Acta Scientiarum

http://periodicos.uem.br/ojs/acta
ISSN on-line: 1807-8664
Doi: 10.4025/actascitechnol.v47i1.70901

ELECTRICAL ENGINEERING

Unlocking precision using *k*-means++- improved genetic algorithm-radial basis function neural network: data-driven evolution of smart gloves for gesture recognition

Liang Xiao Ding[1], Kuan-Way (Guanghui) Chee[1*], Hong Lü[2], Anand Paul[3], Jeonghong Kim[4] and Jang-Myung Lee[5]

[1]School of Electronic and Electrical Engineering, Kyungpook National University, Daegu, 41566, Republic of Korea. [2]School of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China. [3]Biostatistics & Data Science, LSU Health Sciences Center New Orleans, Louisiana, United State of America. [4]School of Computer Science and Engineering, Kyungpook National University, Daegu, Republic of Korea. [5]Department of Mechanical Engineering, Pusan National University, Busan, Republic of Korea. *Author for correspondence. E-mail: aghjuee@kyungpook.kr

ABSTRACT. Human-computer interaction technologies have been used since the 1970s but have only gained growing popularity in recent years with new design paradigms. Ongoing research and development in gesture recognition systems with broad application prospects have focused on improving accuracy and real-time performance as well as the robustness of specific machine learning algorithms against environmental conditions. This paper addresses the accuracy enhancement of a novel Fifth Dimension Technologies data-glove-based gesture recognition system using a genetic-algorithm (GA)-trained *k*-means++-improved radial basis function (RBF) or GK-RBF neural network. First, we analyzed and modeled the sensor distribution in the data glove and proposed joint constraints based on the finger joint angle and sensor mapping. Then, we trained the model and conducted experimental verification to demonstrate the model's excellent real-time performance. Our results showed a training accuracy of 100%, a reduction in training error rate by 89.3%, and an accuracy rate improvement of at least 3.5% between the different static gestures, even with different operators. Specifically, the GK-RBF neural network outperforms the RBF and GA-modified models by 4.36 and 2.21 abs.%, respectively, in terms of recognition accuracy. The 99.85-% accuracy rate of 10-fold cross validation proves a high degree of compatibility with data-glove-based recognition systems.

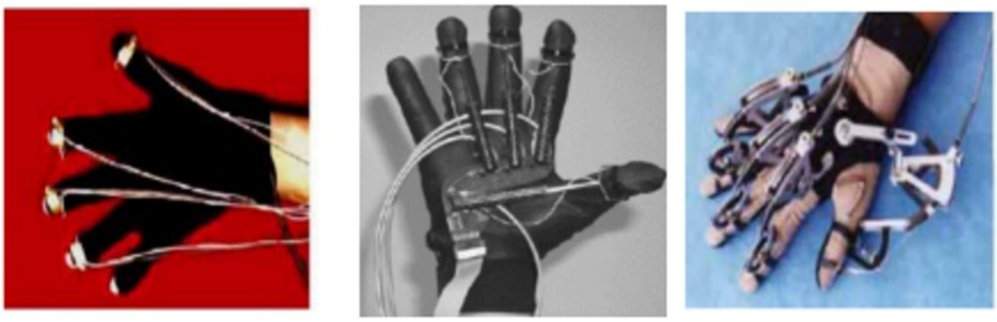
Keywords: Data classification; human-computer interaction; adaptive; training accuracy; intelligent sensors.

Received on January 4, 2024.
Accepted on March 15, 2024.

Introduction

Gesture recognition is a widely adopted technique in computer vision technology that enables the control of devices, systems, and machines through body gestures or movements. The technique finds applications in different fields including distance learning, virtual reality systems, natural user interfaces, automation systems, three-dimensional (3D) virtual games, intelligent robot operation, and smart homes. The image vision approach enables machines to interpret human hand gestures and convert them into commands or instructions (Alnuaim et al., 2022). By leveraging advanced algorithms such as convolutional neural networks (CNNs), different types of movements and specific hand postures can be recognized in real time, revolutionizing how humans interact naturally and conveniently with computers. However, the simultaneous performance of actions in the foreground or background, as well as preceding or subsequent actions, can adversely affect image vision and increase the complexity of data collection. The extent of related calculations increases, requiring high-performance equipment and introduces constraints related to the physical space within which the cameras are installed. To mitigate these challenges, we propose a novel gesture recognition system in this study based on data gloves.

Data gloves are wearable devices that gained popularity in the late 20th century; they consist of multiple sensor devices. Through these sensors, the location, posture, and rotation data of the user's hand and fingers can be extracted using computer systems for all degrees of freedom of the human hand. The bending of fingers can be detected with high precision, and tactile feedback can be provided to the user. Jaron Lanier, the founder of VPL Research, Inc., proposed the concept of human–computer interaction (HCI), and the technology has developed rapidly since (Siegert et al., 2014). During the same period, data gloves were proposed as input devices and introduced as a medium for virtual and real data transmission (Huang et al., 2013; Rautaray & Agrawal, 2015; Song, 2014). Thereafter, the research and development on data gloves gradually matured. Data gloves have since been applied in various industries to provide users with a better sense of HCI experience and immersion. The first wired data glove—called the Syre Glove—developed by Richard Sayre, Dan Sandin, and Tom Defanti in 1976 (Sturman & Zeltzer, 1994), was utilized for manipulating, controlling, and interacting with 3D computer-generated environments, marking the beginning of gesture recognition research in computer science. In the early 1980s, VPL Research Inc. patented (Zimmerman, 1985) a data glove based on light intensity modulation for bending measurements (using an optical flex sensor) to reduce the complex structure of data gloves and maintain high accuracy. The glove detected changes in light intensity through a curved channel for light transmission. Subsequently, force-feedback gloves were developed. Figure 1 shows various wired force-feedback gloves developed by the Chinese Academy of Sciences (CAS-Glove), Rutgers University (Rutgers Master II-ND), and Virtual Technologies (CyberGrasp).



(a) (b) (c)

Figure 1. Wired force feedback gloves: (a) CAS-Glove (Fu et al., 2004), (b) Rutgers Master II-ND (Bouzit et al., 2002), and (c) CyberGrasp (He et al., 2014).

Further, the next generation of pneumatic force-feedback gloves were developed, such as the Fluid Power Glove developed by Hosei University (Figure 2a) and 5DT Data Glove Ultra developed by Fifth Dimension Technologies (5DT) (Figure 2b). As the interest in data gloves for HCI is growing, gesture recognition technology has emerged as one of the core applied solutions. The purpose of this study is to realize human gesture recognition through algorithms. Gestures involve not only hand posture but also facial expressions or physical movements of other parts of the body. These gestures are distinctive, intuitive, and easy to learn. They have become an avenue for mutual interaction between computers and humans, providing a rich technical experience of HCI. CNNs are often used in this category. Neural networks have a multilayer network structure, and the sublayer of the networks contains multiple neurons, unlocking strong data analysis capabilities. With a simple and convenient structure, a neural network can learn nonlinear functions, classify linearly nonseparable data, and solve nonlinear equation systems. Hence, neural networks are widely used in intelligent control and robotics.

(a) (b)

Figure 2. Pneumatic force feedback gloves. (a) Fluid Power Glove (Wang et al., 2020) and (b) 5DT Data Glove Ultra (Arkenbout et al., 2015).

As reported in (Byun & Lee, 2019) a flexible skin tactile sensor-based gesture recognition system was proposed, achieving an extraordinarily high recognition accuracy rate of 97 - 99%. To boost the popularity of robotic assistants, a neural network-based LiDAR gesture recognition system was proposed in (Lindner et al., 2023). Their study reported a classification accuracy that depended on the utilized neural network architecture, providing an accuracy performance upwards of 92%, with the ResNet152 architecture reaching up to 96.1 %. Template matching (Mahbub et al., 2013), hidden Markov model (HMM), and support vector machine (Huu & Phung Ngoc, 2021; Saha et al., 2015) are also suitable for gesture recognition. HMM was developed in the 1970s as a statistical model and has since rapidly developed and seamlessly improved in the subsequent decade. By the 1980s, it became the core development framework for computer communication and signal processing. The model was first proposed by Jim Baker (Carnegie Mellon University) and Fred Jelinek (IBM Research) and was applied to a language recognition system to realize intelligent classification and labeling of computer languages. In (Kapuscinski et al., 2015) a time-of-flight (ToF) camera-based dynamic gesture recognition system was proposed based on the HMM and nearest neighbor technique (with dynamic time warming) classifiers, and its rotation-sensitive (−10° to 10°) dynamic recognition rate ranges between 75.5 to 86.1 %. The application of HMM has been compared with that of other classification methods (Pisharady & Saerbeck, 2015; Siddiqui et al., 2021; Zhang & Li, 2019).

Recently, the development of data gloves has brought about a new paradigm in gesture recognition research. Data gloves are widely used and not easily affected by the peripheral environment. With the high stability of the collected gesture data, meanings associated with different gestures can be more accurately identified. This ‘stability of data’ refers to the consistent and reliable recording of user hand movements and gestures, ensuring minimal variability in the collected information. This reliability is pivotal for accurate gesture recognition, as it signifies that the gloves consistently and accurately capture the intended gestures across diverse conditions. An instance of application is in sign language recognition. As discussed in (Tubaiz et al., 2015) Arabic sign language recognition was performed at a gesturing recognition rate of 98.9 % by combining a pair of DG5-V data gloves with RGB video camera-based information. However, data-glove-based gesture recognition technologies still have some inevitable drawbacks, such as those associated with the size and length of the user’s hand, and the flexibility and curvature of fingers. Therefore, when high-precision operation is required, the recognition accuracy of gestures is affected and the overall performance of the gesture recognition technology deteriorates. Moreover, real-time performance is a critical aspect of gesture recognition systems, and conventional methods may encounter latency issues. Delays in recognizing and processing gestures can impede the seamless interaction between users and applications, affecting the overall user experience. To tackle accuracy-related and real-time problems, we propose the implementation of a method based on a genetic algorithm (GA)-optimized radial basis function (RBF) neural network for 5DT data gloves. Our work aims to overcome the shortcomings of traditional data-glove-based systems and provide a robust solution that excels in accuracy and real-time performance.

Methods

In this study, a GA-based *k*-means++-improved RBF (GK-RBF) neural network was employed to classify curvature data from a data glove, with the aim of accurately identifying 13 distinct hand gestures. Each hand gesture contained curvature data from 5 fingers. Gesture data collection was performed with a total of 100 samples using the GloveManager software. 20 samples of data were collected from each of the 5 participants (wearing the data glove on the right hand) in a standard laboratory setting at 10 s per gesture. These curvature data were normalized, resulting in their values ranging between 0 and 1. The obtained dataset consisted of 6500 datapoints, of which 80% served as training data and 20% as testing data. For model evaluation, the mean squared error was used as the loss function. The experimental environment and parameters were as follows.

1. Software and versions: Anaconda: 4.7.12 software package including Python version 3.7.
2. Hardware specifications: CPU: Intel(R) Core (TM) i5-9400F CPU, @2.90 GHz, 16.0 GB RAM; graphics card: NVIDIA GeForce GTX1070 8 GB/MSI; Mainboard: MSI Z390M-S01(MS-7C24); RAM: Kingston DDR4 2400 MHz 16 GB; hard drive: Samsung SSD 750 EVO 250 GB; operating system: Windows 10 Education 64-bit.
3. Training parameters: Learning rate: 0.01; epochs: 100.
4. GA parameters: population size: 50; crossover probability: 0.8; mutation probability: 0.2; maximum iterations: 100

Gesture design

We used the Data Glove 5 Ultra (right hand) designed by 5DT and extracted features of each static hand gesture using the data glove. High-accuracy data collection was performed using a self-compensating elastic optical fiber sensor with a self-calibration function. Featuring 8-bit flexural resolution, extreme comfort for different palm shapes, low drift, and an open architecture, the elastic data glove provided additional advantages in HCI. Figure 3 shows the distribution of sensors in the glove (Arkenbout et al., 2015). The data collected by each sensor were normalized so that the output value was between 0 and 1.

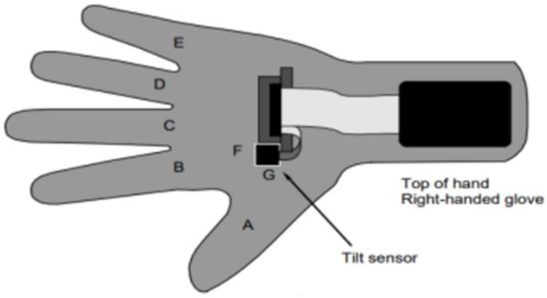


Figure 3. Sensor positions in the 5DT Data Glove 5 Ultra.

To clarify the specific sensor configuration (*c.f.* Figure 3), Table 1 lists the sensor details. The table guides in understanding the origin of the gesture data and related hardware setup, which plays a critical role in ensuring accuracy and consistency during experimental data collection and processing. A clear overview of which sensors remain active/inactive in the current study is given. For example, sensors F and G are no longer integrated in the data glove because they are obsolete.

Table 1. Sensors of the 5DT Data Glove Ultra.

Sensor	Description
A	Thumb flexure
B	Index finger flexure
C	Middle finger flexure
D	Ring finger flexure
E	Little finger flexure
F	Pitch angle of tilt sensor (obsolete)
G	Roll angle of tilt sensor (obsolete)

In the data glove, gesture recognition primarily relies on a preset template system according to the gesture definitions presented in Table 2. We assigned appropriate binary digits to different gesture types depending on the bending or stretching of each finger so that definable outcomes can be obtained. This provides a distinct template for each specific gesture, which can be seen as a baseline model. This system determines whether the fingers are bent based on set thresholds, thus recognizing gestures. In particular, the functions void SetThresholdAll(float[] aUpperVals, float[] aLowerVals) and void SetThreshold(int nSensor, float fUpperVal, float fLowerVal) offer the capability to set gesture recognition thresholds (upper and lower) for all sensors and some specific sensors, respectively (Arkenbout et al., 2015). Static hand gestures may act as special transition states in temporal hand gestures and provide means to identify and classify temporal hand gestures.

The threshold-based template approach encounters certain unmitigated challenges in practical applications. The physical size and form of the hand varies from one operator to another; hence, the same gesture might be differently interpreted because the curvature of the fingers may not be identical. To ensure accurate gesture recognition for each user, there is a need for tedious threshold adjustments for the individual operator. To address these efficiency constraints and enhance the stability and accuracy of real-time gesture recognition performance, we used the GK-RBF neural network model for data processing and gesture recognition. This model optimizes and learns from various collected gesture data used as input and training sets. This approach offers users a more accurate and adaptive gesture recognition experience, eliminating the frequent need for manual threshold adjustments. As shown in Table 3, the real-world data matrix consists of 13 rows and 5 columns, representing 13 gestures, and each gesture comprises 5 datapoints (from 5 sensors).

Table 2. Static gestures and corresponding binary sequences.

Gesture Morphology	Gesture Sequence	Gesture Morphology	Gesture Sequence	Gesture Morphology	Gesture Sequence
	10000		10001		10110
	11000		11001		11101
	11100		10101		10011
	11110		11011		10111
	11111				

Table 3. Representative gesture input data description.

Thumb (A)	Index finger (B)	Middle finger(C)	Ring finger (D)	Little finger (E)
0.1542	0.8878	0.8689	0.7886	0.7031
0.2467	0.1708	0.7909	0.8728	0.5588
0.1243	0.1126	0.1613	0.8907	0.8907
0.2102	0.7909	0.1896	0.2361	0.5030
0.1263	0.2103	0.1596	0.1132	0.552
0.1106	0.8149	0.8774	0.9108	0.2136
0.1356	0.1732	0.8812	0.9184	0.2312
0.1625	0.7145	0.2171	0.8637	0.3125
0.1040	0.1466	0.1088	0.7343	0.2268
0.2189	0.8621	0.6748	0.2080	0.1247
0.2751	0.2169	0.5274	0.2359	0.3167
0.1123	0.8198	0.1615	0.1028	0.1770
0.1352	0.1385	0.1456	0.1264	0.1336

K-means++ algorithm

The *k*-means clustering algorithm is an unsupervised machine learning algorithm proposed in (MacQueen, 1967) and is mainly used to classify and analyze the data. The basic principle for selecting initial cluster centers is that the mutual distance between the initial cluster centers should be as large as possible. The selection sequence is as follows:

1. Randomly select a point from the dataset (X) as the first cluster center.
2. Calculate the shortest distance $D(x)$ between each sample and the existing cluster center, that is, the distance to the nearest cluster center (select the sample with the largest distance as the new cluster center with some probability). The larger the distance, the greater the probability of being selected as the cluster center. Then, calculate the probability of each sample being selected as the next cluster center using the roulette method following:

$$P(x) = \frac{D(x)^2}{\sum_{x \in X} D(x)^2} \tag{1}$$

3. Repeat process 2 until k cluster centers are identified.
4. Assign each point in the sample set to a cluster: calculate the distance between each point and the centroid (commonly used Euclidean distance and cosine distance) and assign it to the cluster corresponding to the nearest centroid.
5. Update the centroid of the cluster: the centroid of each cluster is updated as the average of all points in the cluster.
6. Repeat steps 4 and 5. If the distance between the updated and previous centroid is less than a certain threshold, the clustering algorithm can be considered to have achieved the desired result and the iteration is terminated. Otherwise, continue iteration.

Table 4 enumerates the working principle, strengths, and weaknesses of k-means compared to new variants. While each new version of k-means offers specific benefits, not all align well with the demands of data-glove-based gesture recognition. For example, mini-batch k-means, while efficient for large datasets, can compromise the precision needed for detailed gesture data due to its approximate centroid update steps. Kernel k-means excels with complex, nonlinear separations but may introduce unnecessary computational complexity to our relatively structured gesture data. Meanwhile, spherical k-means is adept for spherical clusters, yet our gesture data does not inherently conform to such geometrical constraints. In contrast, k-means++ is distinguished by its thoughtful initial centroid placement, which is critical in capturing the subtle nuances between different gestures. This approach to initializing clusters significantly curtails the risk of miscalculation inherent to random centroid assignment, a potential issue with standard k-means that could lead to inconsistent gesture recognition. Ultimately, k-means++ offers a balance suited to the specificity of gesture recognition tasks where the clarity of each cluster formed is pivotal.

Table 4. Comparison of k-means and its newer variants.

Algorithm	Working principle	Advantages	Disadvantages	Application Scenarios
K-means	Random initial cluster centers, iterative updates	Simple and intuitive	Sensitive to initial centers, may converge to local optima	Broad clustering applications
Mini-batch k-means (Hicks et al., 2021)	Batch data updates cluster centers	Efficient, suitable for large datasets	May sacrifice quality for speed	Large datasets/Online clustering
K-means++	Improved method for selecting initial cluster centers	Fast convergence, accurate	Slower initialization	Broad clustering applications
Kernel k-means (Wang et al., 2019)	Maps data to high-dimensional space using kernel technique	Handles complex distributions	High computational cost, challenging kernel selection	Clustering complex distributions
Spherical k-means (Kim et al., 2020)	Assumes clusters are spherical, suitable for high-dimensional data	Suitable for clustering spherical clusters	Not suitable for non-spherical clusters	Text data and high-dimensional space clustering
K-means with constraints (Huang et al., 2021)	Introduces additional constraints, such as similarities or dissimilarities between samples	Can produce clusters meeting specific requirements	Introducing constraints may complicate the problem	Clustering tasks with prior knowledge or specific constraints
Fuzzy k-means (Zhao et al., 2023)	Assigns data points to one or more clusters based on membership degrees	Allows soft clustering, data points can belong to multiple clusters with varying degrees	More computationally intensive than standard k-means, sensitive to the choice of fuzziness parameter	Situations where data naturally belongs to overlapping categories or when the boundaries between clusters are not well-defined
Weighted k-means (Yang et al., 2015)	Modifies the k-means algorithm by introducing weights for each data point	Accommodates the importance or reliability of data points, more robust to outliers	Determining the appropriate weights can be challenging, potentially more complex to implement	Clustering tasks where some data points are more important or reliable than others, or when dealing with heterogeneous data

Genetic algorithm

GA is a metaheuristic that is widely used in the optimization of neural networks (Domashova et al., 2021; Feng et al., 2021; Zhang & Qu, 2021) and fuzzy control algorithms (Dang et al., 2020; Huang et al., 2015). Notable features of GA include the following: 1) direct operation on structural objects without restrictions on derivation or function continuity; 2) inherent implicit parallelism that offers more robust global optimization capabilities; and 3) involves probabilistic optimization strategy through which it can autonomously navigate and guide the optimized search space and adaptively modify the search direction rather than adhering to rigid rules. The global characteristics of GA are utilized to train the neural networks; that is, the method of gamblers replaces the evolutionary principle of "survival of the fittest." Individuals are screened for data processing based on biological principles such as genetics, crossover, and mutation. In comparison to the outcomes of genetic screening, individuals demonstrating higher fitness values are preserved. These individuals, identified through the screening process, contribute to the ongoing reorganization and screening of the population in the GA. Ultimately, the optimal solution is derived as the new neural network weight. This training cycle continues until the desired accuracy is achieved. The specific operating steps are shown in Figure 4.

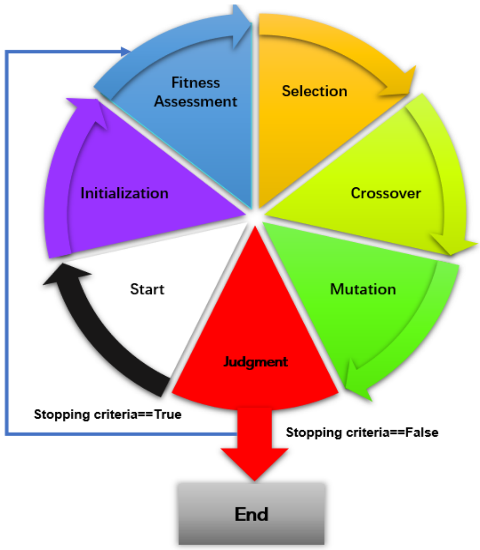


Figure 4. GA flow chart.

GK-RBF neural network

The RBF (Bawazeer et al., 2019; Soltani & El-Hag, 2021; Yang et al., 2022) provides a new system structure for neural network optimization learning with multilayer topology. For gesture recognition in HCI systems, the RBF network achieves universal approximation and exhibits faster learning speed compared to traditional neural networks with various gesture data. Training is completed after a targeted number of iterations or when the calculated error reaches a target value. Due to the simple representation, radial symmetry, and the advantageous property of being differentiable of any order, the hidden layer selects the Gaussian function as the transfer function, and the activation function is expressed by Equation 2 (Chen & Bakshi, 2009):

$$R(x) = \exp\frac{(x-c_i)^2}{-2b^2} \tag{2}$$

where X is the input vector ($X = (x_1, x_2, \ldots, x_i)$), C_i is a reference vector (core or prototype), and b is the dimension of the influence field. When b is smaller, the basis function is more optimal. It can be seen from (1) that C_i and b are crucial parameters of the RBF neural network, which are usually determined by the nearest neighbor clustering method. However, the method may cause the RBF neural network to overfit the data. Thus, the selection of the prototype is very important. Given that reference vector selection has great influence on clustering results, we applied the improved *k*-means algorithm (*k*-means++) to select the prototype. The output of the RBF neural network $y(x)$ is usually determined by Equation (3) (Broomhead & Lowe, 1988):

$$y(x) = \sum_{1}^{m} w_j \varphi(\|x - c_j\|) \tag{3}$$

where w_j are the weights associated with each RBF center c_j, m is the total number of centers, f is the radial basis function used, and ?$x - c_j$? is the Euclidean distance between input x and center c_j.

Integrating GA, *k*-means++, and RBF into the GK-RBF neural network (Figure 5) is not an arbitrary strategy but rooted in addressing the limitations of the RBF neural network. The RBF neural network with nonlinear recognition capabilities, while powerful, often struggles in selecting the right centers and weights. *k*-means++ aids in better center selection through efficient clustering, which eliminates the randomness and potential inefficiency of traditional methods. Meanwhile, GA displays robust optimization capability owing to its global search advantages, ensuring that the weights of the network are optimal. Together, these methods create a network that is efficient in learning and shows effective performance. The basic steps are described as follows (see Figure 5):
Step 1. Initialization: Use the *k*-means++ clustering algorithm to select optimal initial cluster center for the RBF neural network.
Step 2. Weight optimization using GA: Use GA to optimize the weights of the RBF neural network. The

robustness of GA in exploring a broad solution space renders it an ideal choice for weight optimization. Weights play a pivotal role in dictating the performance of a network, and conventional methods might easily get stuck in local minima. GA furnishes a global search strategy, enhancing the probability of pinpointing optimal or near-optimal weights.

Step 3. Training: With the optimized centers and weights in place, train the RBF neural network using standard learning algorithms. The synergistic interplay of *k*-means++ for center initialization and GA for weight optimization lays a robust foundation for the training process, facilitating faster convergence and superior generalization.

Step 4. Evaluation: Once the network is trained, evaluate its performance on validation or test datasets.

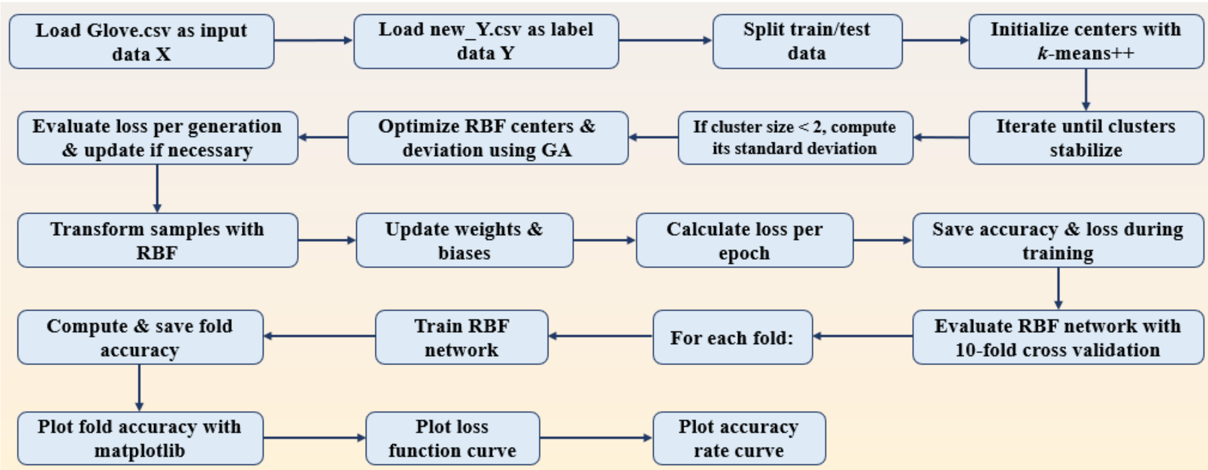


Figure 5. Training process of the GK-RBF neural network.

Results and discussion

Figure 6 compares the clustering visualization on the dataset for the different *k*-means versions. The t-distributed Stochastic Neighbor Embedding (t-SNE) technique (Zhou et al., 2018) was utilized to capture the local structures of complex multidimensional data and display them in a low-dimensional space. The visual differences among any given *k*-means visualization results are not significant, especially when visualizing high-dimensional data reduced to two or three dimensions. The dimensionality reduction inevitably results in some loss of information, leading to similar visual outcomes for different clustering algorithms, particularly when the underlying data structure is relatively simple, or the clustering boundaries are clear. The Calinski-Harabasz (CH) Index (Ekemeyong Awong & Zielinska, 2023) was utilized to assess the different clustering methods. The higher the CH Index value, the better the clustering effect, indicating tighter grouping within clusters and more dispersion between different clusters. *K*-means++ scored the highest on our dataset, proving it to be the most suitable approach. Figures 7a and 7b depict the accuracy and loss curves, respectively, obtained by training the RBF neural network, GA-RBF neural network (Jia et al., 2014), and GK-RBF neural network. The final gesture output data description for the optimized GK-RBF neural network is shown in Table 5. After testing with an identical amount of gesture input data, the GK-RBF network exhibited the highest training accuracy (100%) and test accuracy (98.82%), along with the lowest training error rate (0.0038%), as summarized in Table 6 from the ablation studies. A comparison of the ideal and actual output shows that the optimized network has better approximation ability.

The test accuracy of 98.82 % is a notable result in the field of gesture recognition, especially considering the complexity involved. In comparison, in (Chevtchenko et al., 2018) a state-of-the-art accuracy rate of up to 97.63% was obtained using a multiobjective GA for static hand gesture recognition. Further, in (Smith et al., 2021) an accuracy rate of up to 93 and 95% was reported for range and range angle-profiles, respectively, using a CNN and frequency-modulated-continuous-wave millimeter-wave radars. Separately, in (Cruz et al., 2023) an accuracy rate up to 97.45±1.02 and 88.05±3.10% was demonstrated for classification of signal data and gesture recognition, respectively, using a Deep Q-network-based learning algorithm. Other accuracy results reported in the literature for static hand gesture recognition are mostly lower. Based on these comparisons, our application of the GK-RBF model has clearly distinguished itself in terms of accuracy. However, the standard for “sufficiently high” accuracy varies from one application to another. When compared with other related research findings, for critical applications such as medical surgeries or high-security applications, the adequacy 6of this accuracy remains to be further evaluated. For certain routine applications or less complicated remote robot-control applications, this accuracy may meet the targeted requirements. It should also be recognized that the accuracy assessment of this data glove is closely related to its usage conditions and environment. The accuracy may vary under different experimental conditions and application contexts. Further research may explore how to maintain or improve this accuracy in different application backgrounds.

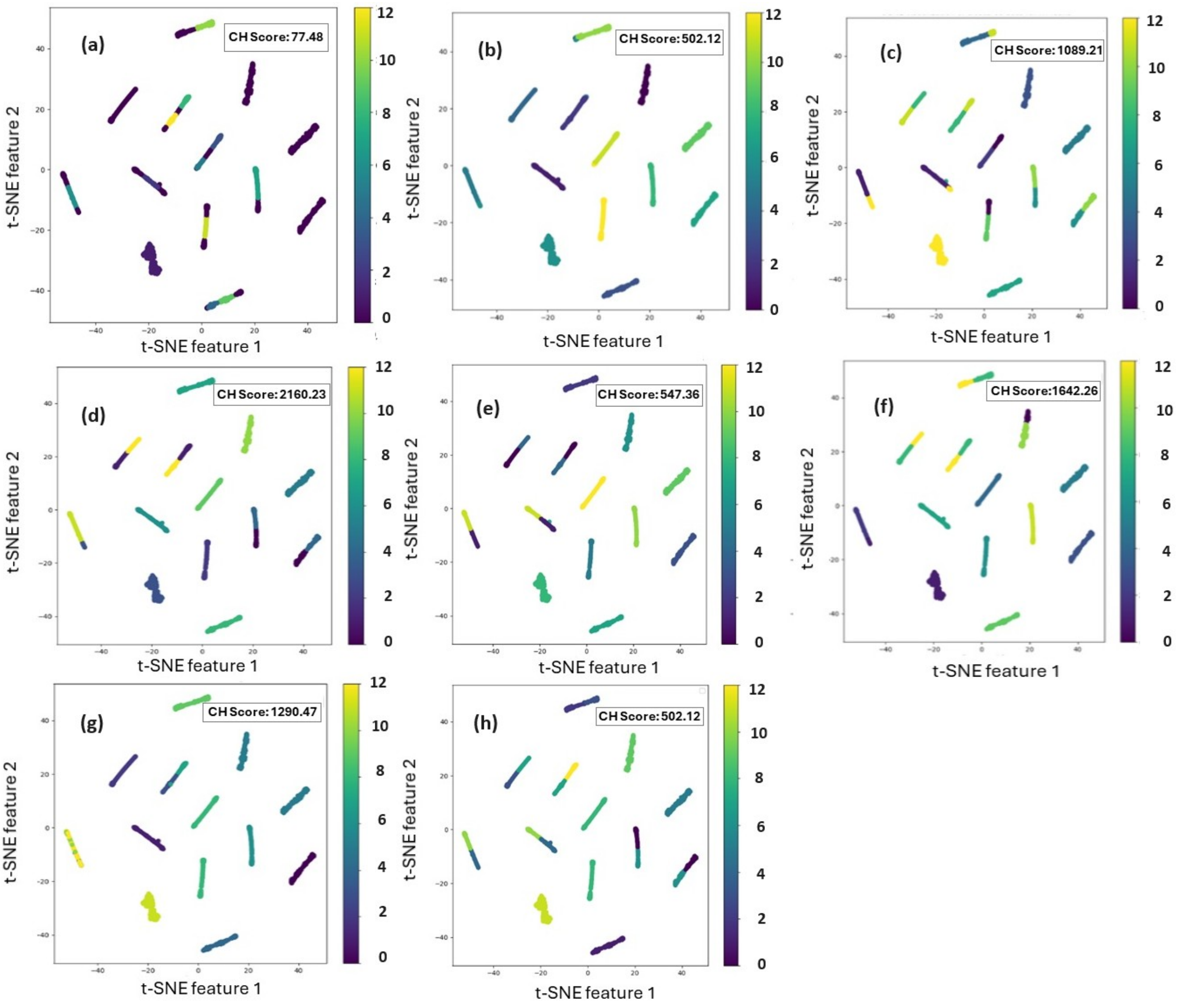


Figure 6. Clustering visualizations for (a) kernel *k*-means, (b) fuzzy *k*-means, (c) *k*-means, (d) *k*-means++, (e) *k*-means with constraints, (f) mini batch *k*-means, (g) spherical *k*-means, and (h) weighted *k*-means. The CH Index values are indicated.

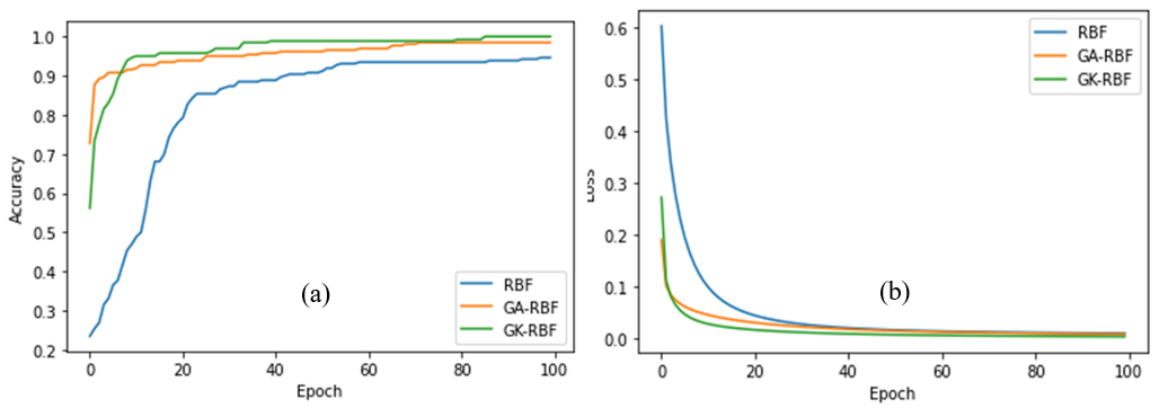


Figure 7. Performance of the RBF, GA-RBF, and GK-RBF neural networks. (a) Accuracy rate and (b) loss function curve. The neural

network hyperparameters were as follows: batch size = 6,500 (batch size = 1,300 per sensor, training samples: 80 %, test samples 20 %), learning rate=0.01, and epoch=100.

Table 5. Gesture data collected from the sensors.

Gesture template	Thumb (A)	Index finger (B)	Middle finger (C)	Ring finger (D)	Little finger (E)
1 0 0 0 0	0.0359	0.9879	0.9987	0.9886	0.9763
1 1 0 0 0	0.0334	0.0447	0.9812	0.9961	0.8981
1 1 1 0 0	0.0396	0.0621	0.0546	0.9369	0.9201
1 0 1 1 0	0.0236	0.8708	0.0324	0.0417	0.7925
1 1 1 1 0	0.0139	0.0268	0.0164	0.0175	0.9316
1 0 0 0 1	0.0087	0.8986	0.8722	0.9521	0.0157
1 1 0 0 1	0.0093	0.0770	0.9319	0.9536	0.0151
1 0 1 0 1	0.0757	0.8269	0.0658	0.8978	0.0571
1 1 1 0 1	0.0743	0.0516	0.0616	0.9522	0.0725
1 0 0 1 1	0.0335	0.8949	0.8748	0.0583	0.0708
1 1 0 1 1	0.0575	0.0216	0.8274	0.0335	0.0316
1 0 1 1 1	0.0114	0.9127	0.0249	0.2751	0.0213
1 1 1 1 1	0.0091	0.0087	0.0103	0.0116	0.0113

Table 6. Performance comparison of the RBF, GA-RBF, and GK-RBF neural networks.

Neural networks algorithm / Performance metric	Traditional RBF	GA-RBF	GK-RBF
Training Accuracy (%)	95.3	98.2	100
Training Error Rate (%)	0.012	0.0085	0.0038
Test Accuracy (%)	94.46	96.61	98.82

To determine performance reliability, we performed repeated *k*-fold cross-validation on our datasets, with *k* = 10, which is a frequently chosen value to balance computational cost and model evaluation accuracy. Cross-validation assists in determining how the outcomes of a statistical analysis will generalize to an independent dataset, bolstering the robustness of our model comparisons. In this evaluation, we observed that the three models—GK-RBF neural network, GA-RBF neural network, and RBF neural network—demonstrated almost similar accuracies (see Figure 8). In particular, the GK-RBF model achieved an accuracy of 99.85%, while the GA-RBF and RBF models achieved accuracy of 99.69 and 99.38%, respectively. Although there is only a subtle difference in the accuracy among the models, such a minor variation highlights the consistency and effectiveness of our experimental method. This marginal discrepancy among the three models might be because they all exhibit superior performance when processing the current dataset, which might not introduce any extreme or unusual fluctuations leading to significant performance differences.

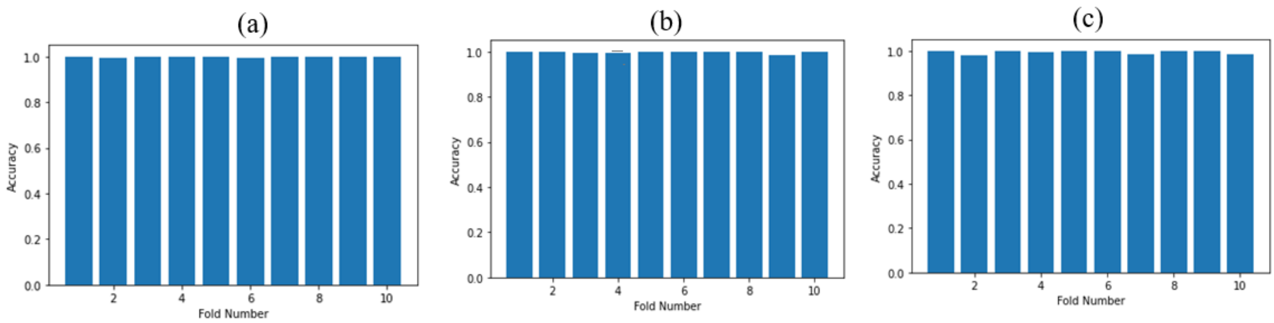


Figure 8. *k*-fold cross-validation results. Comparison of 10-fold cross-validation results for the (a) GK-RBF, (b) GA-RBF, and (c) RBF neural networks.

While *k*-means++ was employed as the clustering algorithm herein, it is still possible to use different clustering algorithms combined with GA to test the training performance. In fact, multiple clustering algorithms may also be used in an integrated approach; for example, *k*-means++ combined with density-based spatial clustering of applications with noise (DBSCAN). DBSCAN has several advantages over other clustering algorithms (Zhang et al., 2022), such as the ability to handle nonlinearly separable clusters, detect clusters of different shapes and sizes, and effectively handle noise and outliers. For further enhancement of the training performance and test accuracy, prospective research directions can be dichotomized into parameter fine-tuning and development of scalability and interpretability. The former involves fine-tuning the values of certain parameters within the system, and the latter involves exploring ways to render the system more scalable to handle larger and more complex datasets and be more interpretable, which means it can be better understood and analyzed by humans. This research direction is considered to have more depth and complexity and may require more advanced machine learning and deep learning techniques.

We have so far presented the use of the GK-RBF neural network algorithm for effective data-based gesture recognition; we will further discuss the general limitations of using RBF networks. Compared with deep learning networks, RBF networks have limited scalability when dealing with large datasets. When using RBF techniques for pattern recognition or data classification, increasing the number of RBF centers with a growing amount of data can improve the accuracy of the model. However, at a certain point, increasing the number of centers leads to a decrease in training speed and accuracy because processing many centers can become computationally expensive and hence slow down the training process. Therefore, it is important to strike a balance between the number of RBF centers and amount of training data to ensure efficient and accurate training. Moreover, RBF networks present shortcomings when handling complex tasks and adapting to changes in data distribution. The network uses fixed centroids, which means that it cannot adjust to changes in the data distribution like deep learning networks can through weight updates. Furthermore, the RBF network is a single-hidden-layer network structure, which may not be suitable for handling complex tasks such as image classification or natural language processing that often require deep learning neural network architectures with multiple hidden layers to extract complex features from the data. Nevertheless, in this study, we have shown how the GK-RBF neural network is particularly effective at optimizing curvature data in a data-glove-based recognition system, with improved training speed, reduced training time, good real-time performance, and high accuracy.

Conclusion

In the field of gesture recognition, particularly concerning data gloves, some traditional recognition methods, such as those based on template matching, sensor thresholds, and features, often encounter limitations. These methods may not achieve sufficient accuracy in certain complex application scenarios. In this study, we adopted the GK-RBF neural network algorithm, with local convergence and global search capabilities, overcoming the shortcomings of the combined k-means++ and RBF neural network algorithms and improving the training quality of the RBF neural network. Through detailed experimentation and validation, it was observed that the algorithm reduced the complexity of data processing in the gesture recognition process, effectively improving the training speed, reducing the training time, and ensuring state-of-the-art real-time performance and high accuracy of the data-glove-based gesture recognition system. The GK-RBF neural network achieved an accuracy of 100% on the training set, while reducing the training loss rate to 0.0038%. In real-world testing, the recognition accuracy reached as high as 98.82%, significantly surpassing those of previously reported methods solely based on RBF or GA-RBF neural networks. In 10-fold cross-validation, the accuracy reached 99.85%, proving that our adopted method showed a high level of compatibility for data-glove-based gesture recognition.

Considering the frequent changes of human operators in actual situations, we anticipate the prospective collection of more gesture data based on different operators, which will enable more accurate recognition results. Although the proposed algorithm for gesture recognition has certain advantages for specific tasks, such as optimizing curvature data from a data glove, it may have limitations when compared with deep learning. Notably, the RBF network structure used in our approach demonstrated inherently fast operation and efficient training performance. Because of the simple model structure, the training time was generally shorter, especially with smaller datasets. Further, better interpretability was achieved, and the model was easy to understand in lieu of more complex deep learning networks. Leveraging GA optimization allowed for global search for the optimal set of weights and parameter configurations in the solution space for the RBF network, resulting in superior performance. This optimization approach is also less likely to become trapped in local optima, which is a pressing concern in traditional optimization algorithms, particularly in deep learning networks. Furthermore, GA optimization can handle optimization problems involving multiple objectives or constraints, making it suitable for more complex and nuanced optimization tasks. While deep learning methods excel with large datasets and complex tasks, our proposed method, which includes GA optimization, remains highly competitive for simple and constrained tasks such as data-glove-based gesture recognition. In future explorations, under diverse application backgrounds and conditions, further validation and refinement of the algorithm will be essential. We anticipate that this research will provide new insights for continuous progress in the field of HCI, particularly gesture recognition.

Acknowlegements

Liangxiao Ding and Kuan-Way (Guanghui) Chee contributed equally to this work and should be considered co–first authors.

References

Alnuaim, A., Zakariah, M., Hatamleh, W. A., Tarazi, H., Tripathi, V., & Amoatey, E. T. (2022). Human-Computer Interaction with Hand Gesture Recognition Using ResNet and MobileNet. *Computational Intelligence and Neuroscience*, 8777355. https://doi.org/10.1155/2022/8777355

Arkenbout, E. A., De Winter, J. C. F., & Breedveld, P. (2015). Robust Hand Motion Tracking through Data Fusion of 5DT Data Glove and Nimble VR Kinect Camera Measurements. *Sensors*, *15*(12), 31644-31671. https://www.mdpi.com/1424-8220/15/12/29868

Bawazeer, S. A., Baakeem, S. S., & Mohamad, A. (2019). A New Radial Basis Function Approach Based on Hermite Expansion with Respect to the Shape Parameter. *Mathematics*, *7*(10), 979. https://www.mdpi.com/2227-7390/7/10/979

Bouzit, M., Burdea, G., Popescu, G., & Boian, R. (2002). The Rutgers Master II-new design force-feedback glove. *IEEE/ASME Transactions on Mechatronics*, *7*(2), 256-263. https://doi.org/10.1109/TMECH.2002.1011262

Broomhead, D. S., & Lowe, D. (1988). Multivariable Functional Interpolation and Adaptive Networks. *Complex Systems*, *2*.

Byun, S.-W., & Lee, S.-P. (2019). Implementation of Hand Gesture Recognition Device Applicable to Smart Watch Based on Flexible Epidermal Tactile Sensor Array. *Micromachines*, *10*(10), 692. https://www.mdpi.com/2072-666X/10/10/692

Chen, H., & Bakshi, B. R. (2009). 3.12 - Linear Approaches for Nonlinear Modeling. In S. D. Brown, R. Tauler, & B. Walczak (Eds.), *Comprehensive Chemometrics* (pp. 453-462). Elsevier. https://doi.org/https://doi.org/10.1016/B978-044452701-1.00060-0

Chevtchenko, S. F., Vale, R. F., & Macario, V. (2018). Multi-objective optimization for hand posture recognition. *Expert Systems with Applications*, *92*, 170-181. https://doi.org/https://doi.org/10.1016/j.eswa.2017.09.046

Cruz, P. J., Vásconez, J. P., Romero, R., Chico, A., Benalcázar, M. E., Álvarez, R., Barona López, L. I., & Valdivieso Caraguay, Á. L. (2023). A Deep Q-Network based hand gesture recognition system for control of robotic platforms. *Scientific Reports*, *13*(1), 7956. https://doi.org/10.1038/s41598-023-34540-x

Dang, X. K., Do, V. D., & Nguyen, X. P. (2020). Robust Adaptive Fuzzy Control Using Genetic Algorithm for Dynamic Positioning System. *IEEE Access*, *8*, 222077-222092. https://doi.org/10.1109/ACCESS.2020.3043453

Domashova, J. V., Emtseva, S. S., Fail, V. S., & Gridin, A. S. (2021). Selecting an optimal architecture of neural network using genetic algorithm. *Procedia Computer Science*, *190*, 263-273. https://doi.org/https://doi.org/10.1016/j.procs.2021.06.036

Ekemeyong Awong, L. E., & Zielinska, T. (2023). Comparative Analysis of the Clustering Quality in Self-Organizing Maps for Human Posture Classification. *Sensors*, *23*(18), 7925. https://www.mdpi.com/1424-8220/23/18/7925

Feng, X., Zhao, J., & Kita, E. (2021). Genetic Algorithm-based Optimization of Deep Neural Network Ensemble. *The Review of Socionetwork Strategies*, *15*(1), 27-47. https://doi.org/10.1007/s12626-021-00074-9

Fu, Y., Yuan, K., Zhu, H., & Du, Q. (2004). Motion Modeling and Software System Design for CAS-Glove Data Glove. *Journal of System Simulation*, *16*(4), 660-662.

He, B., Wang, Y., Song, W., & Tang, W. (2014). Design resource management for virtual prototyping in product collaborative design. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, *229*(12), 2284-2300. https://doi.org/10.1177/0954405414551106

Hicks, S. C., Liu, R., Ni, Y., Purdom, E., & Risso, D. (2021). mbkmeans: Fast clustering for single cell data using mini-batch k-means. *PLOS Computational Biology*, *17*(1), e1008625. https://doi.org/10.1371/journal.pcbi.1008625

Huang, P., Yao, P., Hao, Z., Peng, H., & Guo, L. (2021). Improved Constrained k-Means Algorithm for Clustering with Domain Knowledge. *Mathematics*, *9*(19), 2390. https://www.mdpi.com/2227-7390/9/19/2390

Huang, S. C., Jiau, M. K., & Lin, C. H. (2015). Optimization of the Carpool Service Problem via a Fuzzy-Controlled Genetic Algorithm. *IEEE Transactions on Fuzzy Systems*, *23*(5), 1698-1712. https://doi.org/10.1109/TFUZZ.2014.2374194

Huang, Z., Peng, B., & Wu, J. (2013). Research and application of human-computer interaction system based on gesture recognition technology. In D. Li, & Y. Chen (Eds.), *Computer and Computing Technologies in Agriculture* (IFIP Advances in Information and Communication Technology, Vol. 392). Springer. https://doi.org/10.1007/978-3-642-36124-1_26

Huu, P. N., & Phung Ngoc, T. (2021). Hand Gesture Recognition Algorithm Using SVM and HOG Model for Control of Robotic System. *Journal of Robotics*, *2021*, 3986497. https://doi.org/10.1155/2021/3986497

Jia, W., Zhao, D., Shen, T., Su, C., Hu, C., & Zhao, Y. (2014). A New Optimized GA-RBF Neural Network Algorithm. *Computational Intelligence and Neuroscience*, 982045. https://doi.org/10.1155/2014/982045

Kapuscinski, T., Oszust, M., Wysocki, M., & Warchol, D. (2015). Recognition of hand gestures observed by depth cameras. *International Journal of Advanced Robotic Systems*, *12*(4), 36. https://doi.org/10.5772/60091

Kim, H., Kim, H. K., & Cho, S. (2020). Improving spherical k-means for document clustering: Fast initialization, sparse centroid projection, and efficient cluster labeling. *Expert Systems with Applications*, *150*, 113288. https://doi.org/10.1016/j.eswa.2020.113288

Lindner, T., Wyrwał, D., & Milecki, A. (2023). An Autonomous Humanoid Robot Designed to Assist a Human with a Gesture Recognition System. *Electronics*, *12*(12), 2652. https://www.mdpi.com/2079-9292/12/12/2652

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability* (Vol. 1, pp. 281-297). University of California Press. http://projecteuclid.org/euclid.bsmsp/1200512992

Mahbub, U., Imtiaz, H., Roy, T., Rahman, M. S., & Rahman Ahad, M. A. (2013). A template matching approach of one-shot-learning gesture recognition. *Pattern Recognition Letters*, *34*(15), 1780-1788. https://doi.org/10.1016/j.patrec.2012.09.014

Pisharady, P. K., & Saerbeck, M. (2015). Recent methods and databases in vision-based hand gesture recognition: A review. *Computer Vision and Image Understanding*, *141*, 152-165. https://doi.org/10.1016/j.cviu.2015.08.004

Rautaray, S. S., & Agrawal, A. (2015). Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*, *43*(1), 1-54. https://doi.org/10.1007/s10462-012-9356-9

Saha, S., Konar, A., & Roy, J. (2015). Single Person Hand Gesture Recognition Using Support Vector Machine. In K. Maharatna, G. K. Dalapati, P. K. Banerjee, A. K. Mallick, & M. Mukherjee, *Computational Advancement in Communication Circuits and Systems* New Delhi.

Siddiqui, U. A., Ullah, F., Iqbal, A., Khan, A., Ullah, R., Paracha, S., Shahzad, H., & Kwak, K.-S. (2021). Wearable-sensors-based platform for gesture recognition of autism spectrum disorder children using machine learning algorithms. *Sensors*, *21*(10), 3319. https://www.mdpi.com/1424-8220/21/10/3319

Siegert, I., Haase, M., Prylipko, D., & Wendemuth, A. (2014). *Discourse particles and user characteristics in naturalistic human-computer interaction.* Human-Computer Interaction; Advanced Interaction Modalities and Techniques, Cham.

Smith, J. W., Thiagarajan, S., Willis, R., Makris, Y., & Torlak, M. (2021). Improved Static Hand Gesture Classification on Deep Convolutional Neural Networks Using Novel Sterile Training Technique. *IEEE Access*, *9*, 10893-10902. https://doi.org/10.1109/ACCESS.2021.3051454

Soltani, A. A., & El-Hag, A. (2021). A new radial basis function neural network-based method for denoising of partial discharge signals. *Measurement*, *172*, 108970. https://doi.org/10.1016/j.measurement.2021.108970

Song, C. (2014). Android-based remote-control with real-time video surveillance for wi-fi robot. In Y. Yuan, X. Wu, & Y. Lu, *Trustworthy computing and services* (Communications in Computer and Information Science, Vol. 426). Springer. https://doi.org/10.1007/978-3-662-43908-1_47

Sturman, D. J., & Zeltzer, D. (1994). A survey of glove-based input. *IEEE Computer Graphics and Applications*, *14*(1), 30-39. https://doi.org/10.1109/38.250916

Tubaiz, N., Shanableh, T., & Assaleh, K. (2015). Glove-Based Continuous Arabic Sign Language Recognition in User-Dependent Mode. *IEEE Transactions on Human-Machine Systems*, *45*(4), 526-533. https://doi.org/10.1109/THMS.2015.2406692

Wang, D., Ohnishi, K., & Xu, W. (2020). Multimodal Haptic Display for Virtual Reality: A Survey. *IEEE Transactions on Industrial Electronics*, *67*(1), 610-623. https://doi.org/10.1109/TIE.2019.2920602

Wang, S., Gittens, A., & Mahoney, M. W. (2019). Scalable kernel k-means clustering with Nyström approximation: relative-error bounds. *Journal of Machine Learning Research*, *20*(1), 431-479.

Yang, X., Xi, W., Sun, Y., Zeng, T., Long, T., & Sarkar, T. K. (2015). Optimization of Subarray Partition for Large Planar Phased Array Radar Based on Weighted K-Means Clustering Method. *IEEE Journal of Selected Topics in Signal Processing*, *9*(8), 1460-1468. https://doi.org/10.1109/JSTSP.2015.2465306

Yang, Y., Wang, P., & Gao, X. (2022). A Novel Radial Basis Function Neural Network with High Generalization Performance for Nonlinear Process Modelling. *Processes*, *10*(1), 140. https://www.mdpi.com/2227-9717/10/1/140

Zhang, J., & Qu, S. (2021). Optimization of Backpropagation Neural Network under the Adaptive Genetic Algorithm. *Complexity*, 1718234. https://doi.org/10.1155/2021/1718234

Zhang, X., & Li, X. (2019). Dynamic Gesture Recognition Based on MEMP Network. *Future Internet*, *11*(4), 91. https://www.mdpi.com/1999-5903/11/4/91

Zhang, X., Shen, X., & Ouyang, T. (2022). Extension of DBSCAN in Online Clustering: An Approach Based on Three-Layer Granular Models. *Applied Sciences*, *12*(19), 9402. https://www.mdpi.com/2076-3417/12/19/9402

Zhao, P., Zhang, Y., Ma, Y., Zhao, X., & Fan, X. (2023). Discriminatively embedded fuzzy K-Means clustering with feature selection strategy. *Applied Intelligence*, *53*(16), 18959-18970. https://doi.org/10.1007/s10489-022-04376-5

Zhou, H., Wang, F., & Tao, P. (2018). t-Distributed Stochastic Neighbor Embedding Method with the Least Information Loss for Macromolecular Simulations. *Journal of Chemical Theory and Computation*, *14*(11), 5499-5510. https://doi.org/10.1021/acs.jctc.8b00652

Zimmerman, T. G. (1985). *Optical flex sensor* (Patent Number: 4,542,291). https://patentimages.storage.googleapis.com/50/2c/96/c80d0bd9571d92/US4542291.pdf