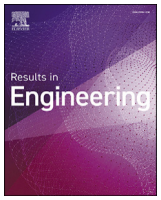
Results in Engineering 20 (2023) 101639

Contents lists available at ScienceDirect

Results in Engineering

journal homepage: www.sciencedirect.com/journal/results-in-engineering



Research paper

Empowering human-robot interaction using sEMG sensor: Hybrid deep learning model for accurate hand gesture recognition

Muhammad Hamza Zafar [a], Even Falkenberg Langås [a], Filippo Sanfilippo [a,b,*]

[a] Department of Engineering Sciences, University of Agder, Grimstad, Norway
[b] Department of Software Engineering, Kaunas University of Technology, Kaunas, Lithuania

ARTICLE INFO

Keywords:
Surface EMG
Human-robot interaction
Deep learning
Discrete wavelet transform

ABSTRACT

In this paper, a novel approach using a Henry Gas Solubility-based Stacked Convolutional Neural Network (HGS-SCNN) for hand gesture recognition using surface electromyography (sEMG) sensors is proposed. The stacked architecture of the CNN model helps to capture both low-level and high-level features, enabling effective representation learning. To begin, we generated a dataset comprising 600 samples of hand gestures. Next, we applied the Discrete Wavelet Transform (DWT) technique to extract features from the filtered sEMG signal. This step allowed us to capture both spatial and frequency information, enhancing the discriminative power of the extracted features. Extensive experiments are conducted to evaluate the performance of the proposed HGS-SCNN model. In addition, the obtained results are compared with state-of-the-art techniques, namely AOA-SCNN, GWO-SCNN, and WOA-SCNN. The comparative analysis demonstrates that the HGS-SCNN outperforms these existing methods, achieving an impressive accuracy of 99.3%. The experimental results validate the effectiveness of our proposed approach in accurately detecting hand gestures. The combination of DWT-based feature extraction and the HGS-SCNN model offers robust and reliable hand gesture recognition, thereby opening new possibilities for intuitive human-machine interaction and applications requiring gesture-based control.

1. Introduction

Human-robot interaction (HRI) holds significant importance in advancing intelligent robotic systems, playing a crucial role in enabling seamless collaboration and enhancing user experience across diverse domains like healthcare, manufacturing, and assistive technologies [1]. An essential component of effective HRI is accurate hand gesture recognition, allowing users to interact with robots naturally and intuitively by conveying commands, instructions, or intentions through hand movements [2]. Traditionally, vision-based techniques, involving cameras or depth sensors, have been the go-to approach for hand gesture recognition, although they face challenges related to lighting conditions, occlusions, variations in hand shapes, and applicability constraints in certain environments [3]. Also research is carried out on Leap motion controller based hand gesture detection in recent years [4].

To overcome these limitations and enhance the robustness and adaptability of hand gesture recognition in HRI, the potential of surface electromyography (sEMG) sensors is being explored. sEMG sensors are capable of detecting and recording electrical signals generated by muscle activation during hand movements [5]. These signals provide valuable insights into the underlying muscle activities, enabling the interpretation of intended hand gestures. Leveraging sEMG sensors for efficient hand gesture classification has led to the proposal of various machine learning (ML) and deep learning-based techniques, which will be comprehensively discussed in this literature review.

Machine learning has emerged as a promising solution in various fields to address and solve diverse challenges [6], [7], [8], [9]. The categorization of sEMG signals using ML approaches necessitates feature extraction, i.e., time-domain [10] or frequency-domain features [11], and time-frequency domain characteristics [10]. In [12], the classification of eight hand motions using the root mean square (RMS) as a feature in a linear Support Vector Machine (SVM) was performed, which made it possible to operate a robotic arm of 4 DoF. Altimemy et al. [13] used Linear Discriminant Analysis and SVM to classify 12 hand motions for amputees and 15 hand movements for those with intact limbs. In [14], Waris et al. classified gesture data obtained over the course of seven days using both surface-extracted EMG signals and intramuscular EMG signals, demonstrating that the performance of the Artificial Neu-

* Corresponding author.
E-mail address: filippo.sanfilippo@uia.no (F. Sanfilippo).

https://doi.org/10.1016/j.rineng.2023.101639
Received 29 June 2023; Received in revised form 12 November 2023; Accepted 26 November 2023
Available online 28 November 2023

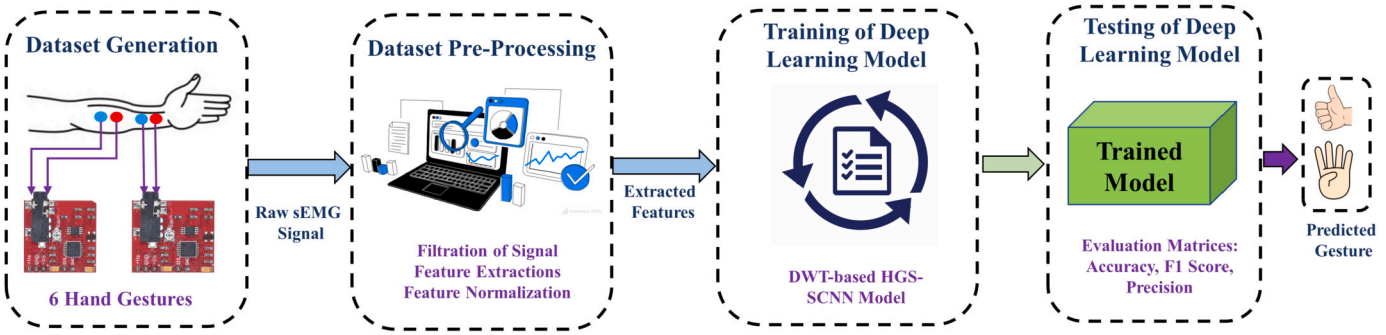


Fig. 1. Proposed hybrid DL-based hand gesture detection using sEMG.

ral Network (ANN) classifier has improved over time in comparison to the traditional k-nearest neighbors (KNN) and SVM classifiers.

Fourier inherent band functions (FIBFs) were created by dividing the sEMG signals, and statistical features were then extracted for SVM and KNN classifier [15]. By employing a jump motion device to capture the depth information, which increases the relabeling of gestures in the training phase, hand gesture detection may be improved. In [16], hand movements based on sEMG signals were classified using energy-based characteristics and a fine KNN. The requirement for manual feature set creation, which is a laborious operation and may not be sufficiently precise, is a major drawback of ML systems. The difficulty in choosing the best classifier for the specified characteristics is another problem.

Although ML algorithms have shown some promise for the categorization of sEMG signals, deep learning (DL) approaches have gained popularity in recent publications. It is because they tend to perform better and instantly pick up on the key aspects [17]. As a result, the exoskeletons' control system may be greatly enhanced by using DL approaches for the sEMG-based categorization.

In [18], Atzori et al. used a deep CNN architecture with two convolutional layers to carry out the sEMG classification job over the NinaPro DB1, DB2, and DB3 datasets. Compared to the KNN, SVM, Random Forests, and latent Dirichlet allocation (LDA) [19] ML classifiers currently in use, the authors have demonstrated a performance gain of 2-5%. Geng et al. [20] established that instantaneous visuals include patterns that are distinguishable between trials and comparable among samples of a single trial. In order to do this, they treated each sample of dimension 1x10 as an instantaneous picture and sent it into the CNN model as an input. Metaheuristic algorithms with deep learning models have gained a lot of attention in recent years [21–26].

A neural network variant that can handle sequential and temporal input is the recurrent neural network (RNN). Koch et al. employed a ConvLSTM cascaded using the LSTM architecture in [27] to classify hand gesture sequences. To identify the high density (HD) and sparse sEMG signals, a stacked RNN with two stage networks was implemented in [28]. An attention-based CNN-RNN architecture that is capable of classifying the sEMG pictures was created by Hu et al. [29]. Using waveform-based classification, an LSTM model and a deep backpropagation (BP) LSTM were contrasted in [30].

1.1. Contributions

Traditional ML and deep learning approaches for surface electromyography (sEMG) signal classification in the context of hand gesture recognition, often require manual feature extraction, which is a laborious and time-consuming process. Additionally, selecting the most appropriate classifier for the specific characteristics of sEMG signals can be challenging. Furthermore, traditional ML and DL classifiers may struggle to accurately classify sEMG signals due to their inability to capture intricate patterns, inefficient tuning of hyperparameters, and exploit the hierarchical representations within the data. These limitations hinder the accuracy, automation, and performance of sEMG signal classification for hand gesture recognition tasks.

This work seeks to address the following key challenges:

- Reliance on manual feature extraction, which is laborious, time-consuming, and sub-optimal. The use of automated DWT-based feature extraction overcomes this.
- Inability to handle intricate spatial and temporal patterns in sEMG signals, due to the use of traditional ML classifiers like SVM, KNN, ANN.
- Lack of robustness to real-world variations in hand shapes, sizes, gesture dynamics etc. The large heterogeneous dataset and deep learning approach aim to improve generalization.
- Difficulty in tuning hyperparameters and finding optimal network architectures.
- Many related works only focus on limited vocabulary or hand-crafted gestures lacking natural variability. This work uses a diversity of unrestrained hand gestures.
- Reliance on visual or depth cameras, which are sensitive to environmental conditions.
- Limited accuracy and reliability compared to vision-based techniques.

To overcome the above-mentioned problems, this work proposed a discrete wavelet transform (DWT) for automatic feature extractions using sEMG onset detection through moving average. After that hybrid DL model is proposed for the efficient classification of hand gestures. The proposed flow of the work is shown in Fig. 1. The contributions of this work are as follows:

- Dataset Generation: Creation of a comprehensive dataset of 600 samples, providing a valuable resource for hand gesture detection research.
- Feature Extraction with DWT: Effective utilization of the DWT for extracting discriminative features from hand gesture data, improving the accuracy of the detection system.
- HGS-SCNN Model: Introduction of the novel Henry Gas Solubility-based Stacked CNN (HGS-SCNN) model, demonstrating its superior performance in hand gesture detection compared to alternative techniques.
- Comparative Analysis: Comprehensive evaluation and comparison with AOA-SCNN, GWO-SCNN, and WOA-SCNN, showcasing the enhanced effectiveness of the HGS-SCNN approach in hand gesture detection.
- Accuracy: Proposed DWT-based HGS-SCNN model achieves 99.3% accuracy in hand gesture detection using only 2 channels of sEMG sensors.

The organization of the paper is structured into various sections. Section 2 presents a novel approach that utilizes the DWT for automated feature extraction and a Henry gas solubility algorithm based

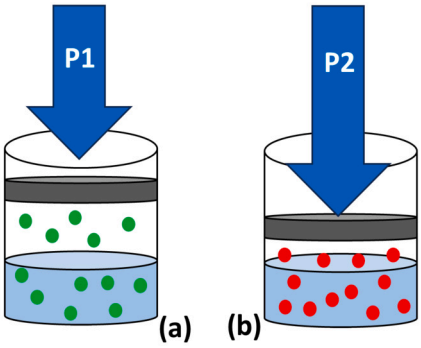


Fig. 2. Working of Henry gass optimization (a) Movement of particles when P1 pressure is applied (b) Movement of particles when P2 pressure is applied.

stacked CNN model for classification. Section 3 explains the acquisition and preparation of the sEMG dataset used for evaluation, including any preprocessing steps. Section 4 presents the experimental results of applying the proposed technique to the dataset, discussing the performance metrics and comparing them to previous approaches. Finally, Section 5 summarizes the findings, highlights the contributions of the proposed technique, and discusses potential avenues for future research in the field of sEMG signal classification for hand gesture recognition.

2. Proposed technique

2.1. Henry Gas Solubility algorithm (HGS)

J.W. Henry initially presented Henry's law in 1800. In general, the most solute that can dissolve in a given amount of solvent at a given pressure or temperature is referred to as the solubility [31]. Consequently, HGS was motivated by Henry's law's conduct. Henry's law may be used to calculate the solubility of low-solubility gases in liquids. Additionally, the two parameters that impact solubility are temperature and pressure. At high temperatures, solids become more soluble, whilst gases become less soluble. With respect to pressure, gas solubility rises as pressure does. As seen in Fig. 2, the subject of this algorithm is the solubility of gases.

In this section, the mathematical model of Henry Gas Solubility algorithm (HGS) is presented [32].

2.1.1. Initialization

The particles are randomly initialized based on the following equation:

$$X_i(t+1) = X_{\min} + r \times (X_{\max} - X_{\min}) \tag{1}$$

where $X(i)$ represents the location of the ith particles among a population N, r is a number chosen at random between 0 and 1, X_{min} and X_{max} are the problem's upper and lower limits, and t is the number of iterations. The Eq. (2)-(4) serves as the start value for no. of particles i, values of Henry's constant of type j ($H_j(t)$), partial pressure of gas i in cluster j (P_i, j), and E/R constant of type j (C_i).

$$h_j = p_1 \times r \tag{2}$$

$$P_{i,j} = p_2 \times r \tag{3}$$

$$C_j = p_3 \times r \tag{4}$$

where p_1, p_2, and p_3 are defined as constants with values equal to 5×10^{-2}, 100, and 1×10^{-2}, respectively and r is the random number between 0 and 1.

2.1.2. Clustering

The particles are separated into equal clusters, each of which is associated with a different kind of gas. Because each cluster is made up of gases that are identical to one another, they all have the same value for Henry's constant (H_j).

2.1.3. Evaluation

The particle that achieves the greatest equilibrium state among other molecules of the same kind is determined for each cluster j. The best particle throughout the whole swarm is then determined by ranking the particles in order of performance.

2.1.4. Henry coefficient updation

The solubility is updated based on the following equation:

$$S_{i,j}(t) = U \times L_j(t+1) \times P_{i,j}(t) \tag{5}$$

where, $S_{i,j}$ represents the solubility of gas i in cluster j, $P_{i,j}$ is the partial pressure of gas i in cluster j, and U is a constant.

2.1.5. Position updation

The position is updated using the following equation:

$$X_{i,j}(t+1) = X_{i,j}(t) + F \times r \times \gamma \times (X_{best,j}(t) - X_{i,j}(t)) F \times r \times \alpha \times (S_{i,j}(t) \times X_{best}(t)) \tag{6}$$

In this equation, $X_{i,j}$ denotes the location of the particle i in the cluster j, F controls the search agent's orientation and adds diversity (pm), and r and t denote the iteration time and random constant, respectively. The best particle in the swarm is X_{best}, whereas the best particle in the cluster is $X_{best,j}$. Additionally, $alpha$ is the impact of other particles on particle i in cluster j (equal to 1), $beta$ is a constant, and $gamma$ reflects the capacity of gas j in cluster i to interact with other gases in its cluster. In contrast to S_{best}, which represents the fitness of the best gas in the overall system, $S_{i,j}$ specifies the fitness of gas i in cluster j. The parameters $X_{best,j}$ and X_{best}, which denote the best particle in the cluster j and the best particle in the swarm, respectively, play a critical role in balancing the exploration and exploitation abilities.

2.1.6. Local optimum avoidance

The number of worst agents (denoted as N_w) is determined using the following equation:

$$N_w = N \times (rand(c2 - c1) + c1) \tag{7}$$

Here, N represents the total number of search agents, and $c1$ and $c2$ are constants with values of 0.1 and 0.2, respectively.

2.2. Discrete Wavelet Transform (DWT)

DWT is a mathematical tool that decomposes a signal into a set of wavelet coefficients at different scales [33]. A collection of wavelet functions, which are the dilations and translations of a mother wavelet function, serve as the foundation for the DWT. A signal x(n)'s DWT is given by:

$$c_{j,k} = \langle x, \psi_{j,k} \rangle = \sum_n x(n) \psi_{j,k}(n) \tag{8}$$

where j and k are integer values that define the scale and translation of the wavelet functions, $\psi_{j,k}(n)$ and $\phi_{j,k}(n)$ are the wavelet and scaling functions at scale j and translation k, and $\langle \cdot, \cdot \rangle$ denotes the inner product between two functions. The mother wavelet function $psi(n)$ and scaling function $phi(n)$ are dilated and translated to produce the wavelet and scaling functions as follows:

$$\psi_{j,k}(n) = 2^{j/2} \psi \left(2^j n - k\right) \tag{9}$$

The wavelet coefficients $c_{j,k}$ capture the high-frequency components of the signal at scale j and translation k, while the scaling coefficients $d_{j,k}$ capture the low-frequency components of the signal at scale j and translation k. The DWT can be computed iteratively by applying a series of high-pass and low-pass filters to the signal, followed by down

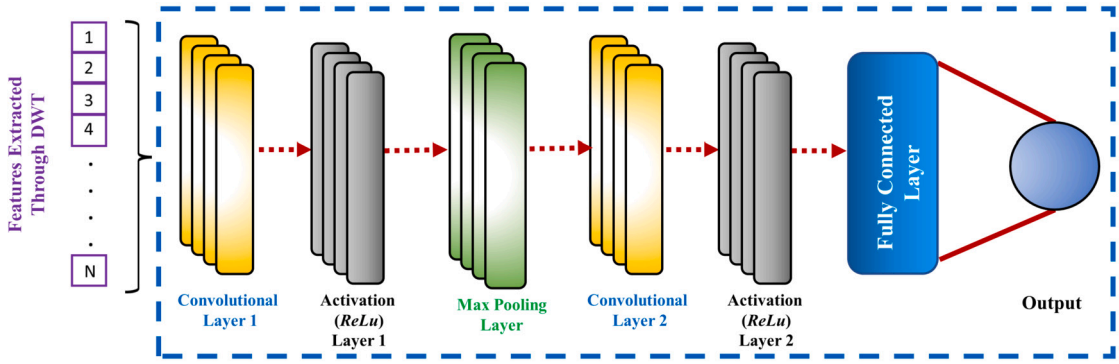


Fig. 3. Proposed stacked CNN architecture with detailed layer structure.

sampling by a factor of 2. The Daubechies 10 (db10) wavelet is a popular wavelet used in the DWT due to its good time-frequency localization and smoothness properties. The db10 wavelet is obtained by applying a series of high-pass and low-pass filters to a scaling function $\phi(n)$, which is a piecewise polynomial function of degree 9. The db10 wavelet has 20 filter coefficients, which can be computed using the following recursive equations:

$$h_0 = \frac{1+\sqrt{3}}{4\sqrt{2}} \tag{10}$$

$$h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}} \tag{11}$$

$$h_2 = \frac{3-\sqrt{3}}{4\sqrt{2}} \tag{12}$$

$$h_3 = \frac{1-\sqrt{3}}{4\sqrt{2}} \tag{13}$$

$$h_{4+i} = (-1)^i h_{3-i}, \quad i = 0,1,2,3 \tag{14}$$

$$g_0 = h_3 \tag{15}$$

$$g_1 = -h_2 \tag{16}$$

$$g_2 = h_1 \tag{17}$$

$$g_3 = -h_0 \tag{18}$$

$$g_{4+i} = (-1)^i g_{3-i}, \quad i = 0,1,2,3 \tag{19}$$

where h_i and g_i are the filter coefficients for the low-pass and high-pass filters, respectively. The first four coefficients, h_0 to h_3, are the coefficients for the low-pass filter, and the remaining coefficients, h_4 to h_9 and g_4 to g_9, are the coefficients for the high-pass filter. The remaining filter coefficients are obtained by applying a symmetry condition to the first four coefficients for both the low-pass and high-pass filters.

In this study, we specifically opted for the Daubechies 10 (db10) wavelet as our primary mother wavelet function for conducting the Discrete Wavelet Transform (DWT). The choice of db10 wavelet was grounded in its advantageous properties that render it particularly suitable for our intended analysis. Firstly, db10 exhibits remarkable time-frequency localization, effectively capturing transient features and local patterns within the surface electromyography (sEMG) signals. This precision in localization is pivotal for an accurate representation of the signal's dynamics. Moreover, db10 stands out for its smoothness, a characteristic that circumvents the issue of abrupt discontinuities that may manifest with other wavelet functions. This attribute significantly contributes to the stability of the feature extraction process, ensuring a reliable and consistent analysis of the signals. Another notable quality of db10 is its possession of 10 vanishing moments, an aspect critical for effectively representing complex signals by suppressing higher-order polynomial behaviors. Furthermore, the db10 wavelet encompasses a set of 20 filter coefficients, computed through recursive relationships as defined in the relevant literature. This distinctive feature equips the db10 wavelet with an optimal filter length, enhancing its efficacy in signal processing and analysis. The careful consideration of these properties collectively informed our decision to employ the db10 wavelet as a fundamental tool for the wavelet-based analysis of sEMG signals in this study.

Regarding the determination of decomposition levels for the Discrete Wavelet Transform (DWT), a deliberate and empirical approach was taken, resulting in the application of a 4-level decomposition to the surface electromyography (sEMG) signals. This decision was reached through thorough experimentation involving various levels, ranging from 2 to 6, with the aim of finding the most effective and suitable depth for our analysis. After comprehensive testing, it became evident that a 4-level decomposition struck an optimal balance between frequency resolution and feature dimensionality for the hand gestures under examination. Lower decomposition levels were found to lack the necessary frequency resolution, potentially leading to an inadequate representation of signal nuances. On the other hand, higher decomposition levels presented a challenge by introducing excessive feature dimensions without a proportional gain in informative signal characteristics. The 4-level decomposition was a judicious choice, offering a well-rounded solution by providing localized frequency information across discernible sub-bands. This localization was vital for ensuring robust and effective feature extraction specifically tailored to the nuances of hand gestures. The sub-band distribution achieved through this decomposition proved to be particularly conducive to the accurate and meaningful extraction of features from the sEMG signals associated with the hand gestures being studied. Therefore, the rationale behind selecting the Daubechies 10 (db10) wavelet function and implementing a 4-level DWT decomposition was grounded in achieving an optimal trade-off between frequency resolution and feature dimensionality, ultimately enhancing the efficacy of the feature extraction process crucial for analyzing hand gestures in this research. These carefully considered design choices shed light on the thoughtfully constructed DWT-based feature extraction methodology utilized in this study.

2.3. CNN

Convolutional Neural Networks (CNNs) are a class of deep learning models that have proven to be highly effective in many applications [34]. CNNs are particularly well-suited for tasks involving spatial and temporal data, such as images, videos, and time-series data. CNN network uses convolutional layers to automatically learn hierarchical representations of the input data, which are then fed into fully connected layers for classification or regression. The detailed structure of stacked CNN is shown in Fig. 3.

A one-dimensional CNN is a variant of the standard CNN architecture that is designed for processing one-dimensional input data, such as time-series data or sequences of feature vectors [35]. In a 1D CNN, the

Table 1
Range of hyperparameters of CNN.

Parameter	Range
No. of Filters	[2^0-2^9]
Filter Size in Each Layer	[1-7]
Activation Functions	LeakyReLu, ReLu, Tanh
Learning Rate	[10^-5-10^-1]
Dropout Rate	[0-0.7]

input data is convolved with a set of filters, each of which slides over the input in a single dimension. The output of each filter is then passed through ReLu activation, before being downsampled using max pooling or average pooling.

The mathematical equations for a 1D CNN can be expressed as follows. Given an input signal $\mathbf{x} \in \mathbb{R}^{T\times C}$ and a filter $\mathbf{W}_k \in \mathbb{R}^{F\times C}$, where T is the length of the signal, C is the number of channels, and F is the filter size.

The convolution of the k-th filter with the input signal is computed by sliding the filter over the input channels and summing the element-wise product at each position, including a bias term b_k:

$$(W_k * x)[t] = \sum_{i=1}^{C}\sum_{j=0}^{F-1} w_{ki,j} \cdot x_{t+j,i} + b_k, \quad t = 1, 2, \ldots, T - F + 1, \tag{20}$$

where $w_{ki,j}$ is the element at the i-th row and j-th column of the filter $\mathbf{W}_k$, $x_{t+j,i}$ is the element in the i-th channel at position $t + j$ in the input signal, and b_k is the bias term for the k-th filter.

Apply the activation function $f(\cdot)$ to each element of the convolution result:

$$z_k[t] = f((W_k * x)[t]), \quad t = 1, 2, \ldots, T - F + 1 \tag{21}$$

The activation function introduces nonlinearity into the model. The result of the convolution and activation for the k-th filter forms a feature map $\mathbf{z}_k$:

$$z_k = [f((W_k * x)[1]), f((W_k * x)[2]), \ldots, f((W_k * x)[T - F + 1])] \tag{22}$$

The feature map has a length of $(T - F + 1)$. Repeat the above steps for each filter k to obtain the complete set of feature maps $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_K$.

$$\mathbf{z}_k = [f((W_k * x)[1]), f((W_k * x)[2]), \ldots, f((W_k * x)[T - F + 1])] \tag{23}$$

for $k = 1, 2, \ldots, K$.

One of the main benefits of using 1D CNNs is their ability to learn meaningful features. 1D CNNs can also capture local dependencies in the data, allowing for the detection of patterns that may not be visible at the global level. Moreover, 1D CNNs can handle variable-length time series data and are robust to noise and missing values. Therefore, 1D CNNs offer a powerful and flexible approach to time series analysis that can yield state-of-the-art results in a wide range of applications.

2.4. Hyperparameters of SCNN

Hyperparameters are vital components that significantly influence the performance and optimization of a stacked CNN for classification tasks. They dictate the architecture and behavior of the network, and proper selection and optimization are crucial for enhancing accuracy, convergence speed, and generalization capability. The range of hyperparameters of SCNN is shown in Table 1.

2.4.1. Number and size of filters

The number and size of filters determine the receptive field of the network. While a larger number of filters can capture more diverse features, it also increases computational complexity. Striking a balance between the number and size of filters is important to extract relevant features efficiently.

2.4.2. Kernel size

The kernel size defines the size of the convolutional window moving across the input. A smaller kernel size can capture local details, while a larger kernel size can capture more global patterns. Selecting an appropriate kernel size depends on the characteristics of the input data and the complexity of the classification task.

2.4.3. Stride and padding

The stride determines the step size of the convolutional window during the convolution operation. Larger stride values reduce the spatial dimensions of the output feature maps, resulting in faster processing but potentially losing fine-grained details. Padding can be used to preserve spatial dimensions by adding zeros around the input. Proper stride and padding selection help maintain relevant information while controlling computational requirements.

2.4.4. Pooling

Pooling layers reduce the spatial dimensions of the feature maps, aiding in translation invariance and reducing computation. Pooling can be performed using operations like max pooling or average pooling. The choice of pooling size affects the amount of downsampling and the retention of important features.

2.4.5. Learning rate

The learning rate determines the step size during the optimization process. A high learning rate may lead to overshooting and failure to converge, while a low learning rate can slow down the convergence or get stuck in local optima. Tuning the learning rate is essential to ensure efficient convergence and accurate classification.

2.4.6. Regularization

Regularization techniques such as dropout and weight decay are crucial for preventing overfitting, especially when dealing with limited training data. The choice of regularization strength can significantly affect the model's generalization ability.

2.5. Importance and difficulty of optimization

Optimizing the hyperparameters of a CNN is a critical aspect of designing an effective neural network for a specific task, such as image classification. Hyperparameters are configurations that dictate the architecture, behavior, and training process of the neural network, distinct from the model's learnable parameters (weights and biases). Properly chosen hyperparameters can significantly influence the network's performance, convergence speed, generalization ability, and resource efficiency.

2.5.1. Effect on model performance

The hyperparameters, such as the number of filters, filter sizes, learning rates, and activation functions, directly affect the model's ability to learn intricate patterns and features from the input data. For instance, an optimal learning rate can ensure faster convergence and better accuracy, while an unsuitable one might lead to overshooting or slow convergence.

2.5.2. Generalization and overfitting

Hyperparameters play a pivotal role in combating overfitting, a situation where the model learns to memorize the training data instead of learning useful patterns. Techniques like dropout rates and weight regularization are hyperparameters crucial for improving generalization, preventing overfitting, and making the model perform well on unseen data.

2.5.3. Search space and optimization difficulty

The space of possible hyperparameters is vast, and the effect of each hyperparameter is often interdependent and non-linear. This complexity makes manual selection impractical. Algorithms such as grid search,

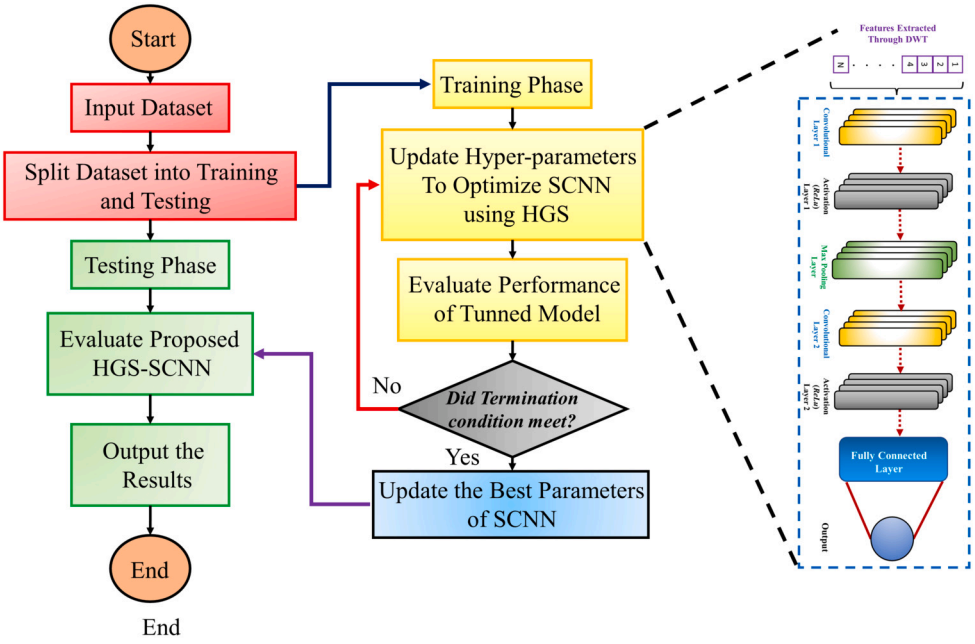


Fig. 4. Hyperparameter tuning flow of proposed HGS-SCNN technique.

random search, Bayesian optimization, and evolutionary methods like genetic algorithms attempt to navigate this expansive search space efficiently.

2.5.4. Computation and time complexity

Optimizing hyperparameters involves training and evaluating multiple models, making it computationally expensive and time-consuming, especially for deep neural networks. The need for substantial computational resources adds to the challenge, particularly when dealing with large datasets and intricate CNN architectures.

2.5.5. Trial and error experimentation

Finding the optimal set of hyperparameters usually involves a trial-and-error approach, where various combinations are tested. This iterative process can be laborious and requires a good understanding of the problem, the model, and the dataset.

2.6. Henry gass solubility based SCNN (HGS-SCNN)

As described above, the main demerit of the CNN architecture is that it involves a large number of hyperparameters, including the size of the filters, the number of filters, and the learning rate for the optimizer. Tuning these hyperparameters can be a time-consuming and challenging process, requiring extensive trial and error experimentation.

To address these challenges, researchers employ various techniques like grid search [36], random search [37], Bayesian optimization, or evolutionary intelligence-based methods such as genetic algorithms (GAs) [38], particle swarm optimization (PSO) or Grey wolf optimizer (GWO). These algorithms are designed to efficiently search for optimal hyperparameters by exploring the hyperparameter space using heuristic techniques and mathematical optimization methods. In this work, we employed the Henry Gas Solubility (HGS) algorithm to tune the hyperparameters of the CNN architecture. The proposed flow of HGS based SCNN model is shown in Fig. 4, while the tuned hyperparameters of the SCNN model are shown in Table 2.

2.7. Motivation of using HGS algorithm

The selection of the Henry Gas Solubility (HGS) algorithm for hyperparameter tuning in the Stacked Convolutional Neural Network (SCNN) is underpinned by the aspiration for an optimization technique that resonates with the intrinsic nature of the hand gesture recognition problem. The HGS algorithm, inspired by gas solubility principles, presents a nature-inspired optimization approach. Emulating the behavior of gas molecules in confined spaces, it offers a novel perspective to solving optimization problems. One of the pivotal motivations for employing this approach is its efficiency in exploring the high-dimensional hyperparameter space characteristic of SCNNs. By simulating the diffusion of gas molecules, the algorithm strives to efficiently navigate this space, crucial for discovering an optimal set of hyperparameters significantly impacting SCNN performance. Moreover, the gas-inspired exploration carries the potential to achieve global optimization, a desirable trait for developing a robust SCNN model with improved generalization capabilities, particularly in the domain of hand gesture recognition. This approach aligns seamlessly with the fundamental design principles of CNNs, particularly in feature extraction tasks, making it a suitable choice for optimizing SCNNs.

Table 2
Hgs based Tuned Hyperparameters of SCNN.

Parameter	Range
No. of Filters	64
Filter Size in Each Layer	3
Activation Functions	ReLu
Learning Rate	10^{-2}
Dropout Rate	0.5

3. Dataset collection and processing

3.1. Dataset generation

The dataset consists of surface electromyography (sEMG) signals recorded from 2 sensors interfaced with an Arduino MEGA 2560 microcontroller. Data are collected from 5 subjects performing 6 different gestures and every gesture is repeated 20 times, resulting in a comprehensive dataset for training and evaluation. The hardware setup involves connecting the Arduino MEGA 2560 with MATLAB Simulink 2022a. This integration allows for real-time data acquisition and communication with the microcontroller board. Two sEMG sensors are

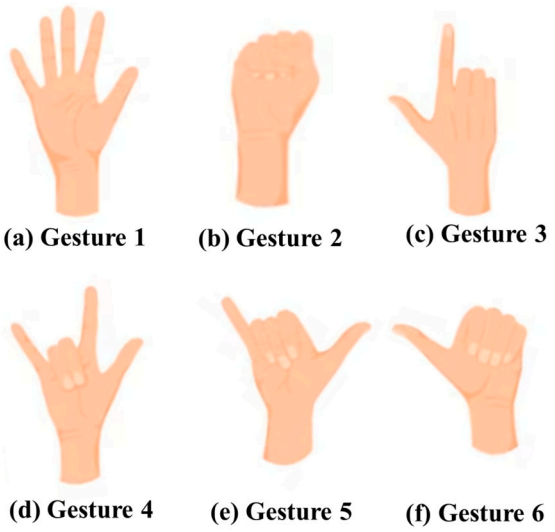


Fig. 5. Hand gestures used for dataset generation and classification.

Table 3
Specifications for the hand gesture dataset generation.

Variable	Value
No. of Subject	5
No. of Gestures	6
No. of Channels	2
No. of Repetitions	20
Sampling Frequency (Hz)	1000
Activity Duration (s)	2
Rest Time (s)	2

connected to the Arduino MEGA, providing simultaneous recording of muscle activation signals from multiple hand muscles. The six hand gestures used for dataset generation are shown in Fig. 5.

The sEMG sensors used in this study are non-invasive electrodes that detect electrical signals generated by muscle contractions. These sensors are carefully placed on specific hand muscles to capture the corresponding muscle activation patterns during hand gestures. Subjects were instructed to position the sensors according to standardized electrode placement guidelines. A well-defined gesture protocol was employed to ensure consistency across data collection sessions. Each hand gesture consisted of an action phase and a rest phase, both lasting 2 seconds. During the action phase, subjects were instructed to perform the target hand gesture, while the rest phase involved relaxation with no intentional muscle activity. This protocol aimed to capture the distinct sEMG patterns associated with each gesture.

Data collection sessions were conducted with 5 subjects, who were briefed about the experimental procedure and provided informed consent. Each subject performed the 6 hand gestures, with 20 repetitions per gesture. The order of gestures was randomized to minimize any potential order effects. Subjects were given adequate rest intervals between repetitions and gestures to minimize muscle fatigue. During the data collection process, the sEMG signals were continuously recorded from the sensors at a sampling rate of 1000 Hz. This high sampling rate ensured capturing fine-grained details of the muscle activation signals. The acquired signals were transmitted in real-time from the Arduino MEGA to MATLAB Simulink for further processing and storage. The dataset generated from the data collection process comprised a total of 120 instances for each subject (20 repetitions x 6 gestures). Considering the 5 subjects, the final dataset consisted of 600 instances. This sufficiently large dataset facilitates robust training and evaluation of hand gesture recognition models. The specifications of the dataset generation are shown in Table 3.

3.2. Filtering

Filtering is a crucial preprocessing step in enhancing the quality and reliability of electromyography (EMG) signals. In this study, a bandpass filter with a frequency range of 20 Hz to 300 Hz was employed to selectively pass signals within the desired frequency band while attenuating frequencies outside this range. This specific frequency range was chosen based on its biological relevance to muscle activation patterns and the need to remove low-frequency noise and high-frequency interference [39]. By implementing the bandpass filter, unwanted noise, such as baseline drift and power line interference, was effectively eliminated, allowing for a clearer representation of the underlying muscle activity.

The implementation of the bandpass filter involved employing suitable digital signal processing techniques. The filter's performance was evaluated by assessing its frequency response, magnitude response, and phase response. This evaluation ensured that the bandpass filter effectively attenuated out-of-band noise while preserving the relevant frequency components within the 20 Hz to 300 Hz range.

3.3. Feature extractions

The analysis of sEMG sensor data of hand gestures involves a multi-step process to extract meaningful information and features. The initial step in this flow is to calculate the moving average of the signal. This is accomplished by applying a sliding window technique, where the average value of the signal within a specific window size is computed. The moving average helps to reduce noise and smooth out the signal, enhancing the visibility of underlying patterns and features related to hand gestures. By utilizing this technique, the overall signal quality is improved, enabling subsequent analysis steps to be more effective.

Following the calculation of the moving average, the next crucial step is EMG onset detection. EMG onset refers to the initiation of muscle activity associated with the hand gesture. Detecting the precise onset of EMG activity is vital for accurately capturing the relevant data during the hand gesture performance. Various techniques can be employed for EMG onset detection, such as amplitude threshold-based methods, slope-based methods, or ML algorithms. These approaches analyze the characteristics of the moving average signal and identify the point at which the EMG activity exceeds a certain threshold or exhibits a significant change, indicating the start of the hand gesture.

After identifying the EMG onset, a sample window of 300 milliseconds is selected from the original signal. This window represents a segment of the signal that encapsulates the duration of the hand gesture. In this segment, the DWT is performed to extract valuable features. DWT reveals both the time and frequency domain information simultaneously. By applying DWT to the sample window, the signal is analyzed at various scales or levels of resolution, providing a multi-resolution representation of the hand gesture. Features such as amplitude, frequency content, and energy distribution across different frequency sub-bands can be extracted from the DWT coefficients. These features capture important characteristics of the hand gesture, enabling further analysis, classification or recognition tasks. The comparison of extracted features using DWT for different gestures is shown in Fig. 6.

The Discrete Wavelet Transform (DWT) plays a crucial role in the process of extracting distinct and discernible features from surface electromyography (sEMG) signals for hand gesture recognition. By employing wavelet functions, the DWT breaks down the signal into various frequency sub-bands at different scales, revealing both time and frequency domain information in a simultaneous manner. The lower frequency bands provide insight into the global contour and overall trends present within the signal, while the higher frequency bands capture transient spikes and local patterns. Through wavelet coefficients, the distribution of energy across these sub-bands is made evident, showcasing how different gestures manifest distinct coefficient distributions. For instance, a pinch gesture may exhibit a higher concentration of high-frequency coefficients compared to a grip gesture. The advantage of wavelets lies

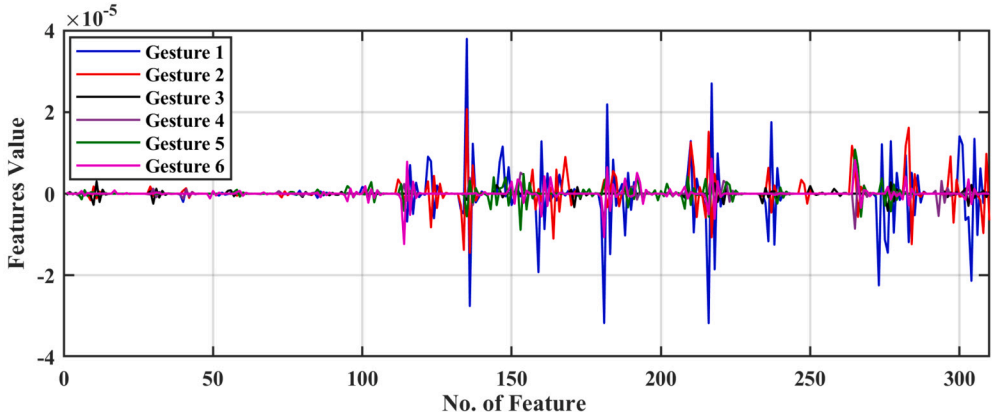


Fig. 6. Extracted features through DWT from Channel 1.

in their ability to be localized in time, enabling the capture of transient muscle activation spikes. The multi-resolution view that DWT offers accentuates the nuances tied to gesture dynamics, revealing patterns that might not be visible when examining just the raw signal. Therefore, the DWT not only surfaces hidden patterns but also accentuates subtleties, providing an augmented feature space when compared to using solely raw sEMG data or basic statistical measures. In summary, DWT decomposes the signal in a manner that uncovers unique spatio-temporal and spectral characteristics of muscle activity associated with each gesture, facilitating a more robust feature extraction and discrimination compared to utilizing only the original sEMG recordings. Feel free to let me know if this elucidation adequately conveys how DWT contributes to distinctive feature extraction.

3.4. Dataset pre-processing

Normalizing the data is an essential step in data preprocessing, as it helps to improve the performance of many ML algorithms. One of the most common normalization techniques is the min-max scaling technique. In this technique, the values of a feature are scaled to a range between 0 and 1. The min-max scaling technique is given by the following equation:

$$X_{norm} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{24}$$

where X is current sample, X_{min} and X_{max} are the min. and max. values of the sample, and X_{norm} is the normalized sample value.

3.5. Evaluation matrices

Evaluation matrices are commonly used to assess the performance of classification models, including those used for hand gesture recognition. Here, we present four widely used evaluation matrices: Accuracy, Precision, Specificity, and F1 score. These matrices depend upon the True positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) of the predicted classes.

3.6. Accuracy

Accuracy is the ratio of the correctly predicted samples to the total number of samples in the dataset;

$$Accuracy = \frac{TP + TN}{Total\ number\ of\ samples} \tag{25}$$

3.7. Precision

Precision is a metric that indicates how well the model performs in terms of minimizing false positives:

$$Precision = \frac{TP}{TP + FP} \tag{26}$$

3.8. Specificity

Specificity is a metric that indicates how well the model performs in terms of minimizing false negatives:

$$Specificity = \frac{TN}{TN + FP} \tag{27}$$

3.9. F1 score

The F1-score is calculated as:

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{28}$$

These evaluation matrices help in assessing the performance of hand gesture classification models and provide insights into the model's accuracy, precision, specificity, and overall effectiveness.

3.10. Proposed scheme

The flow of analysis for sEMG sensor data of hand gestures involves multiple steps. First, the moving average of the signal is calculated to reduce noise and enhance the visibility of underlying patterns. Next, the EMG onset is detected to pinpoint the time window associated with the hand gesture. Subsequently, a sample window is selected, and the DWT is applied to extract informative features from the signal. This systematic approach provides a scientific framework for processing and analyzing sEMG data, facilitating the understanding and interpretation of hand gestures for various applications such as prosthetics, rehabilitation, or human-computer interaction systems. After feature extraction and pre-processing, the dataset is divided into 70-30% training and testing dataset ratios. After that, the HGS-based SCNN model is trained on training data and tested on the testing dataset to check the performance of the proposed model. The detailed DWT-based HGS-SCNN scheme for hand gesture detection is elaborated in Fig. 7. Fig. 8 shows the loss and accuracy of tuned SCNN model. A detailed analysis on the results is presented in the next section.

4. Results and analysis

4.1. Prediction performance

The evaluation of prediction performance for different techniques based on stacked CNN networks (SCNN) in classifying six distinct hand gestures is presented in Table 4. To delve deeper into the predictive abilities of these techniques, a thorough analysis utilizing confusion matrices (as depicted in Fig. 9) and a comparative evaluation of key

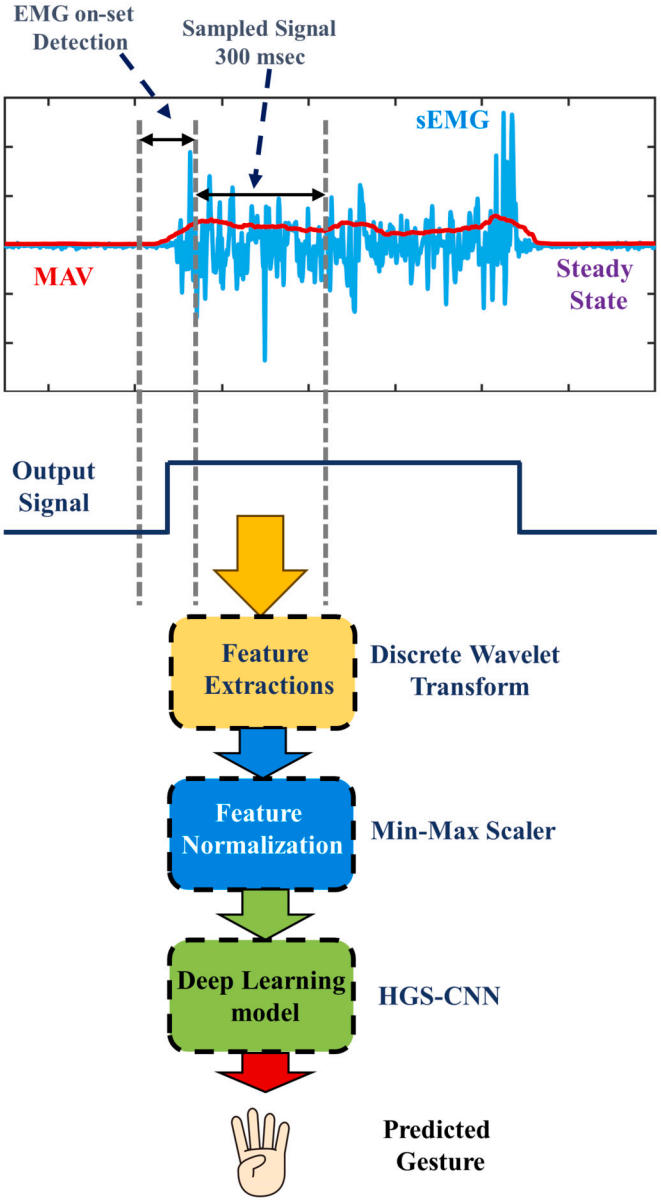


Fig. 7. Proposed DWT-based HGS-CNN technique.

Table 4
Gesture prediction comparative analysis.

Technique	Accuracy	Precision	Specificity	F1 Score
HGS-SCNN	0.9944	0.9944	0.9989	0.9944
AOA-SCNN	0.9778	0.9778	0.9756	0.9778
WOA-SCNN	0.9611	0.9611	0.9622	0.9613
GWO-SCNN	0.9556	0.9556	0.9512	0.9557

metrics (shown in Fig. 10) is conducted. A comprehensive breakdown and analysis of these results are presented in the subsequent subsections.

4.1.1. Accuracy

Accuracy is a fundamental metric reflecting the proportion of correctly predicted instances out of the total. Among the techniques analyzed, HGS-SCNN stands out with the highest accuracy of 0.9944. This implies an outstanding ability to predict hand gestures with a staggering accuracy rate of 99.44%. AOA-SCNN closely follows with an accuracy of 0.9778, indicating a slightly lower but still impressive accuracy rate

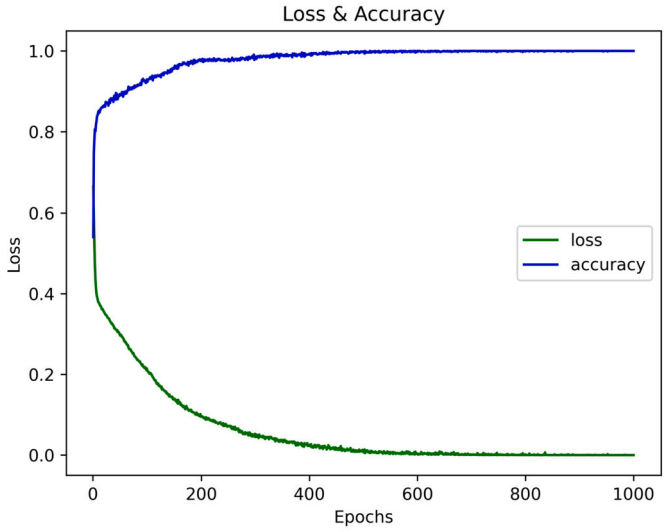


Fig. 8. Loss and accuracy curve of the HGS tuned SCNN model.

of 97.78%. WOA-SCNN achieves an accuracy of 0.9611, demonstrating a high precision but slightly less than the previous two techniques. GWO-SCNN, while effective, exhibits the lowest accuracy among the listed techniques, with a value of 0.9556.

4.1.2. Precision

Precision is a crucial measure denoting the proportion of true positive predictions out of all predicted positives. HGS-SCNN leads in precision with a score of 0.9944, implying that 99.44% of the hand gestures predicted as positive by HGS-SCNN are indeed correct. AOA-SCNN closely follows with a precision score of 0.9778, indicating a high level of precision in its predictions. WOA-SCNN and GWO-SCNN also demonstrate substantial precision scores of 0.9611 and 0.9556, respectively, signifying a high level of correctness in their positive predictions.

4.1.3. Specificity

Specificity measures the ability to identify non-target gestures accurately. HGS-SCNN excels in specificity, achieving the highest score of 0.9989. This suggests that HGS-SCNN identifies non-target gestures with an impressive accuracy rate of 99.89%. AOA-SCNN follows with a specificity of 0.9756, indicating a high level of accuracy in identifying non-target gestures. WOA-SCNN and GWO-SCNN also exhibit commendable specificity scores of 0.9622 and 0.9512, respectively, underscoring their ability to discern non-target gestures with substantial accuracy.

4.1.4. F1 score

The F1 score, a balanced metric considering both precision and recall, provides a comprehensive assessment of the overall predictive performance. HGS-SCNN achieves an F1 score of 0.9944, suggesting a harmonious trade-off between precision and recall and indicating a high overall performance. AOA-SCNN closely follows with an F1 score of 0.9778, representing a balanced performance in terms of precision and recall. WOA-SCNN and GWO-SCNN also present respectable F1 scores of 0.9613 and 0.9557, respectively. These scores emphasize the ability of these techniques to strike a balance between precision and recall, contributing to a robust overall predictive performance.

Based on the results presented in Table 4, HGS-SCNN stands out by showcasing superior performance compared to the other techniques. Several key aspects highlight its exceptional capabilities:

- High Accuracy: HGS-SCNN attains the highest accuracy among all the techniques, signifying its remarkable precision in predicting hand gestures.

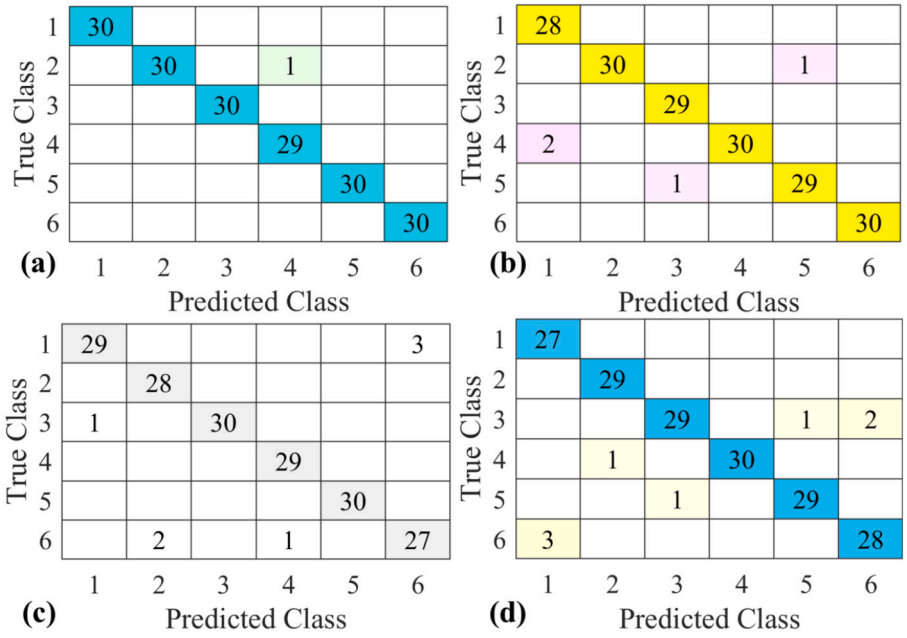


Fig. 9. Confusion matrix comparison of (a) HGS-SCNN, (b) AOA-SCNN, (c) WOA-SCNN and (d) GWO-SCNN.

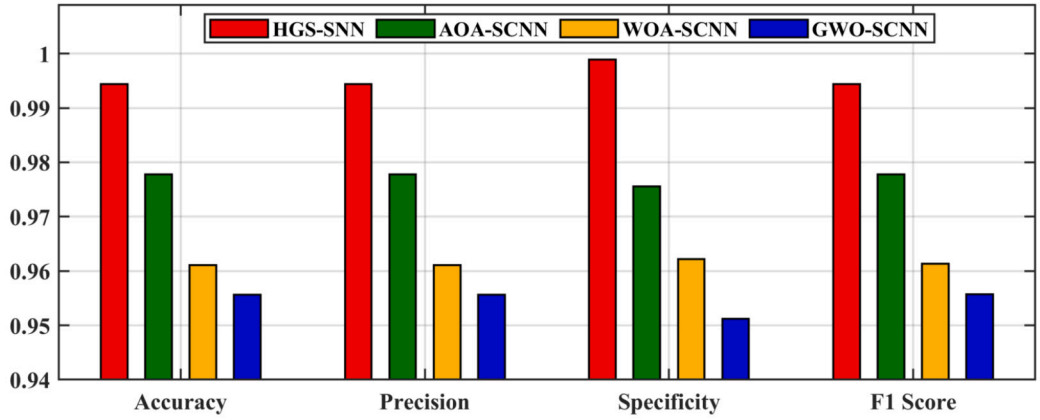


Fig. 10. Bar graph comparison of competing techniques.

- High Precision: HGS-SCNN achieves an impressive precision score of 0.9944, indicating an extremely low false positive rate. Consequently, the gestures predicted as positive by HGS-SCNN are highly likely to be accurate.
- High Specificity: HGS-SCNN secures the highest specificity among the techniques, showcasing its remarkable ability to precisely identify non-target gestures. This aspect is crucial in avoiding false positives.
- High F1 Score: HGS-SCNN achieves a well-balanced F1 score of 0.9944, showcasing its adeptness in maintaining an effective trade-off between precision and recall. This score signifies that HGS-SCNN excels in minimizing both false positives and false negatives.

4.2. Comparative analysis

The Table 5 presented provides a comparative analysis of various hand gesture detection techniques found in the literature. Each technique is evaluated based on the dataset used, the specific methodology employed, and the accuracy achieved. Among the listed approaches, our proposed technique stands out as a superior solution.

Firstly, the proposed technique utilizes a comprehensive dataset consisting of six hand gestures. In contrast, other references in the table used datasets with specific sign language gestures, spatial data, or a limited number of static images. The breadth and depth of our dataset suggest a more representative and robust training environment. Secondly, the proposed technique leverages a stacked convolutional neural network (CNN) algorithm for hand gesture detection. CNNs have proven to be highly effective in image-related tasks, thanks to their ability to identify patterns and extract features from visual data. This choice of algorithm showcases the sophistication and advanced nature of our approach. In comparison, the other techniques in the table include feature selection with ANN, RBF functions, SVM with feature selection, and deep neural networks (DNN). While these techniques are valuable in their respective contexts, our utilization of a stacked CNN algorithm demonstrates a more cutting-edge and potentially more accurate approach. Lastly, the proposed technique achieves an outstanding accuracy of 0.993, surpassing the accuracies reported in the other references. The highest accuracy in the table is 0.98, achieved by a DNN-based approach for a limited set of three hand gestures. The significantly higher accuracy of our proposed technique highlights its exceptional capability in accurately classifying and recognizing hand gestures. This level of accuracy is vital for real-world applications where precise and reliable gesture detection is required.

Table 5
Comparative analysis of hand gesture detection presented in literature.

Ref.	Data-set	Technique Used	Acc.
[40]	Indian Sign Language (Pictorial 6 static colored images)	Feature Selection + NN for Classification (trained using hybrid meta-heuristic deer hunting + gwo)	0.97
[41]	Spatial Data-set (Arabic Sign Language Numerals 1-9)	Radial Basis Function	0.942
[42]	Static 9 gestures	FS (static; based on Orientation) + SVM Classification	0.9137
[43]	Six static images	HOG + SVM (multi-class)	0.92
[44]	Three static gestures (Rock, paper, scissors)	FS (static 11 points) + DNN Classification	0.98
Our approach	Six Hand gestures	HGS Algorithm based Stacked CNN	0.993

5. Discussion

In this section, we delve into the insights and implications drawn from the presented results and the comparative analysis of hand gesture detection techniques.

5.1. Performance comparison and interpretations

The evaluation of prediction performance using stacked CNN networks (SCNN) for hand gesture classification has showcased the outstanding performance of the HGS-SCNN technique. With an accuracy of 0.9944, HGS-SCNN has demonstrated a remarkable ability to accurately predict hand gestures. Such high accuracy is of paramount importance, particularly in applications like human-computer interaction and robotics, where precise gesture recognition is a critical factor. The high precision of HGS-SCNN (0.9944) underscores its capability to maintain an extremely low false positive rate, implying that the predicted hand gestures are highly likely to be correct. This characteristic is vital in applications where inaccurate predictions could lead to adverse outcomes. Furthermore, HGS-SCNN exhibited the highest specificity (0.9989) among the techniques analyzed, showcasing its proficiency in accurately identifying non-target gestures. This aspect is crucial in gesture recognition systems to avoid false positives, which can be particularly detrimental in applications such as medical diagnostics. The well-balanced F1 score of 0.9944 achieved by HGS-SCNN emphasizes its effectiveness in maintaining a trade-off between precision and recall. This is a vital characteristic for achieving a high-performing model that minimizes both false positives and false negatives.

In contrast to the HGS-SCNN model, the competitive techniques, namely GWO-SCNN, WOA-SCNN, and AOA-SCNN, exhibit suboptimal performance in the classification of hand gestures, yielding comparatively lower values across evaluation matrices. Specifically, GWO-SCNN encounters challenges in effectively fine-tuning the Stacked CNN (SCNN) model, resulting in a suboptimal resolution with convergence being hindered in local minima during the cost reduction process. The resultant accuracy achieved by GWO-SCNN is recorded at 95.56%. AOA-SCNN and WOA-SCNN demonstrate relatively higher accuracy in comparison to GWO-SCNN. AOA-SCNN leverages a superior exploration strategy, enabling it to navigate away from local minima, thereby contributing to its elevated accuracy. Similarly, WOA-SCNN, while exhibiting less accuracy than AOA-SCNN, outperforms GWO-SCNN, showcasing a more proficient optimization in the SCNN model tuning process. AOA-SCNN and WOA-SCNN attain accuracy values of 97.78% and 96.11%, respectively. However, it is noteworthy that the HGS-SCNN model surpasses GWO-SCNN, WOA-SCNN, and AOA-SCNN across all evaluated metrics. The efficacy of HGS-SCNN is attributed to its adeptness in avoiding local minima through a judicious interplay of exploration and exploitation phases. This attribute enhances the model's capacity for accurate classification of hand gestures, positioning it as a superior choice compared to GWO-SCNN, WOA-SCNN, and AOA-SCNN in the studied context.

5.2. Comparative analysis and key insights

The comparative analysis of various hand gesture detection techniques highlighted the strengths of the proposed HGS-SCNN approach. Our technique leveraged a comprehensive dataset encompassing six diverse hand gestures, providing a more representative and robust training environment compared to other techniques that used specific sign language gestures or a limited set of static images.

The adoption of a stacked convolutional neural network (CNN) algorithm demonstrated the advanced nature of our approach. CNNs are known for their effectiveness in image-related tasks due to their ability to identify intricate patterns and extract features from visual data. This choice of algorithm underlines our commitment to employing cutting-edge methodologies for hand gesture detection.

Lastly, achieving an accuracy of 0.993 with our proposed technique surpassed the accuracies reported in other references, underscoring its superior performance. This high accuracy level hints at the potential practical applicability of our approach in various domains.

5.3. Practical implications, challenges and future avenues

The superior performance of HGS-SCNN in hand gesture detection carries significant practical implications across multiple domains. In human-computer interaction, our technique has the potential to significantly enhance user experience by providing accurate and intuitive gesture-based control systems. In the field of robotics, precise gesture recognition can enable seamless and efficient control of robotic devices, contributing to advancements in automation and robotics.

While the proposed HGS-SCNN model shows impressive accuracy in hand gesture recognition, like any technology, it's not without its limitations. One potential challenge lies in variations in sensor conditions. Real-world scenarios may involve different environments, lighting conditions, or hardware variations, which could affect the performance of the model. The model might struggle to generalize well across diverse conditions not accounted for during training. Moreover, user-specific nuances can be a hurdle. People have unique ways of performing hand gestures, and individual differences in anatomy, muscle structure, or even the placement of the sEMG sensors can introduce variability. The model might not adapt perfectly to all users, potentially leading to lower accuracy or misclassifications for certain individuals. The effectiveness of the model could be influenced by the size and diversity of the dataset used for training. If the dataset does not adequately represent the wide range of potential users and scenarios, the model might not generalize well to unforeseen conditions. Furthermore, the reliance on the Discrete Wavelet Transform (DWT) for feature extraction might introduce limitations. While DWT is effective in capturing spatial and frequency information, it might not be optimal for all types of hand gestures or could be sensitive to certain signal variations. It's crucial to consider these limitations when implementing the HGS-SCNN model in practical applications. Ongoing research and refinement could address these

challenges, making the model more robust and adaptable to a broader range of conditions and user-specific nuances.

For future research, further exploration of deep learning architectures and optimization techniques could potentially elevate the accuracy and efficiency of hand gesture detection systems. Additionally, investigating the application of hand gesture detection in real-time and dynamic environments would provide valuable insights for practical deployments, such as in gaming, virtual reality, and assistive technologies.

6. Conclusion

This paper introduced a novel approach for hand gesture detection using a Henry Gas Solubility-based Stacked Convolutional Neural Network (HGS-SCNN). Hand gesture detection has become increasingly important in various domains, and our proposed approach offers enhanced accuracy and robustness in this task. The stacked architecture of the CNN model allows for effective representation learning by capturing both low-level and high-level features.

Through the utilization of the Discrete Wavelet Transform (DWT) technique for feature extraction, our approach successfully captures spatial and frequency information, leading to improved discriminative power in the extracted features. Extensive experiments were conducted using a dataset of 600 hand gesture samples, and the performance of the HGS-SCNN model was evaluated. Comparative analysis with SOTA techniques demonstrates the superiority of our proposed approach, achieving an impressive accuracy of 99.3%.

The results validate the effectiveness of our approach in accurately detecting hand gestures and highlight the potential of combining DWT-based feature extraction with the HGS-SCNN model. This combination offers reliable and robust hand gesture recognition, opening up new possibilities for intuitive human-computer or human-robot interaction and applications that require gesture-based control.

Possible future works that can be further carried out are elaborated below:

- Expanding the gesture vocabulary - The current 6 gestures, while covering a useful range, are still limited. Adding more complex uni-manual and bimanual gestures can enhance the system's capabilities.
- Evaluating personalized models - Training user-specific models that adapt to individual variations in muscle anatomy and sEMG patterns may further boost accuracy.
- Exploring sensor fusion - Supplementing sEMG with inertial or depth data could make the recognition more robust to ambiguities.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgement

This research is supported by the Biomechatronics and Collaborative Robotics research group at the Top Research Center Mechatronics (TRCM), University of Agder (UiA), Norway.

References

[1] O. Mazhar, B. Navarro, S. Ramdani, R. Passama, A. Cherubini, A real-time human-robot interaction framework with robust background invariant hand gesture detection, Robot. Comput.-Integr. Manuf. 60 (2019) 34–48.
[2] N. Mendes, Surface electromyography signal recognition based on deep learning for human-robot interaction and collaboration, J. Intell. Robot. Syst. 105 (2) (2022) 42.
[3] H. Chen, M.C. Leu, Z. Yin, Real-time multi-modal human–robot collaboration using gestures and speech, J. Manuf. Sci. Eng. 144 (10) (2022) 101007.
[4] H. Mohyuddin, S.K.R. Moosavi, M.H. Zafar, F. Sanfilippo, A comprehensive framework for hand gesture recognition using hybrid-metaheuristic algorithms and deep learning models, Array 19 (2023) 100317.
[5] S.A. Khomami, S. Shamekhi, Persian sign language recognition using imu and surface emg sensors, Measurement 168 (2021) 108471.
[6] F. Belmajdoub, S. Abderafi, Efficient machine learning model to predict fineness, in a vertical raw meal of Morocco cement plant, Results Eng. 17 (2023) 100833.
[7] J. Zhou, Y. Dai, M. Tao, M. Khandelwal, M. Zhao, Q. Li, Estimating the mean cutting force of conical picks using random forest with salp swarm algorithm, Results Eng. 17 (2023) 100892.
[8] A. Nazir, A.K. Shaikh, A.S. Shah, A. Khalil, Forecasting energy consumption demand of customers in smart grid using temporal fusion transformer (tft), Results Eng. 17 (2023) 100888.
[9] S. Sreelakshmi, G. Malu, E. Sherly, R. Mathew, M-net: an encoder-decoder architecture for medical image analysis using ensemble learning, Results Eng. 17 (2023) 100927.
[10] A. Phinyomark, E. Scheme, An investigation of temporally inspired time domain features for electromyographic pattern recognition, in: 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, 2018, pp. 5236–5240.
[11] A. Phinyomark, P. Phukpattaranont, C. Limsakul, Feature reduction and selection for emg signal classification, Expert Syst. Appl. 39 (8) (2012) 7420–7431.
[12] P. Shenoy, K.J. Miller, B. Crawford, R.P. Rao, Online electromyographic control of a robotic prosthesis, IEEE Trans. Biomed. Eng. 55 (3) (2008) 1128–1135.
[13] A.H. Al-Timemy, R.N. Khushaba, G. Bugmann, J. Escudero, Improving the performance against force variation of EMG controlled multifunctional upper-limb prostheses for transradial amputees, IEEE Trans. Neural Syst. Rehabil. Eng. 24 (6) (2015) 650–661.
[14] A. Waris, I.K. Niazi, M. Jamil, K. Englehart, W. Jensen, E.N. Kamavuako, Multiday evaluation of techniques for emg-based classification of hand motions, IEEE J. Biomed. Health Inform. 23 (4) (2018) 1526–1534.
[15] B. Fatimah, P. Singh, A. Singhal, R.B. Pachori, Hand movement recognition from semg signals using Fourier decomposition method, Biocybern. Biomed. Eng. 41 (2) (2021) 690–703.
[16] N.K. Karnam, A.C. Turlapaty, S.R. Dubey, B. Gokaraju, Classification of semg signals of hand gestures based on energy features, Biomed. Signal Process. Control 70 (2021) 102948.
[17] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[18] M. Atzori, M. Cognolato, H. Müller, Deep learning with convolutional neural networks applied to electromyography data: a resource for the classification of movements for prosthetic hands, Front. Neurorobot. 10 (2016) 9.
[19] M. Atzori, A. Gijsberts, C. Castellini, B. Caputo, A.-G.M. Hager, S. Elsig, G. Giatsidis, F. Bassetto, H. Müller, Electromyography data for non-invasive naturally-controlled robotic hand prostheses, Sci. Data 1 (1) (2014) 1–13.
[20] W. Geng, Y. Du, W. Jin, W. Wei, Y. Hu, J. Li, Gesture recognition by instantaneous surface EMG images, Sci. Rep. 6 (1) (2016) 36571.
[21] M.H. Zafar, M. Mansoor, M. Abou Houran, N.M. Khan, K. Khan, S.K.R. Moosavi, F. Sanfilippo, Hybrid deep learning model for efficient state of charge estimation of Li-ion batteries in electric vehicles, Energy 282 (2023) 128317.
[22] S.K. Raza Moosavi, M.H. Zafar, S. Mirjalili, F. Sanfilippo, Improved barnacles movement optimizer (ibmo) algorithm for engineering design problems, in: International Conference on Artificial Intelligence and Soft Computing, Springer, 2023, pp. 427–438.
[23] A. Muqeet, A. Israr, M.H. Zafar, M. Mansoor, N. Akhtar, A novel optimization algorithm based pid controller design for real-time optimization of cutting depth and surface roughness in finish hard turning processes, Results Eng. 18 (2023) 101142.
[24] Z.A. Kadhuim, S. Al-Janabi, Codon-mrna prediction using deep optimal neurocomputing technique (dlstm-dsn-woa) and multivariate analysis, Results Eng. 17 (2023) 100847.
[25] V. Balaji, S. Narendranath, et al., Optimization of wire-edm process parameters for Ni–Ti-Hf shape memory alloy through particle swarm optimization and CNN-based sem-image classification, Results Eng. 18 (2023) 101141.
[26] J.F. Ruma, M.S.G. Adnan, A. Dewan, R.M. Rahman, Particle swarm optimization based lstm networks for water level forecasting: a case study on Bangladesh river network, Results Eng. 17 (2023) 100951.
[27] P. Koch, M. Dreier, M. Maass, H. Phan, A. Mertins, Rnn with stacked architecture for semg based sequence-to-sequence hand gesture recognition, in: 2020 28th European Signal Processing Conference (EUSIPCO), IEEE, 2021, pp. 1600–1604.
[28] I. Ketykó, F. Kovács, K.Z. Varga, Domain adaptation for semg-based gesture recognition with recurrent neural networks, in: 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, 2019, pp. 1–7.
[29] Y. Hu, Y. Wong, W. Wei, Y. Du, M. Kankanhalli, W. Geng, A novel attention-based hybrid cnn-rnn architecture for semg-based gesture recognition, PLoS ONE 13 (10) (2018) e0206049.
[30] Y. Wang, Q. Wu, N. Dey, S. Fong, A.S. Ashour, Deep back propagation–long short-term memory network based upper-limb semg signal classification for automated rehabilitation, Biocybern. Biomed. Eng. 40 (3) (2020) 987–1001.

[31] V. Mohebbi, A. Naderifar, R. Behbahani, M. Moshfeghian, Determination of Henry's law constant of light hydrocarbon gases at low temperatures, J. Chem. Thermodyn. 51 (2012) 8–11.
[32] F.A. Hashim, E.H. Houssein, M.S. Mabrouk, W. Al-Atabany, S. Mirjalili, Henry gas solubility optimization: a novel physics-based algorithm, Future Gener. Comput. Syst. 101 (2019) 646–667.
[33] G. Othman, D.Q. Zeebaree, The applications of discrete wavelet transform in image processing: a review, J. Soft Comput. Data Min. 1 (2) (2020) 31–43.
[34] C. Tian, Y. Xu, Z. Li, W. Zuo, L. Fei, H. Liu, Attention-guided cnn for image denoising, Neural Netw. 124 (2020) 117–129.
[35] A. Kanwal, M.F. Lau, S.P. Ng, K.Y. Sim, S. Chandrasekaran, Bicudnnlstm-1dcnn—a hybrid deep learning-based predictive model for stock price prediction, Expert Syst. Appl. 202 (2022) 117123.
[36] F.J. Pontes, G. Amorim, P.P. Balestrassi, A. Paiva, J.R. Ferreira, Design of experiments and focused grid search for neural network parameter optimization, Neurocomputing 186 (2016) 22–34.
[37] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (2) (2012).
[38] M. Ilbeigi, M. Ghomeishi, A. Dehghanbanadaki, Prediction and optimization of energy consumption in an office building using artificial neural network and a genetic algorithm, Sustain. Cities Soc. 61 (2020) 102325.
[39] S. Ma, B. Lv, C. Lin, X. Sheng, X. Zhu, Emg signal filtering based on variational mode decomposition and sub-band thresholding, IEEE J. Biomed. Health Inform. 25 (1) (2020) 47–58.
[40] M. Kowdiki, A. Khaparde, Automatic hand gesture recognition using hybrid meta-heuristic-based feature selection and classification with dynamic time warping, Comput. Sci. Rev. 39 (2021) 100320, https://doi.org/10.1016/j.cosrev.2020.100320.
[41] W. Zeng, C. Wang, Q. Wang, Hand gesture recognition using leap motion via deterministic learning, Multimed. Tools Appl. 77 (2018) 28185–28206.
[42] Y. Xu, Q. Wang, X. Bai, Y.-L. Chen, X. Wu, A novel feature extracting method for dynamic gesture recognition based on support vector machine, in: 2014 IEEE International Conference on Information and Automation (ICIA), 2014, pp. 437–441.
[43] K.-p. Feng, F. Yuan, Static hand gesture recognition based on hog characters and support vector machines, in: 2013 2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation (IMSNA), 2013, pp. 936–938.
[44] Q. Yang, W. Ding, X. Zhou, D. Zhao, S. Yan, Leap motion hand gesture recognition based on deep neural network, in: 2020 Chinese Control and Decision Conference (CCDC), 2020, pp. 2089–2093.