and challenges. It is essential to consider these aspects when deciding whether to use NHMMs for a particular application. Here are some potential disadvantages: (1) Computational Complexity: Neutrosophic logic involves the manipulation of truth values, indeterminacy, and falsity, which can increase the computational complexity of NHMMs. This may result in more extended training and inference times than traditional HMMs. (2) Learning Neutrosophic Parameters: Estimating the parameters of neutrosophic sets during the training process can be complex. The learning algorithm must effectively capture the various degrees of truth, falsehood, and indeterminacy, which requires careful tuning and validation. (3) Resource Intensity: Implementing NHMMs may demand more computational resources and memory than simpler models. This can be a concern in resource-constrained environments, such as embedded systems or real-time applications.

Algorithm 1: Neutrosophic HMM-based hand gesture recognition system

```
Step 1: Data Preprocessing
    # Extract features from hand gesture images
        Def extract_features (gesture images):
            # Implement feature extraction technique
                Features = []
                For img in gesture_images:
                    Feature = SVD_extract_feature_from_image (img)
                    Features. Append (feature)
        Return features
Step 2: Neutrosophic HMM Training
Initialize:
- Define the number of states ($N$) and observations ($M$).
- Initialize neutrosophic parameters (truth, indeterminacy, falsity) for
- Transition and emission probabilities.
- Initialize initial state probabilities using neutrosophic parameters.
Training (Baum–Welch Algorithm):
{
1. Initialize transition and emission probability matrices with random neutrosophic parameters.
2. Repeat until convergence {
        a. Forward pass: Compute the forward probabilities using neutrosophic arithmetic.
            # Neutrosophic arithmetic would be employed for operations involving neutrosophic
                parameters, such as addition, multiplication, and comparison
        b. Backward pass: Compute the backward probabilities using neutrosophic arithmetic.
        c. Update transition and emission probabilities using neutrosophic arithmetic.
}

Inference (Viterbi Algorithm):
{
1. Given an observation sequence $O$ ($O$ = Features), initialize the Viterbi matrix and back pointer
matrix.
2. For each observation in $O$ {
    a. Update the Viterbi matrix using neutrosophic arithmetic.
    b. Update back pointer matrix.
}
3. Terminate: Find the most likely sequence of states using backtracking.
```

4. Results and Discussions

The accuracy of the proposed NHMM-based sign language recognizer was verified using gesture image samples downloaded from the Web and the Kaggle website. These samples included an extensive vocabulary with over 6000 isolated gesture signs selected from several sign language dictionaries, with ten samples for each gesture sign. Variations in brightness, scaling, distortion, rotation, and viewpoint were applied to those samples. Five samples, chosen randomly for each sign, were employed as the training samples; the registered test group (Reg) received two of the remaining samples, and the unregistered test group (Unreg) received the other samples. A hand gesture recognition rate, the proportion of hand postures that were correctly identified relative to the total number of hand postures, was used as the objective evaluation metric [3]. The prototype recognition method was developed using Google Colab Python version 2.7 (https://saturncloud.io/blog/how-to-update-google-colabs-python-version/ accessed on 1 January 2024) in a modular form and tested on a DellTM InspironTM N5110 Laptop, Dell Computer Corporation, Texas, which included the following specifications: 64-bitWindows 7 Home Premium, 4.00 GB RAM, Intel (R) Core (TM) i5-2410 M CPU, 2.30 GHz.

The current version of our model can effectively recognize single words. While determining word boundaries in sentences can be challenging, modern NLP systems have developed sophisticated methods to handle these complexities effectively. Python offers various libraries and frameworks for hand gesture recognition, leveraging techniques from computer vision and machine learning. Some commonly used functionalities and libraries include:

- OpenCV is a popular Python library for computer vision tasks. It provides functionalities for image and video processing, including hand detection and tracking. OpenCV can be used for tasks such as contour detection, hand segmentation, and gesture recognition.
- Scikit-learn: Scikit-learn is a widely used machine learning library in Python. It provides a high-level interface for various machine learning algorithms, including dimensionality reduction techniques such as Principal Component Analysis (PCA) and Truncated SVD (a variant of SVD). The Truncated SVD class in scikit-learn allows for an SVD-based feature extraction that is suitable for large datasets.
- Gesture Recognition Toolkit (GRT): GRT is a C++ library with Python bindings that provides tools for real-time gesture recognition. It offers algorithms for feature extraction, classification, and gesture spotting, making it suitable for building gesture recognition systems.
- NumPy: This is a fundamental package for Python numerical computing. It provides support for multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.
- SciPy offers a wide range of numerical algorithms and mathematical functions that are not available in NumPy, including optimization, interpolation, integration, signal processing, and statistical functions. It provides high-level interfaces to optimize and integrate numerical algorithms efficiently, making it suitable for various scientific and engineering applications.
- While specific Python libraries or packages dedicated to NHMMs may not be readily available, we can leverage general-purpose libraries for probabilistic modeling and machine learning (such as NumPy, SciPy, and scikit-learn) to implement the NHMMs functionalities.

We tested our method on a large vocabulary with varying numbers of states and variable numbers of Gaussian mixtures per state, using the same observation vector of 52 SVD features per sign. Table 1 displays the test results of utilizing HMM based on conventional Gaussian mixtures model (GMM)-, fuzzy Gaussian mixtures model (FGMM)-, type-2 fuzzy Gaussian mixtures model (IT2GMM)-, and neutrosophic Gaussian mixtures model (NGMM)-based isolated SRL performances [70]. On the unregistered test set, the tests show that the suggested system improves recognition accuracy by 7%, 17%, and 24% compared to IT2FHMM, FHMM, and HMM, respectively, while dealing with large vocabulary sizes and sign data variations. Table 1 reveals that six states and ten Gaussian mixtures are optimal. In this experiment, we used a training set of five samples for each gesture class. These statistics are often altered in various datasets due to the direct impact of observational distribution.

One probable explanation is that neutrosophic sets' greater uncertainty ability and HMM's excellent sequential handling features, when combined in a different structure, may work against one other to provide better results. In other words, when faced with large vocabulary sizes, the suggested approach shows promising results in singer-independent hand gesture recognition. Neutrosophic logic explicitly accommodates unknown or undefined information through the concept of "indeterminacy". This can be useful when dealing with incomplete or missing data, providing a way to express the lack of clear information, as in the case of hand movement. Type-2 fuzzy logic, on the other hand, deals with uncertainty through the use of membership functions but may not provide a direct representation of unknown information. The three truth values (truth, indeterminacy, falsity) can be more naturally associated with real-world situations where information might be vague or incomplete. In contrast, the additional complexity introduced by type-2 fuzzy logic, with higher-order fuzzy sets, might make interpretation more challenging.

Table 1. Comparison results of different HMM models with different numbers of states and Gaussian mixtures for an unregistered test set.

Model	**Recognition Rate %**			
	GMM	**FGMM**	**IT2FGMM**	**NGMM**
6 states/10 mixtures	77.2	88.6	95.8	97.1
6 states/6 mixtures	74.5	77.2	88.3	95.4
6 states/4 mixtures	70.3	75.7	86.9	94.6
6 states/3 mixtures	67.8	73.5	85.3	93.1
4 states/2 mixtures	60.7	70.2	82.9	92.8
Overall Average	70.10	77.04	87.84	94.60

The relationship between the recognition ratio and the size of the SVD vector is also shown in Table 2. It is well-known that adding additional features causes the recognition rate to increase with time. Controlling the right feature size is obviously crucial. The relationship between the feature vector size and recognition rate is complex, and finding the right balance is often a matter of experimentation and fine-tuning. It involves considering the trade-offs between computational efficiency, model complexity, and the quality of the information represented by the features. In high-dimensional spaces, the "curse of dimensionality" can become a challenge. This phenomenon can lead to increased computational complexity, data sparsity, and the risk of overfitting. A more significant feature vector size only sometimes guarantees improved recognition rates. In fact, including irrelevant or redundant features may negatively impact model performance. Feature selection and dimensionality reduction techniques are often employed to identify and retain only the most informative features. Generally, a well-balanced feature vector size containing relevant information without unnecessary complexity is more likely to lead to better generalization [71].

Table 2. Recognition rate of different sizes of SVD feature for an unregistered test set for 6 states/10 mixtures.

Proposed Model	**No. of Features (SVD Coefficients)**		
	25 Features	**30 Features**	**50 Features**
Recognition Rate	92.7	94.6	97.1

In the third experiment, using signer-independent Arabic sign language, the suggested system is compared to the one in ref. [72]. Both systems use the same datasets. However, they employ distinct feature extraction techniques, settings, and classifiers. According to Table 3, the proposed system outperforms the vision-based deep learning approach (an average improvement of 3%). This is because NHMM is capable of handling modeling parameters that, because of issues with sign identification, insufficient or noisy data, or both, might fail to accurately represent the initial distribution of the observations. In general, NHMMs can handle missing data more gracefully than many deep learning architectures. They have the ability to model the uncertainty associated with missing observations through their probabilistic framework.

Furthermore, NHMMs inherently possess a stateful memory, which allows for them to capture the dependencies and temporal relationships between observations. This makes them practical for tasks where the history of observations is crucial. The parameters of an HMM (transition probabilities, emission probabilities) often have clear interpretations, making it easier for domain experts to provide insights and fine-tune the model. Finally, NHMMs can perform well with relatively small amounts of training data. This is especially useful in scenarios where obtaining large labeled datasets for deep learning may be challenging.

Table 3. Comparison with a similar free-hands signer-independent isolated SLR system.

	Instruments Used	Feature Vector Length	Average Recognition Rate
Proposed System (SVD coefficients + NHMM classifier)	None: Free Hands	30 SVD coefficients	94.6
Vision-based deep learning approach [72]		Determined according to the CNN parameters	91.2

The efficacy of the proposed model for addressing uncertainty associated with hand signs in Arabic SLR applications was further demonstrated in a subsequent set of experiments, which compared it to other solutions stated in ref. [73] that were employed as a black box with default parameters. As shown in Table 4, while specific classifiers, including Naïve Bayes and MLP neural networks, achieved a better accuracy than the suggested model, the results are still relatively close. Traditional classifiers, such as k-Nearest Neighbors (kNN), Support Vector Machines (SVM), and Naïve Bayes, typically operate under assumptions of deterministic classification, where each input instance is assigned to a single class label, without quantifying uncertainty. Here are some reasons why traditional classifiers may struggle to handle the uncertainty associated with hand signs:

- Traditional classifiers aim to find deterministic decision boundaries that separate different classes in the feature space. This approach may not adequately capture uncertainty when there are overlapping regions between classes or the decision boundary is ambiguous due to variations in hand signs.
- Traditional classifiers typically assign a single class label to each input instance based on its feature representation. This binary decision-making process does not provide information about the confidence or uncertainty associated with the assigned class label, making it challenging to assess the reliability of classification results.
- Traditional classifiers often make simplifying assumptions about data distribution and may not explicitly model uncertainty in their classification process. This can lead to suboptimal performance, especially when dealing with ambiguous or uncertain hand signs.
- Hand signs in sign language can be inherently ambiguous, with similar hand configurations or movements representing multiple meanings depending on context or subtle variations. Traditional classifiers may struggle to disambiguate between such signs and accurately assign class labels in uncertain scenarios.

To address these limitations and better handle the uncertainty associated with hand signs, our work explores neutrosophic HMM, which explicitly represents uncertainty using neutrosophic logic, allowing for the representation of indeterminacy, uncertainty, and contradiction in sign language data. This enables NHMMs to capture and quantify uncertainty associated with variations in hand positions, orientations, and movements, providing a principled framework for handling uncertainty in SLR. Finally, having a larger number of samples per person helps to reduce uncertainty in the recognition process. Uncertainty arises from variations in how individuals produce signs, as well as from environmental factors such as lighting conditions, background clutter, and occlusion. By providing more examples of each sign from different perspectives and contexts, the recognition system can better learn to distinguish between different variations and reduce uncertainty in the recognition process.

The impact of NHMM topologies on the proposed system's recognition rate is examined in the fourth experiment. There are three types of models: the fully connected Ergodic model, the left–right model (LR), and the left–right banded (LRB) model. In the former, every state may be reached from any other state, while in the latter two, each state can only move backward or forward [74]. The six states' average LRB topology ratio was 95.76%, as shown in Table 5. Compared to LR and ergodic topologies, the LRB topology performed better. In general, LRB topology often requires fewer parameters compared to fully connected topologies. The reduced number of parameters can lead to more efficient training, especially when dealing with limited data. Furthermore, the constrained structure of LRB topology can help mitigate overfitting, as the model is forced to learn a more constrained set of transitions. This is particularly beneficial when dealing with limited training data.

Table 4. Set of results of different solutions stated in Ref. [73] that use isolated static gesture signs.

Database	No. of Samples	Classifiers	Accuracy	Comments
28 ArSL	10 samples/letter	KNN	97.1%	Traditional kNN relies on instance-based classification and does not explicitly model uncertainty
		HMM	97.7%	They assume that observations are generated from a finite set of hidden states with known transition probabilities and emission probabilities, but they do not provide a formal mechanism for quantifying uncertainty
		Naïve Bayes	98.3%	They assume independence between features and rely on the strong conditional independence assumption, which may not accurately capture the complex dependencies present in hand gesture data.
		MLP	99.1%	They learn deterministic mappings from input features to output labels and do not provide a formal mechanism for quantifying uncertainty.
		SVM	90.7%	Traditional SVM relies on margin-based classification and does not explicitly model uncertainty.
		Proposed Model	97.1%	NHMMs explicitly model uncertainty using neutrosophic logic, allowing for the representation of indeterminacy, uncertainty, and contradiction in hand gesture data.

While LR strictly enforces transitions from left to right without skipping any states, LRB introduces a degree of flexibility by allowing transitions between neighboring states within a specified band or distance. The choice between LR and LRB depends on the characteristics of the underlying process being modeled. LR may be sufficient for scenarios with strict sequential dependencies, while LRB can be beneficial when more flexible state transitions are needed while still maintaining a general left-to-right structure [74].

Table 5. Isolated gesture recognition results with different NHMM topologies (six states/four mixtures).

Proposed Model	No. of Features (SVD Coefficients)		
	Ergodic	LR	LRB
Average Recognition Rate	73.56	89.34	95.76

Using a confusion matrix that contrasts the classifier's assigned class (column) with their correct initial class (row)—as shown in Table 6 of randomly selected samples—the fifth experiment can assess the accuracy of the NHMM recognizer with the experimentally obtained optimal parameters. This allows for the accurate recognition of items on the diagonal. Recognition errors occur when items are not on the diagonal. The results of the tests reveal that the proposed system is reliable, as it correctly recognizes sign language with few false alarms. It is clear that some signs have abysmal recognition rates. When it comes to the gesture "B", for instance, the recognition rate is 80%, and it is often categorized as "C". This is because both motions have striking similarities in the placement, motion, and orientation of the center hand. Hence, it is quite probable that the observation (feature) vectors generated from the hand-tracking segment will be exceptionally closely spaced. Therefore, the system will disorganize between the two signs, resulting in a high mistake rate for these particular gestures.

Table 6. Confusion matrix (10 samples per class) with experimentally obtained optimal parameters.

Input Samples	A	B	C	Five	Point	V	Accuracy (%)
A	10	0	0	0	0	0	100
B	0	8	2	0	0	0	80
C	0	2	8	0	0	0	80
Five	0	0	0	10	0	0	100
Point	0	0	0	0	10	0	100
V	0	0	0	0	0	10	100
Accuracy (%)	100	80	80	100	100	100	93.33%

Sign language involves a wide range of gestures, and there can be significant variability in how individuals express the same concept. This variability can make it challenging for NHMMs to accurately model and recognize different signing patterns. Furthermore, the recognition rates may be affected by environmental factors such as lighting conditions, background noise, or camera angle. NHMMs may be sensitive to such variations, and if the training conditions differ significantly from the testing conditions, recognition rates can drop. Finally, NHMMs may face challenges in adapting to individual signing styles. If there is a significant variation in signing styles among different signers, the model may struggle to adapt and generalize well across diverse signing patterns.

There are several scenarios where a hand gesture recognition system may provide incorrect classifications, leading to entries in the confusion matrix that do not correspond to the diagonal (where the predicted class equals the actual class). Here are a few common scenarios: (1) Gestures that are visually or semantically similar may be confused with each other, leading to misclassifications. For example, two gestures that involve similar hand movements may be too complicated for the system to distinguish accurately. (2) In real-world scenarios, input data may be noisy or ambiguous, making it challenging for the system to make accurate predictions; for example, poor lighting conditions, occlusions, or variations in hand orientation may lead to misclassifications. (3) The hand gesture recognition model may have limitations in capturing the complexity or variability of gestures in the dataset; for example, the model may not generalize well to unseen variations or may be biased towards certain gestures. Analyzing these errors can help identify areas for improvement, such as collecting more training data, refining the feature extraction process, or fine-tuning the model parameters.

4.1. Model's Computational Complexity

Determining the exact time complexity (in terms of big O notations) of a Neutrosophic Hidden Markov Model (NHMM)-based hand gesture recognition system is complex and would depend on various factors, such as the size of the input data, the complexity of the NHMM model, the number of states in the NHMM, the number of observations, the training algorithm used, and the inference algorithm that was employed. Below, we will provide an overview of the standard components of the suggested system and their computational complexities:

- The complexity of the preprocessing step in hand gesture recognition can be influenced by the specific combination of preprocessing techniques that is used and the size of the input images. In many cases, the preprocessing step has a linear or near-linear time complexity with respect to the number of pixels in the input image with complexity $O(N \times M)$, where $N \times M$ is the number of pixels in the image.

- The complexity of the feature extraction step based on SVD depends on the size of the input data and the desired number of features to be extracted. Consider an input matrix of size $T_S \times T_f$, where T_S is the number of samples (e.g., gesture images) and T_f is the number of features. The time complexity of computing the full SVD of a matrix of size $T_S \times T_f$ is generally considered to be $O(min(T_S {T_f}^2, {T_S}^2 T_f)$. After computing the SVD, the next step might involve selecting a subset of the computed singular vectors/values as features. Common methods for selecting features include choosing the top-k singular vectors/values.
- Training Complexity: Training an NHMM typically involves estimating parameters, which often involves the Baum–Welch algorithm. The time complexity of training an NHMM can be pretty high, and it is often polynomial in terms of the number of observations and the number of states with complexity $O(ST_O I)$, where S is the number of states, T_O is the number of observations, and I is the number of iterations required for convergence.
- Inference Complexity: Inference in NHMMs involves computing the likelihood of the observed data given the model, which typically requires the Viterbi algorithm. The time complexity of inference in NHMMs is often polynomial in terms of the number of states and the number of observations with complexity $O(S^2 T_O)$.
- Prediction Complexity: If the prediction of future observations is required, it generally involves the use of complexities as the inference.

Thus, the overall complexity of NHMM-based hand gesture recognition would be the dominant term among them $O(S^2 T_O)$. The overall complexity of the NHMM-based hand gesture recognition system would depend on the combination of these factors and any additional preprocessing or post-processing steps involved. It is important to note that while NHMMs offer certain advantages over traditional HMMs in terms of handling uncertainty and imprecision, they also introduce additional computational complexity.

4.2. Limitations

The suggested model handles static hand posters and cannot capture dynamic gestures that involve continuous movement or changes in hand position over time. As a result, dynamic gestures, such as waving or pointing, cannot be effectively represented in static images alone. In real-time communication or interaction scenarios, dynamic gesture recognition systems are more suitable. The quality of the image-capturing device and the resolution of the image can significantly affect the accuracy of hand gesture recognition. Low-quality images or low-resolution images may not provide enough detail for accurate recognition. Finally, NHMMs may require a significant amount of data to accurately estimate model parameters, especially when dealing with neutrosophic uncertainties. Insufficient data can lead to poor model performance and unreliable inference results. The use of inference and learning algorithms for NHMMs may be computationally intensive, particularly when dealing with large datasets or complex models. This can result in longer training times and increased computational resources compared to traditional HMMs.

5. Conclusions

To improve the accuracy of signer-independently separated SRL identification, this research introduces an adaptive technique based on neutrosophic HMM. Using the collected SVD feature space, NHMM generates a robust approximation of the distinct recognition gestures. After training the set of pictures using the neutrosophic-based forward–backward technique, the approach constructs a database based on the test image's similarity using the neutrosophic-based Viterbi algorithm. Then, it uses that database to identify the most likely gestures. NHMMs can offer certain advantages for sign language recognition, especially when dealing with the inherent uncertainties and complexities in sign language data.

Integrating neutrosophic logic with HMMs can provide a more robust framework for sign language recognition. Here are some potential advantages: (1) Sign language gestures often exhibit inherent uncertainty and vagueness. Neutrosophic logic allows for the representation of indeterminate and vague information, providing a more flexible and realistic approach to model uncertainty in sign language recognition. (2) Ambiguity is common in sign languages, where different signs may share similar or overlapping visual characteristics. Neutrosophic logic can handle ambiguity by representing the degree of truth, falsity, and indeterminacy associated with different interpretations of a sign. (3) Neutrosophic logic can handle contradictions, which can be helpful when dealing with conflicting or inconsistent information in sign language recognition. This can enhance the robustness of the model in scenarios where multiple interpretations or conflicting visual cues are present. (4) Neutrosophic logic extends traditional fuzzy logic by explicitly handling indeterminacy, thus providing a more nuanced representation of fuzzy concepts. This can be beneficial when dealing with subtle and fuzzy distinctions in sign language recognition.

While NHMMs offer particular advantages in dealing with uncertainty and vagueness, they also have disadvantages, such as the following: (1) Complexity: NHMMs can introduce additional complexity compared to traditional HMMs due to the incorporation of neutrosophic logic. This complexity may manifest in model formulation, inference algorithms, and parameter estimation. (2) Generalization: The generalization of NHMMs to new or unseen data may be challenging, particularly if the model is overfit to the training data or if the neutrosophic uncertainties are not representative of real-world uncertainties. (3) Model Selection: NHMMs introduce additional parameters related to neutrosophic uncertainties, which can complicate the model selection process. Choosing an appropriate model structure and complexity for a given application may require careful consideration and experimentation.

The results of the experiments demonstrate that the proposed model is better than the type-2 fuzzy HMM, with an improvement of around 7% in terms of accurately recognizing SLR. In the future, there will be more investigations aiming to develop sophisticated structures for neutrosophic-based HMMs to capture the dynamic and complex nature of sign language gestures. This may involve exploring different state transition models and emission probabilities, incorporating additional contextual information, and enhancing the computational efficiency of neutrosophic-based HMMs to enable real-time processing for applications such as interactive sign language recognition systems or assistive devices. This may involve optimization techniques, parallel processing, or hardware acceleration.

Author Contributions: Conceptualization, M.A.-S., Á.B. and S.M.S.; methodology, M.A.-S., Á.B. and O.A.H.; software, M.A.-S., Á.B. and O.A.H.; validation, M.A.-S., Á.B. and O.A.H.; formal analysis, M.A.-S., Á.B. and S.M.S.; investigation, Á.B. and O.A.H.; resources, O.A.H.; data curation, Á.B. and O.A.H.; writing—original draft preparation, M.A.-S., Á.B. and S.M.S.; writing—review and editing, Á.B. and O.A.H.; visualization, M.A.-S.; supervision, Á.B. and O.A.H.; project administration, M.A.-S.; funding acquisition, M.A.-S. and Á.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All data were contained in the main text.

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

References

1. Sarhan, N.; Frintrop, S. Unraveling a Decade: A Comprehensive Survey on Isolated Sign Language Recognition. In Proceedings of the IEEE International Conference on Computer Vision, Paris, France, 1–6 October 2023; pp. 3210–3219.
2. Wali, A.; Shariq, R.; Shoaib, S.; Amir, S.; Farhan, A. Recent progress in sign language recognition: A review. *Mach. Vis. Appl.* **2023**, *34*, 127–140. [CrossRef]
3. Núñez-Marcos, A.; Perez-de-Viñaspre, O.; Labaka, G. A survey on Sign Language machine translation. *Expert Syst. Appl.* **2023**, *213*, 18993. [CrossRef]

4. Minu, R. An Extensive Survey on Sign Language Recognition Methods. In Proceedings of the 7th International Conference on Computing Methodologies and Communication, Erode, India, 23–25 February 2023; pp. 613–619.
5. Robert, E.; Duraisamy, H. A review on computational methods based automated sign language recognition system for hearing and speech impaired community. *Concurr. Comput. Pract. Exp.* **2023**, *35*, e7653. [CrossRef]
6. Singh, S.; Chaturvedi, A. Applying Machine Learning for American Sign Language Recognition: A Brief Survey. In Proceedings of the International Conference on Communication and Intelligent Systems, Moscow, Russia, 14–16 December 2023; Springer Nature: Singapore, 2022; pp. 297–309.
7. Liang, Z.; Li, H.; Chai, J. Sign Language Translation: A Survey of Approaches and Techniques. *Electronics* **2023**, *12*, 2678. [CrossRef]
8. Rakesh, S.; Venu, M.; Jayaram, D.; Gupta, I.; Agarwal, K.; Nishanth, G. A Review on Sign Language Recognition Techniques. In Proceedings of the International Conference on Information and Management Engineering, Zhenjiang, China, 21–23 October 2022; Springer Nature: Singapore, 2022; pp. 301–309.
9. Ingle, T.; Daware, S.; Kumbhar, N.; Raut, K.; Waghmare, P.; Dhawase, D. Sign Language Recognition. *Scand. J. Inf. Syst.* **2023**, *35*, 294–298.
10. Kamble, N.; More, N.; Wargantiwar, O.; More, S. Deep Learning-Based Sign Language Recognition and Translation. In Proceedings of the International Conference on Soft Computing for Security Applications, Dhirajlal, Gandhi, India, 21–22 April 2023; Springer Nature: Singapore, 2023; pp. 49–63.
11. Qahtan, S.; Alsattar, H.; Zaidan, A.; Deveci, M.; Pamucar, D.; Martinez, L. A comparative study of evaluating and benchmarking sign language recognition system-based wearable sensory devices using a single fuzzy set. *Knowl. Based Syst.* **2023**, *269*, 110519. [CrossRef]
12. Kanavos, A.; Papadimitriou, O.; Mylonas, P.; Maragoudakis, M. Enhancing sign language recognition using deep convolutional neural networks. In Proceedings of the 14th International Conference on Information Intelligence, Systems and Applications, Volos, Greece, 10–12 July 2023; pp. 1–4.
13. Kumar, R.; Goyal, V.; Goyal, L. State of the Art of Automation in Sign Language: A Systematic Review. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **2023**, *22*, 1–80. [CrossRef]
14. Kuppuswami, G.; Sujatha, R.; Nagarajan, D.; Kavikumar, J. Markov chain based on neutrosophic numbers in decision making. *Kuwait J. Sci.* **2021**, *48*, 1–16.
15. Patil, A.; Kulkarni, A.; Yesane, H.; Sadani, M.; Satav, P. Literature survey: Sign language recognition using gesture recognition and natural language processing. Data Management, Analytics and Innovation. In Proceedings of the International Conference on Data Management, Analytics and Innovation, Pune, India, 19–21 January 2024; Springer: Berlin/Heidelberg, Germany, 2021; Volume 1, pp. 197–210.
16. Sultan, A.; Makram, W.; Kayed, M.; Ali, A. Sign language identification and recognition: A comparative study. *Open Comput. Sci.* **2022**, *12*, 191–210. [CrossRef]
17. Fadel, N.; Kareem, E. Computer Vision Techniques for Hand Gesture Recognition: Survey. In Proceedings of the International Conference on New Trends in Information and Communications Technology Applications, Baghdad, Iraq, 20–21 December 2023; Springer Nature: Cham, Switzerland, 2022; pp. 50–76.
18. Al-Farid, F.; Hashim, N.; Abdullah, J.; Bhuiyan, M.; Shahida Mohd, W.; Uddin, J.; Haque, M.; Husen, M. A structured and methodological review on vision-based hand gesture recognition system. *J. Imaging* **2022**, *8*, 153. [CrossRef]
19. Bhiri, N.; Ameur, S.; Alouani, I.; Mahjoub, M.; Khalifa, A. Hand gesture recognition with focus on leap motion: An overview, real world challenges and future directions. *Expert Syst. Appl.* **2023**, *225*, 120125. [CrossRef]
20. Wu, S.; Li, Z.; Li, S.; Liu, Q.; Wu, W. An overview of gesture recognition. In Proceedings of the International Conference on Computer Application and Information Security, Dubai, United Arab Emirates, 24–25 January 2023; Volume 12609, pp. 600–606.
21. Franslin, N.; Ng, G. Vision-based dynamic hand gesture recognition techniques and applications: A review. In Proceedings of the 8th International Conference on Computational Science and Technology, Labuan, Malaysia, 28–29 August 2021; Springer: Singapore, 2022; pp. 125–138.
22. Parihar, S.; Shrotriya, N.; Thakore, P. Hand Gesture Recognition: A Review. In Proceedings of the International Conference on Mathematical Modeling and Computational Science, Bangkok, Thailand, 10–11 January 2025; Springer Nature: Singapore, 2023; pp. 471–483.
23. Parcheta, Z.; Martínez-Hinarejos, C. Sign language gesture recognition using HMM. In Proceedings of the 8th Iberian Conference in Pattern Recognition and Image Analysis, Faro, Portugal, 20–23 June 2017; Springer International Publishing: Berlin/Heidelberg, Germany, 2017; pp. 419–426.
24. Buttar, A.; Ahmad, U.; Gumaei, A.; Assiri, A.; Akbar, M.; Alkhamees, B. Deep Learning in Sign Language Recognition: A Hybrid Approach for the Recognition of Static and Dynamic Signs. *Mathematics* **2023**, *11*, 3729. [CrossRef]
25. Tu, G.; Li, Q.; Jiang, D. Dynamic Gesture Recognition Based on HMM-DTW Model Using Leap Motion. In Proceedings of the International Symposium of Artificial Intelligence Algorithms and Applications, Vienna, Austria, 16–17 March 2024; Springer: Singapore, 2020; pp. 788–798.
26. Sagayam, K.; Hemanth, D.; Vasanth, X.; Henesy, L.; Ho, C. Optimization of a HMM-based hand gesture recognition system using a hybrid cuckoo search algorithm. In *Hybrid Metaheuristics Image Analysis*; Springer International Publishing: Berlin/Heidelberg, Germany, 2018; pp. 87–114.

27. Sawicki, A.; Daunoravičienė, K.; Griškevičius, J. Recognition of human-computer interaction gestures acquired by internal motion sensors with the use of hidden Markov models. *Adv. Comput. Sci. Research.* **2021**, *15*, 1–14.
28. Elmezain, M.; Alwateer, M.; El-Agamy, R.; Atlam, E.; Ibrahim, H. Forward hand gesture spotting and prediction using HMM-DNN model. *Informatics* **2022**, *10*, 1. [CrossRef]
29. Miah, A.; Hasan, M.; Shin, J.; Okuyama, Y.; Tomioka, Y. Multistage spatial attention-based neural network for hand gesture recognition. *Computers* **2023**, *12*, 13. [CrossRef]
30. Mohammed, A.; Lv, J.; Islam, M.; Sang, Y. Multi-model ensemble gesture recognition network for high-accuracy dynamic hand gesture recognition. *J. Ambient Intell. Humaniz. Comput.* **2023**, *14*, 6829–6842. [CrossRef]
31. Dubey, A. Enhanced hand-gesture recognition by improved beetle swarm optimized probabilistic neural network for human–computer interaction. *J. Ambient Intell. Humaniz. Comput.* **2023**, *14*, 12035–12048. [CrossRef]
32. Miah, A.; Hasan, M.; Shin, J. Dynamic Hand Gesture Recognition using Multi-Branch Attention Based Graph and General Deep Learning Model. *IEEE Access* **2023**, *11*, 4703–4716. [CrossRef]
33. Alabdullah, B.; Ansar, H.; Mudawi, N.; 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. [CrossRef]
34. Damaneh, M.; Mohanna, F.; Jafari, P. Static hand gesture recognition in sign language based on convolutional neural network with feature extraction method using ORB descriptor and Gabor filter. *Expert Syst. Appl.* **2023**, *211*, 118559. [CrossRef]
35. Bhaumik, G.; Verma, M.; Govil, M.; Vipparthi, S. Hyfinet: Hybrid feature attention network for hand gesture recognition. *Multimed. Tools Appl.* **2023**, *82*, 4863–4882. [CrossRef]
36. Ibrahim, I. Hand Gesture Recognition System Utilizing Hidden Markov Model for Computer Visions Applications. *Int. J. Adv. Acad. Res.* **2023**, *9*, 36–44.
37. John, J.; Deshpande, S. A Comparative Study on Challenges and Solutions on Hand Gesture Recognition. In *Computational Intelligence for Engineering and Management Applications*; Springer Nature: Singapore, 2023; pp. 229–240.
38. Saboo, S.; Singha, J. Dynamic hand gesture tracking and recognition: Survey of different phases. *Int. J. Syst. Innov.* **2023**, *7*, 47–70.
39. John, J.; Deshpande, S. Hand Gesture Identification Using Deep Learning and Artificial Neural Networks: A Review. In *Computational Intelligence for Engineering and Management Application*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 389–400.
40. Yalçın, S.; Kaya, I. Analyzing of process capability indices based on neutrosophic sets. *Comput. Appl. Math.* **2022**, *41*, 287. [CrossRef]
41. Nanni, L.; Loreggia, A.; Lumini, A.; Dorizza, A. A Standardized Approach for Skin Detection: Analysis of the Literature and Case Studies. *J. Imaging* **2023**, *9*, 35. [CrossRef] [PubMed]
42. ArulMurugan, S.; Somaiswariy, S. Virtual mouse using hand gestures by skin recognition. *J. Popul. Ther. Clin. Pharmacol.* **2023**, *30*, 251–258.
43. Abujayyab, S.; Almajalid, R.; Wazirali, R.; Ahmad, R.; Taşoğlu, E.; Karas, I.; Hijazi, I. Integrating object-based and pixel-based segmentation for building footprint extraction from satellite images. *J. King Saud Univ. Comput. Inf. Sci.* **2023**, *35*, 101802. [CrossRef]
44. Sengupta, S.; Mittal, N.; Modi, M. Morphological Transformation in Color Space-Based Edge Detection of Skin Lesion Images. In Proceedings of the International Conference in Innovations in Cyber Physical Systems, Delhi, India, 22–23 October 2021; Springer: Singapore, 2021; pp. 265–273.
45. Khanam, R.; Johri, P.; Diván, M. Human Skin Color Detection Technique Using Different Color Models. In *Trends and Advancements of Image Processing and Its Applications*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 261–279.
46. Oudah, M.; Al-Naji, A.; Chahl, J. Hand gesture recognition based on computer vision: A review of techniques. *J. Imaging* **2020**, *6*, 73. [CrossRef] [PubMed]
47. Chen, G.; Dong, Z.; Wang, J.; Xia, L. Parallel temporal feature selection based on improved attention mechanism for dynamic gesture recognition. *Complex Intell. Syst.* **2023**, *9*, 1377–1390. [CrossRef]
48. Kowdiki, M.; Khaparde, A. Automatic hand gesture recognition using hybrid meta-heuristic-based feature selection and classification with dynamic time warping. *Comput. Sci. Rev.* **2021**, *39*, 100320. [CrossRef]
49. Nogales, R.; Benalcázar, M. Hand Gesture Recognition Using Automatic Feature Extraction and Deep Learning Algorithms with Memory. *Big Data Cogn. Comput.* **2023**, *7*, 102. [CrossRef]
50. Yadukrishnan, V.; Anilkumar, A.; Arun, K.; Madhu, M.; Hareesh, V. Robust Feature Extraction Technique for Hand Gesture Recognition System. In Proceedings of the International Conference on Intelligent Computing & Optimization, Hua Hin, Thailand, 27–28 April 2023; Springer Nature: Cham, Switzerland, 2023; pp. 250–259.
51. Jiang, J.; Yang, W.; Ren, H. Seismic wavefield information extraction method based on adaptive local singular value decomposition. *J. Appl. Geophys.* **2023**, *210*, 104965. [CrossRef]
52. Zhu, H.; Wu, C.; Zhou, Y.; Xie, Y.; Zhou, T. Electric shock feature extraction method based on adaptive variational mode decomposition and singular value decomposition. *IET Sci. Meas. Technol.* **2023**, *17*, 361–372. [CrossRef]
53. Shahsavani, F.; Nasiripour, R.; Shakeri, R.; Gholamrezaee, A. Arrhythmia detection based on the reduced features with K-SVD sparse coding algorithm. *Multimed. Tools Appl.* **2023**, *82*, 12337–12350. [CrossRef]
54. Liu, B.; Pejó, B.; Tang, Q. Privacy-Preserving Federated Singular Value Decomposition. *Appl. Sci.* **2023**, *13*, 7373. [CrossRef]
55. Mifsud, M.; Camilleri, T.; Camilleri, K. HMM-based gesture recognition for eye-swipe typing. *Biomed. Signal Process. Control* **2023**, *86*, 105161. [CrossRef]

56. Manouchehri, N.; Bouguila, N. Human Activity Recognition with an HMM-Based Generative Model. *Sensors* **2023**, *23*, 1390. [CrossRef]
57. Hassan, M.; Ali, S.; Kim, J.; Saadia, A.; Sanaullah, M.; Alquhayz, H.; Safdar, K. Developing a Novel Methodology by Integrating Deep Learning and HMM for Segmentation of Retinal Blood Vessels in Fundus Images. *Interdiscip. Sci. Comput. Life Sci.* **2023**, *15*, 273–292. [CrossRef]
58. Nagarajan, D.; Kavikumar, J. Single-Valued and Interval-Valued Neutrosophic Hidden Markov Model. *Math. Probl. Eng.* **2022**, *2022*, 5323530. [CrossRef]
59. Nagarajan, D.; Kavikumar, J.; Tom, M.; Mahmud, M.; Broumi, S. Modelling the progression of Alzheimer's disease using Neutrosophic hidden Markov models. *Neutrosophic Sets Syst.* **2023**, *56*, 31–40.
60. Li, J.; Pedrycz, W.; Wang, X.; Liu, P. A Hidden Markov Model-based fuzzy modeling of multivariate time series. *Soft Comput.* **2023**, *27*, 837–854. [CrossRef]
61. Mahdi, A.; Nazaruddin, Y.; Mandasari, M. Driver Behavior Prediction Based on Environmental Observation Using Fuzzy Hidden Markov Model. *Int. J. Sustain. Transp.* **2023**, *6*, 22–27. [CrossRef]
62. Ren, X.; He, D.; Gao, X.; Zhou, Z.; Ho, C. An Improved Hidden Markov Model for Indoor Positioning. In Proceedings of the International Conference on Communications and Networking, Guilin, China, 21–24 August 2022; Springer Nature: Cham, Switzerland, 2022; pp. 403–420.
63. Nwanga, M.; Okafor, K.; Achumba, I.; Chukwudebe, G. Predictive Forensic Based—Characterization of Hidden Elements in Criminal Networks Using Baum-Welch Optimization Technique. In Proceedings of the International Conference in Illumination of Artificial Intelligence in Cybersecurity and Forensics, New York, NY, USA, 30 November–1 December 2023; Springer International Publishing: Cham, Switzerland, 2022; pp. 231–254.
64. Zhang, S.; Yang, L.; Zhang, Y.; Lu, Z.; Yu, J.; Cui, Z. Tensor-Based Baum–Welch Algorithms in Coupled Hidden Markov Model for Responsible Activity Prediction. *IEEE Trans. Comput. Soc. Syst.* **2023**, *10*, 2924–2937. [CrossRef]
65. Sleem, A.; Abdel-Baset, M.; El-henawy, I. PyIVNS: A python based tool for Interval-valued neutrosophic operations and normalization. *SoftwareX* **2020**, *12*, 100632. [CrossRef]
66. Qi, J.; Ma, L.; Cui, Z.; Yu, Y. Computer vision-based hand gesture recognition for human-robot interaction: A review. *Complex Intell. Syst.* **2023**, *9*, 1581–1606. [CrossRef]
67. Gupta, R.; Singh, A. Hand Gesture Recognition using OpenCV. In Proceedings of the 2023 10th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 15 March 2023; pp. 145–148.
68. Padgal, G.; Oza, S. An efficient Viterbi algorithm for communication system. *Res. J. Eng. Technol.* **2022**, *13*, 10–16. [CrossRef]
69. Huang, Q.; Wei, S.; Zhang, L. Radar Interferometric Phase Ambiguity Resolution Using Viterbi Algorithm for High-Precision Space Target Positioning. *IEEE Signal Process. Lett.* **2023**, *30*, 1242–1246. [CrossRef]
70. Huang, K.; Xu, M.; Qi, X. NGMMs: Neutrosophic Gaussian mixture models for breast ultrasound image classification. In Proceedings of the 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society, Guadalajara, Mexico, 1–5 November 2021; pp. 3943–3947.
71. Liu, X.; Wang, S.; Lu, S.; Yin, Z.; Li, X.; Yin, L.; Tian, J.; Zheng, W. Adapting feature selection algorithms for the classification of Chinese texts. *Systems* **2023**, *11*, 483. [CrossRef]
72. Balaha, M.; El-Kady, S.; Balaha, H.; Salama, M.; Emad, E.; Hassan, M.; Saafan, M.A. vision-based deep learning approach for independent-users Arabic sign language interpretation. *Multimed. Tools Appl.* **2023**, *82*, 6807–6826. [CrossRef]
73. Galván-Ruiz, J.; Travieso-González, C.M.; Pinan-Roescher, A.; Alonso Hernández, J.B. Robust Identification System for Spanish Sign Language Based on Three-Dimensional Frame Information. *Sensors* **2023**, *23*, 481. [CrossRef]
74. Kashlak, A.; Loliencar, P.; Heo, G. Topological Hidden Markov Models. *J. Mach. Learn. Res.* **2023**, *24*, 1–49.

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.