



An Embedded Neural Network Approach for Reinforcing Deep Learning: Advancing Hand Gesture Recognition

Anwar Mira

(University of Babylon, College of Information Technology, Iraq
 <https://orcid.org/0009-0000-8417-7257>, anwar.jaafar@uobabylon.edu.iq)

Olaf Hellwich

(Technische Universität Berlin, Computer Vision & Remote Sensing, Germany
 <https://orcid.org/0000-0002-2871-9266>, olaf.hellwich@tu-berlin.de)

Abstract: Deep neural networks (DNNs) can face limitations during training for recognition, motivating this study to improve recognition capabilities by optimizing deep learning features for hand gesture image recognition. We propose a novel approach that enhances features from well-trained DNNs using an improved radial basis function (RBF) neural network, targeting recognition within individual gesture categories. We achieve this by clustering images with a self-organizing map (SOM) network to identify optimal centers for RBF training. Our enhanced SOM, employing the Hassanat distance metric, outperforms the traditional K-Means method across a comparative analysis of various distance functions and the expanded number of cluster centers, accurately identifying hand gestures in images. Our training pipeline learns from hand gesture videos and static images, addressing the growing need for machines to interact with gestures. Despite challenges posed by gesture videos, such as sensitivity to hand pose sequences within a single gesture category and overlapping hand poses due to the high similarities and repetitions, our pipeline achieved significant enhancement without requiring time-related training data. We also improve the recognition of static hand pose images within the same category. Our work advances DNNs by integrating deep learning features and incorporating SOM for RBF training.

Keywords: Deep Neural Network, Radial Basis Function Neural Network, Self Organizing Maps Network, K-Means clustering

Categories: I.2.1, I.2.10, I.2.6, I.4.0, I.4.6, I.4.8, I.4.9, I.5.3

DOI:10.3897/jucs.110291

1 Introduction

Deep learning has attracted widespread interest in recent years due to its powerful ability to extract features from data, allowing for its effective integration into various machine learning methods. Despite the associated high costs of training deep learning models, such as the need for high-end machines or high-performing GPUs, its advantages, including reduced learning and decision-making time; made it a popular choice for a wide range of applications. Among the essential tools in machine learning, neural networks have been applied to a wide range of domains, including image-pattern recognition [Zerdoumi et al., 2018], self-driving vehicle trajectory prediction

[Spielberg et al., 2019], facial recognition [Modi et al., 2021], data mining [Yuan et al., 2021], email spam filtering [Sumathi et al., 2021], and medical diagnosis [Oza et al., 2022]. Presently, neural networks are being deployed in a multitude of ways, and their adoption is rapidly proliferating.

Over the years, researchers have made significant progress in the development of neural network architectures, making them capable of effectively addressing a wide array of tasks. These architectures include a variety of models, such as the perceptron, feed-forward neural network, multilayer perceptron, convolutional neural network (CNN), radial basis function network, recurrent neural network (RNN), LSTM (long short-term memory), sequence models, and modular neural network. Each of these architectures has unique strengths and is optimally suited for specific tasks, thus providing a wide range of opportunities for using neural networks effectively. By exploiting the strengths of these different models, researchers and practitioners alike can maximize the potential of neural networks in solving complex problems and driving advancements in diverse fields.

In addition, many researchers have explored hybrid neural network approaches that combine different architectures to exploit their complementary strengths. For example, [Gerardo et al. 2020] introduced two innovative hybrid architectures that merge morphological neurons with perceptrons. [Lin et al., 2020] proposed a hybrid model that integrates CNNs and bidirectional gated recurrent units for cognitive radio applications. [Zhao et al., 2021] presented a hybrid neuro-probabilistic reasoning algorithm for verifiable attribute-based medical image diagnosis. [Teyeb et al., 2021] developed a driver vigilance monitoring system based on video processing. The system employed two novel transfer learning classifier architectures, based on fast wavelet transform and separator wavelet networks, to classify the driver's eye state and head posture. The fusion of these two sub-systems yielded more accurate assessments of driver vigilance. [Ding et al., 2022] combined graph neural networks and CNNs for hyperspectral image classification. [Yao et al., 2023] introduced a deep hybrid multi-graph neural network (DHMG) for hyperspectral image (HSI) classification. These hybrid approaches demonstrate the potential to achieve improved performance by exploiting the strengths of multiple neural network architectures. In this paper, our primary objective is to advance the performance of traditional neural networks by leveraging the features extracted from deep learning. By incorporating the strengths of deep learning into other neural network models, we anticipate notable improvements in recognition tasks. Our proposed method relies on a well-trained deep neural network. To effectively contribute to training the traditional radial basis function (RBF) neural network, we integrate the self-organizing map (SOM) neural network within the training process to select the optimal cluster centers. After investigating the potential for performance enhancement by varying distance functions, we find that the Hassanat function [Hassanat, 2014] exhibits clear superiority in network performance compared to other functions. Figure (1) illustrates how this metric's output is confined to the range of (0, 1), where values are more similar when they are close to zero and more different when they are closer to one. It had been demonstrated that the measurement was independent of the similarity metric, or anti-noise and anti-outliers. Accordingly, the distance between two values will be closer to one when they are more dissimilar and closer to zero when they are more similar. Through this research, we seek to contribute to the broader field of artificial intelligence and its applications by augmenting the

capabilities of neural networks and enabling the development of more precise and efficient recognition systems.

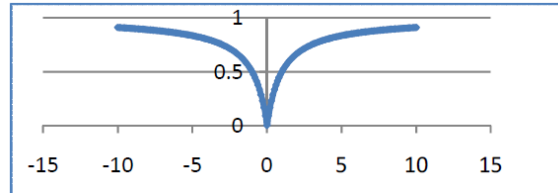


Figure 1: Hassanat Distance

As a result, several contributions could be summarized as follows:

1. **Enhanced Recognition Capabilities:** The study focuses on enhancing the recognition capabilities of neural networks in the context of hand gesture image recognition. By optimizing deep learning features, the proposed approach improves the accuracy and performance of recognizing various hand gestures.

2. **Improved Radial Base Function (RBF) Neural Network:** The study introduces an improved RBF neural network by utilizing a Self-Organizing Map (SOM) network for clustering. This approach determines optimal centers for training the RBF network, outperforming the traditional K-Means method and enhancing the recognition of hand gestures.

3. **Training Pipeline for Videos and Static Images:** The suggested training pipeline addresses the demand for machines to interact with hand gestures in both video and static image formats. By incorporating deep learning features and optimizing the SOM-enhanced RBF network, the pipeline achieves significant improvements in video to enhance the recognition of static hand pose images within the same category.

4. **Advancement in standard Neural Networks:** This work contributes to the advancement of standard neural networks by integrating deep learning features and exploring distance functions. By incorporating the SOM neural network for enhanced training, the study demonstrates progress in the field of artificial intelligence, specifically in the context of hand gesture recognition.

The subsequent sections of this paper will delve into the employed methodologies, implemented algorithms, experimental results, and discussions, providing a comprehensive understanding of the proposed approach. The findings and conclusions derived from this research endeavor will pave the way for further advancements in the field of neural networks and their integration with deep learning, ultimately resulting in improved recognition performance.

2 Methodology of the proposed system

Our proposed system aims to recognize every input image from various datasets, including the MONTALBANO 2 dataset [Escalera et al., 2015], the Hand Gestures and Leap Motion dataset [Marin et al., 2014; Marin et al., 2015], and the Hand Posture and

Gesture datasets [Triesch et al., 2002]. Figure (2) illustrates the overall system, while algorithm (1) has been developed to address outliers in the training system path. The training process involves three main phases: (1) training the CNN on video images, (2) training the RBF using optimal centers previously identified through training the SOM network and (3) gesture recognition by evaluating the system on a separate testing dataset. The following sections will provide a detailed description of each phase, as explained by algorithm (1).

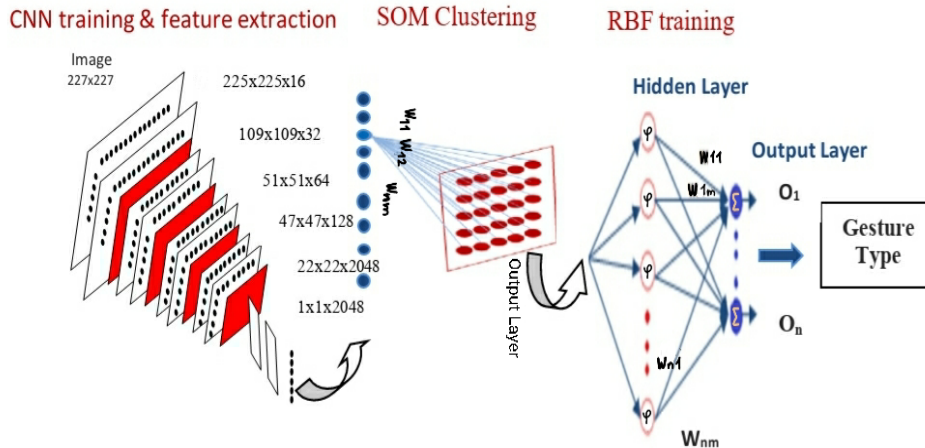


Figure 2: Diagram of the proposed system architecture

Algorithm 1: The proposed work

Input: Sequence of labelled training images $Z = \{(im_1, y_1), \dots, (im_n, y_n)\}$

Output: classified images C_{images}

Begin:

$F_n = \text{TrainCNN}(im_n, y_n)$ /* extract features F of length n by training CNN for classification */

$(C_s, M_s) = \text{TrainSOM}(F_n)$ /* find cluster centers C and their memberships of length s by training SOM */

$C_{images}_n = \text{RBF}(F_n, C_s, M_s, Y_n)$ /* Train RBF to classify images */

END

2.1 Deep Neural Network

The network includes models of different network designs that have the ability to learn, whereas the network design depends on a set of layers that have the ability to extract important features of different sizes. In most networks, the layers shrink so that they flatten at the end of the network. Learning begins with initial weight values \mathbf{W} , to be adjusted by the training to reach optimal weights. The training has been achieved through a loss function that aims to match the values of the required labels \mathbf{G} with the

prediction values P , as in (3), to reduce the errors during back training and adjust weights W by accumulate the derivative of error across layers L as in (4), which depend on momentum Mom and learning rate Lr to descent as in (5) and (6), Where D , M and R are small constant of weight decay and learning rate respectively.

$$E = \text{loss}(G, P) \dots \dots \dots (3)$$

$$W_L = W_{L-1} + Lr_L * Mom_L * W_L \dots \dots \dots (4)$$

$$Lr = R * Lr_L \dots \dots \dots (5)$$

$$Mom_L = M * Mom_{L-1} - D * W_L, Mom_L \dots \dots (6)$$

Numerous deep-learning networks have been proposed in significant studies and have been extremely successful at classifying data. Three networks of different designs have been shared to train each type of data on which this work is applied. The first network has trained on data from scratch and was inspired by the VGG network [Simonyan and Zisserman 2014]; since it was widely used to process video images, it has been referred to as the "small VGG". The practiced network design has consisted of four blocks of twin convolutional layers with a 3x3 kernel filter in each, each block is followed by a max-pooling layer, and feature maps for each block are (16, 32, 64, 128) with two full connection flattened layers of 2048 neurons and each block having a RELU activation function. The second practiced network, "Alex_ImageNet", proposed by [Krizhevsky et al., 2017] on ImageNet, has 20 layers, alternating between convolutional layers of different kernel filters and Max Pooling layers, followed by an activation layer, which is RELU, and two normalization layers. The third network is "VGG_face", which was proposed in the work published by [Parkhi, et al. 2015] on more than two million face images, which is represented by 36 layers consisting of five blocks, as the first two blocks contain two convolutional layers that alternate with the activation layers, while the other three blocks consist of three convolutional layers that alternate with the activation layers. Each convolutional layer consists of a fixed-length filter, which is 33, and each block is followed by a pooling layer to reduce the overall layer size. The network ends with two flattening layers, which represent a convolutional layer with a kernel filter of length 1x1.

2.2 Feature Extraction

The feature extraction stage plays a crucial role in connecting any two techniques by exploring the significant characteristics of the input data that trained the neural networks. It facilitates the transfer of outputs from the deeply trained network to the RBF network, thereby enhancing overall recognition performance. Specifically, the final flat layer, located just before the last prediction layer in the deep network, serves as an ideal location for feature extraction as it encapsulates the essence of a single video frame.

Feature extraction involves transforming the input image through network layers using weights, biases, and activation functions. These operations aim to capture meaningful characteristics of the image frame in the dataset. Equation (7) represents this process:

$$\text{Features} = f(\text{Weights} * \text{Image frame}) \dots \dots \dots (7)$$

In this expression:

- Weights represent the adjusted parameters of the neural network.
- Image frame corresponds to the input data or activations from the previous layer.
- The function f() denotes the element-wise application of an activation function to the weighted sum of input image frames.

2.3 RBF Neural Network

One model of the neural network is RBF, which consists of three layers [McCormick, 2013]. The first layer corresponds to the network inputs, the second is a hidden layer consisting of a number of RBF non-linear activation units, which can be expressed as in (8) and (9); and the last one corresponds to the final output of the network, as expressed in (10), which optimizes the weights of Theta T. As we can see, finding the activation function represents the membership of each input vector **F** of one image frame in the training set videos to the cluster centers **K** of all categories, whereas the centers of the clusters of each category are also involved in finding the average distance of each value in the cluster, So clustering the input training data has to be the first step before calculating the activation function **A**, as in (9), to end up with the output **T**, as in (10), which represents the optimal weights in our case, whereas **Y** is the binary label of each category.

$$\beta = \frac{1}{2(\frac{1}{m} \sum_{i=1}^m \|F_i - K\|^2)^2} \dots \dots \dots (8)$$

$$A(F) = e^{-\beta \|F - K\|^2} \dots \dots \dots (9)$$

$$T = A(F)^T A(F). A(F)^T * Y_K \dots \dots \dots (10)$$

The RBF network's nonlinear mapping capabilities, feature extraction through clustering especially inside every single category, and comprehensive training contribute to its effectiveness in video image gesture recognition. It can capture complex relationships, extract meaningful features, and then achieve accurate classification, making it a valuable tool for this task.

The purpose of using the RBF network in this work is to improve image features that are gotten from the well-trained deep neural network, so the inputs to RBF are image frame features, while the output is the corresponding predicted class. Despite not considering sequential time correlation between image frames in a single hand gesture video during CNN training, the clustering process in the RBF network can group similar hand gesture images within the same gesture video. RBF network work as localization functions, meaning it gives higher values for images closer to the centers of the RBF and lower values for images further away (Equation 9). This allows sequences of time-series images that form a single gesture in the video to be grouped together, helping to find meaningful connections between the video images. Additionally, in classifying static hand images, RBF can find spatial similarity between hand gestures belonging to

the same category, which enhances the classification ability. Additionally, RBF network is inherently non-linear, meaning that it can capture complex relationships between input features and output labels through the activation function. Moreover, unsupervised clustering using the SOM network has been used as a first stage to detect each image membership and its related image centers; our proposed work has proven SOM network training overcomes the K-means traditional clustering, as in algorithm (1).

2.4 Self Organizing Maps Network (SOM)

Training the SOM network has been formulated [Brixy, 2017] as the problem of clustering to find the optimal cluster centers and their memberships for every category, such that the output neurons layer are cluster centers in which every input feature in one category should belong to one of the centers, as in the presented algorithm (2).

Two layers of neurons have only the neural network of SOM: one for the input and one for calculating the outcome. The output layer is made up of a square set of neurons, each of which starts out with a weight that can be trained to compete with neighbouring cells to find input vectors that match; after that, the weight of each neuron starts to change cooperatively with neighbouring cells until it reaches ideal weights that are proportional to each input feature. While it uses competitive learning to extract features that are close topologically from one input training vector, as in the phase of assigning each input vector to the nearest weight of output neurons using the distance function, it should be kept in mind that this network does not really contain the actual value of the desired class; therefore, the network learns unsupervised. The second stage in SOM training is to detect the optimal cluster centers by considering the label of the closest distance by one of the distance functions for each input feature to each neuron in the SOM output layer, as shown in the algorithm (2). This algorithm has been implemented on the same input features that have trained the SOM network.

The great role of the distance function in training SOM neural networks has been noticed, either to determine the winner neuron in the output layer or to determine the label of each neuron in the output layer also after the training to assign each input feature to the optimal center, so some of the distance calculation functions have been adopted as in table (1).

Function Name	Definition
Manhattan, [Gan et al., 2007]	$\sum_{i=1}^n X_i - Y_i $
Euclidean , [Chase et al., 2008]	$\sqrt{\sum_{i=1}^n X_i - Y_i ^2}$
Average Euclidean, [Gan et al., 2007]	$\sqrt{\frac{1}{2} \sum_{i=1}^n (X_i - Y_i)^2}$
Canberra , [Akila et al.,2013]	$\sum_{i=1}^n \frac{ X_i - Y_i }{ X_i + Y_i }$
Jaccard, [Jaccard, 1901]	$\frac{\sum_{i=1}^n (X_i - Y_i)^2}{\sum_{i=1}^n X_i^2 + \sum_{i=1}^n Y_i^2 - \sum_{i=1}^n Y_i X_i}$
ChoD , [Orlóci, 1967]	$\sqrt{2 - 2 \frac{\sum_{i=1}^n X_i Y_i}{\sum_{i=1}^n X_i^2 \sum_{i=1}^n Y_i^2}}$
Hassanat, [Hassanat, 2014]	$\begin{cases} 1 - \frac{1 + \min(X_i, Y_i)}{1 + \max(X_i, Y_i)}, & \min(X_i, Y_i) \\ 1 - \frac{1 + \min(X_i, Y_i) + \min(X_i, Y_i) }{1 + \max(X_i, Y_i) + \min(X_i, Y_i) }, & \min(X_i, Y_i) \end{cases}$

Table 1: Different Function Distance

In all functions, depending on the required distance calculation, we utilize the variables X and Y to represent either the input feature or the winning neuron, as indicated in the algorithm (2).

The SOM network is beneficial for image gesture clustering because it can preserve the topological structure of the input images, enabling it to capture spatial additional to temporal relationships of patterns in video image frames, this means that the SOM grid (output layer) will form a map of the video gesture, where nearby neurons represent similar frames in the video gesture and similar images in static dataset. SOM is capable of capturing non-linear relationships through the competitive learning, making it a suitable choice for modeling complex hand gestures, while hand gestures often involve non-linear variations in shape and appearance. These make SOM network suitable for clustering and organizing video and static images based on their hand poses similarity. Furthermore, SOM training is more robust to outliers and noise. This is because SOM training uses a neighborhood function to update the neuron weights. The neighborhood function ensures that the neuron weights are updated in a way that preserves the local structure of the data. This makes SOM training less sensitive to outliers and noise than k-means training and can converge faster.

Algorithm 2: Training of the proposed pipeline

Input:

Sequence of labelled image frame features of $Z = \{(F_1, y_1), \dots, (F_n, y_n)\}$

Output:

Optimized weights of output Layer Theta θ

Begin:

1. For each Category C in Z
2. Set a value for the dimensions of the neuron matrix of the output layer $m=i*j$
3. Set a value for Learning rate Lr
4. Initialise random W_{ij}
5. For each iteration t $\leftarrow 0$ to P
6. Randomly select one feature vector F of length L from a set S of feature vectors belonging to a specific category.
7. $idx \leftarrow D(w_{ij}, F_L)$ /* Distance functions as in Table (1) to determine the winner image feature index idx from the output neurons */
8. $N_{ij}(F_L) \leftarrow \exp\left(-\frac{idx_{ij}^2}{2\sigma^2}\right)$ /* determine the neighbourhood radius using Decay function N of each Output neuron i & j */
9. $Lr(t+1) \leftarrow Lr(t) \cdot e^{-(t-1)/P}$ /* adjust learning rate Lr */
10. $WO_{ij}(t+1) \leftarrow W_{ij}(t) + Lr * N_{ij}(t)[F_L - W_{ij}(t)]$ /* Adjust weights */
11. End
- /* determine the predicted label of each neuron in the output layer */
12. For k l $\leftarrow 1$ to m /* number of output layer neurons are $m=i*j$ */
13. $idxx \leftarrow D(F_{LS}, W_k)$ /* Distance functions as in Table (1) to find index idxx of winner (closest) image feature F of length L*S for the output neurons */
14. $OptimalC_k \leftarrow \text{Find}(idxx \text{ in } y)$ /* attribute idxx to the label y on the output layer to find the optimal center values Optimal C */
15. End For
- /* Assign each input feature to the optimal predicted center */
16. For each feature F of length L from a set S in C
17. $M_m \leftarrow D(OptimalC_m, F_L)$ /* Distance functions Table (1) to find the closest output Neuron from the current image feature F, in order to determine the membership index for each feature in M (of length $m = i * j$) */
18. End For
19. $Sigma_L \leftarrow \frac{1}{N} \sum_{i=1}^m \sum_{j=1}^m \|M_i - OptimalC_j\|^2$ /* find sigma of length L */
20. $Beta_L \leftarrow \frac{\sum_{i=1}^m Sigma_i}{N}$ /* find Beta of length L */
21. $Beta_{All} \leftarrow Beta_{All} + Beta_L$ /* Accumulate the beta for each category */
22. $OptimalC_{All} \leftarrow OptimalC_{All} + OptimalC_L$ /* Accumulate the cluster centers for each category */
23. End For /* per category */
24. $\varphi(F) \leftarrow e^{-Beta_{All} \|F - OptimalC_{All}\|^2}$ /* find Activation */
/* Find θ_{All} which is the weights for the output Layer over all categories C */
25. $\theta_{All} \leftarrow \varphi(F)^T \varphi(F) \cdot \varphi(F)^T * Y_c$ /* Y_c is a binary output for Specified category C */

End

2.5 System Evaluation

The evaluation stage is critical for assessing the proposed system's performance, as the aim is to be able to recognize images that have not been previously trained by calculating the accuracy (Acc) as in (11) and applying the system to other data for testing. The process starts by adapting the optimal weights to extract features from the last layer, i.e., before the prediction layer of the deeply trained networks. The adaptation has been done by applying the optimal weights to all layers of the network without back-adjusting the weights as in expression (7). The process has been followed by getting the prediction P from the trained RBF network by matching the optimal weights from RBF output layer neurons, as in (12). The prediction P is the output value from the activation function $\phi(F)$, which represents the relation between the optimal value of clustered centers and beta that has been obtained from training SOM as in (9), multiplied by the optimal weights W for the output layer that corresponds to each image.

$$Acc = \frac{\text{Number of right predicted image frames}}{\text{Total image frames in training set}} \dots\dots\dots (11)$$

$$P = \phi(F) * W \dots\dots\dots (12)$$

Therefore, the predicted values will have the same number of categories that the network has been trained to recognize. Since the goal of training the RBF network is to maximize the recognition values of one category by reducing the differences between the target label and the winning output layer neuron, the winning prediction value represents the value of the cell corresponding to the highest value among the predicted values.

3 Experimental Results

In this work, three different datasets of hand gestures were tested: RGB and depth videos, as well as skeletal data that was taken from a sensor for motion tracking, were used in the multi-modal gesture detection system Chalearn 2014 [Escalera et al. 2015], and the imaging resolution was 640 x 480. The ground truth labels for the beginning and ending frames were given, as well as skeleton data for each signer. One person was seated in front of the camera while 27 signers performed the gestures. The backgrounds were varied, and the signers had a vocabulary of 20 Italian gesture categories of total 268517 single image frames. The way the signer uses their hands to represent each category differs; therefore, they could alternate between using their left and right hands to represent the same movement. Each signer presents a long, continuous video in which he performs several hand gestures of different types.

Initially, the continuous video of one signer was cut into a group of meaningful gesture videos, where the static movements were neglected, so that each video represented a single gestural motion of a specific category. Thus, very short and separate videos were obtained for each category. As part of the pre-processing of video images, each frame has been trimmed to perfectly fit the size of the signer using the skeletal values previously provided in the dataset to reduce the background effectively.

The second experimented dataset for hand gesture and leap motion images [G. Marin et al. 2014; G. Marin et al. 2015] includes 14 individuals performing ten different hand gestures ten times each, with each gesture image differing in the pose and location of the hand, bringing the total number of images to 1400. We have extracted three examples of each gesture belonging to the same person to be included in the test data; thus, seven examples of one gesture for each person were included in the training data. With this division, we obtained training data for 980 hand gestures, while the test data contained 420 hand gestures for the same people included in the training set, but with different hand poses and locations. We have chosen this division of the data to match the conditions surrounding the first dataset.

The third dataset experimented is the Hand Posture and Gesture dataset [Triesch et al., 2002]. It consists of 10 static hand gestures from 24 people with three different backgrounds. The data was divided into three parts, with two-thirds used for training and one-third used for testing, so that each class contained 54 samples during training.

Whether in the static or video image datasets, it is common to encounter different hand pose images that represent the same gesture category. However, many semi-similar hand pose images have been spotted. Figure (3) showcases samples from the three datasets utilized in our work, highlighting instances where different images exhibit various hand poses, despite belonging to the same gesture category.

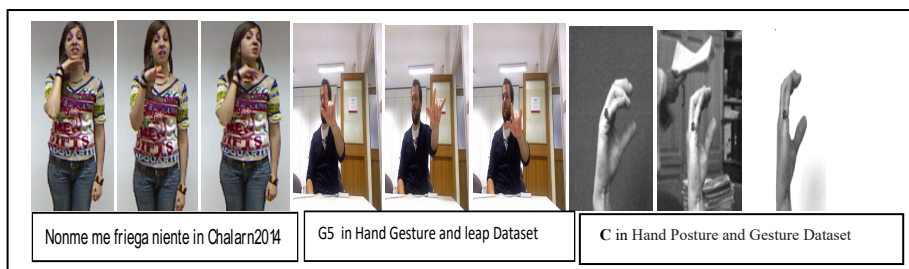


Figure 3: Various Hand poses within the same hand gesture category

The three datasets of hand gesture images, both moving and static, were trained on three deep neural networks with different designs, each of which was previously described in Section 2, and the efficiency of each of them was compared by calculating the accuracy before and after applying the training of the traditional RBF network with k-mean[Likaset al., 2003;Vedaldi et al., 2010] clustering on all of them as in table (2). The number of initial centers for each category in training RBFN-based K-Means clustering is the length of the category divided by 100 for the Chalarn2014 dataset, 20 for both hand gesture and leap dataset and Hand Posture and Gesture Dataset. The initial cluster centers are randomly chosen within each category, and the clustering process was enhanced by neglecting zero-cluster categories. This neglecting has led to a variable-length sigma for each category, and therefore a variable-length beta, as it is an accumulation of each sigma category. The relation between the length of centers and the recognition accuracy in RBF is explained in Table 2. For the purpose of clarity, we

have used the sign (_) in giving the name of each pipeline to denote the path of network training.

Network name	Chalearn 2014			Hand Gesture&Leap			Hand Posture and Gesture		
	Acc. % CNN	Cent. length	Acc.% Kmeans _RBF	Acc. % CNN	Cent. length	Acc.% Kmeans _RBF	Acc. % CNN	Cent. length	Acc.% Kmeans _RBF
Small VGG	56.2	2240	64.2	35.9	46	44.3	71.2	26	80.1
Alex ImageNet	68.1	2252	74.2	65.2	49	70.3	90.0	28	94.7
VGG face	69.1	2255	75.5	77.6	50	85.00	94.5	30	96.1

Table 2: Deep network performances versus RBF K_means

For each practiced network, the Softmax function was selected as the loss function, while training weights were adjusted using gradient descent [Krizhevsky et al., 2012]. During training, the image sequences were scrambled. The network was trained on randomly batched images, and the learning rate was equal to 0.001, with a weight decay of 0.0005 and momentum equal to 0.9; the training was done using a batch size of 36, while the RBF network had been trained in one shot.

Weights were trained from scratch in the VGG network for a resized square image of size 227, whereas in the Alex network, training occurred after matching the weights from the pre-trained network on ImageNet for a square image of length 227, as well as for training the third network, VGG_face, after transferring the weights from the pre-trained network to a square image of dimensions 224. The image sizes used for training across the three networks were purposefully adapted to conform with the architecture of the pre-trained network. This alignment facilitated the successful fine-tuning of weights across all layers of the CNN. Notably, the only modification made was in the prediction layer, adjusting it to accommodate the 20 or 10 classes specific to the problem.

For both the pre-trained networks, an additional full connection layer of length 2048 was added before the decision layer with an activation layer in order to match the length of the features extracted from the last layer that precedes the prediction layer. By comparing the performances of the different participating networks, we have noticed the superiority of the VGG face network as a pretraining network, considering the accuracy metric in (10). This effect was reflected in the RBF network recognition, which depends on the features extracted from it; therefore, in both types of data participating in the experiment, we adopted the features extracted from the VGG face network to present the RBF network performance comparison that adopts the SOM training for clustering.

Table 3 shows the impact of initial cluster centers selection for each category (No. of output neurons), the final length of centers (Cent. Leng.), which has the same length as beta with a width equal to the length of the input feature, and the impact of network training SOM iterations on the recognition accuracy of the radial basis function (RBF) network. The table also highlights the Euclidean distance function used for measuring distances. The results showcase these factors' influence on the network's overall

performance in accurately recognizing hand gestures. The significant differences observed between the initial number of cluster centers belonging to each class and the final total number of cluster centers (Cent. leng.) across all data can be attributed to the discarding step. This step involves removing the cluster centers of empty clusters, i.e., clusters to which no data points were assigned during the training process. As a result, the final beta value which represents the aggregated cluster centers from each category, exhibit inconsistency in length.

Consequently, the final total number of cluster centers (Cent. leng.) obtained from network training displays inconsistency in length also. Important note: It is worth mentioning that in our study, the initial learning rate was set to 0.0001 during the training phase to adjust the Self-Organizing Map (SOM) weights. This parameter setting is crucial as it affects the convergence and performance of the SOM network.

Chalearn 2014				Leap and hand Gesture				Hand Posture and Gesture			
No. of output neurons	Cent. Leng.	No. of iter.	Acc %	No. of output neurons	Cent. Leng.	No. of iter.	Acc %	No. of out.	Cent. Leng.	No. of iter	Acc %
43x43	4012	10	76,2	2*2	27	10	82.4	2*2	24	10	96.2
44x44	4329	10	76,3	3*3	50	10	85.4	3*3	39	10	97.5
45x45	4413	10	76,6	4*4	70	10	86.5	4*4	47	10	97.1
46x46	5415	10	76,5	5*5	93	10	86.4	5*5	60	10	96.1

Table 3: RBF_SOM performance in different datasets

The comparison highlights that SOM networks exhibit a higher number of cluster centers compared to K-means, which can be attributed to several factors. Firstly, SOM networks possess a topological structure by arranging images in a grid-like pattern, enabling them to capture subtler relationships within the image datasets. This is a feature absent in K-means, which is limited to partitioning data into disjoint clusters. Consequently, the SOM network can create cluster centers that represent distinct regions of the input images, even if they are not linearly separable. This results in a more nuanced and informative representation of the data compared to K-means. Secondly, SOM updates its weight vectors based on the number of neighboring images. This mechanism helps in identifying clusters that may be densely packed or have complex relationships within the image datasets. By considering neighboring images, the SOM network can better capture the underlying structure of the data, leading to a more accurate and detailed representation of the clusters. Thirdly, SOM networks exhibit greater resilience to noise compared to K-means. This resilience allows them to accurately identify clusters even in noisy images. The neighborhood function in SOM plays a crucial role in this capability, enabling the network to filter out the effects of noise and make robust cluster assignments. This makes SOM a more suitable choice for analyzing data that is corrupted by noise or outliers. Additionally, we presented a comparison in Table 4 between the different methods for calculating the distance when

considering the settings that achieve the highest accuracy in Table 3 in order to clarify the effect of each method on the results of recognizing the RBF network that has been used for the SOM on both datasets.

Name of Distance functions	Acc.% Chalearn 2014	Acc.% Hand Gesture & Leap	Acc.% Hand Posture and Gesture
Manhattan, [Gan et al., 2007]	76.06	86.2	97.5
Euclidean, [Chase et al., 2008]	76.6	86.5	97.6
Average Euclidean, [Gan et al. 2007]	76.71	86.72	97.8
Canberra, [Akila et al., 2013]	76.89	86.81	97.91
Jaccard, [Jaccard, 1901]	77.1	86.91	98.01
ChoD, [Orlóci, 1967]	76.93	86.87	98.1
Hassanat, [Hassanat, 2014]	77.5	87.1	98.3

Table 4: SOM_RBF versus distance function

The Hassanat distance function [Hassanat, 2014] has been proven to outperform other distance functions when evaluating SOM networks, as evidenced in Table 4. One notable advantage of the Hassanat distance function is its invariance to scaling. This has been achieved through normalization using the maximum and minimum values of the image feature, ensuring that the distance between two images remains consistent regardless of scaling. Additionally, the Hassanat distance function has demonstrated robustness to variations in image densities and clusters, making it well-suited for handling hand gesture images with diverse characteristics. It also exhibited resilience to outliers and noise, enabling effective capture of the essential features of hand gesture images. As a result, the Hassanat distance function could be a compelling choice for a wide range of hand gesture recognition tasks.

Despite, in this research, our objective was not to outperform previous works in terms of data performance. However, we managed to achieve accuracy values that are comparable. When working on the Hand Posture and Gesture dataset [Triesch et al., 2002], our method have proven to enhance the features of the Convolutional Neural Network (CNN) using RBF based on SOM. As a result, we attained a high accuracy of 98.3, which is only slightly lower than the highest accuracy of 98.87 attained by [Bouchrika et al., 2018]. It's worth noting that our method remains on par with the state-of-the-art and has the potential for further improvement. Furthermore, the comprehensive explanation of the final confusion matrix for each datasets can be found

in Figure 4 and Figure 5. These figures provide a clear and detailed depiction of the performance evaluation, presenting the true positives, true negatives, false positives, and false negatives for each class. By referring to these visual representations, one can gain valuable insights into the model's accuracy and error patterns, facilitating a deeper understanding of its overall effectiveness in handling the respective datasets.

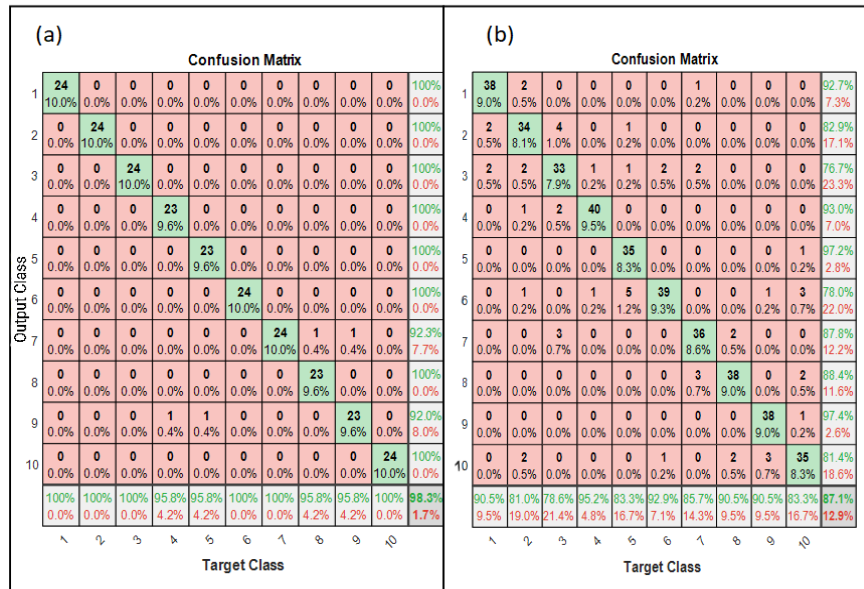


Figure 4: Confusion matrix: (a) Hand Posture and Gesture (b) Hand Gesture & Leap dataset

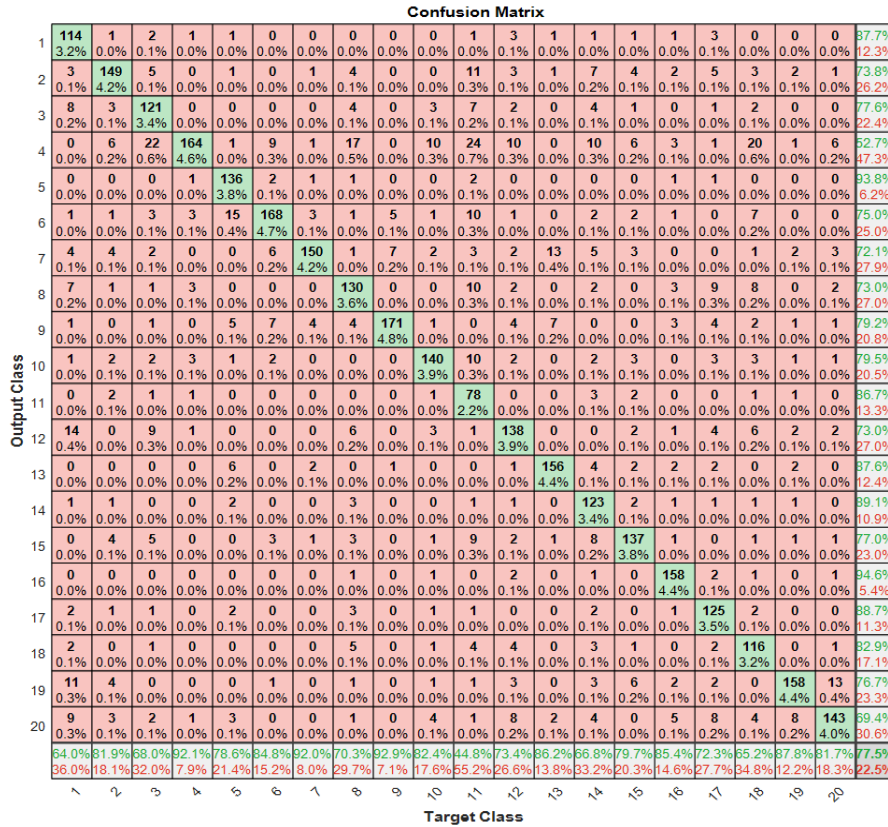


Figure 5: Confusion matrix for Chalearn 2014dataset

According to the efficient pipeline, we have conducted a comparison in Table 5 to evaluate the performance in terms of RBF recognition time. We compared the recognition time using SOM with the recognition time using K-means clustering, specifically considering the average time required to recognize one image during the evaluation (testing) phase.

RecognitionMechanisms	Chalearn 2014	Leap and hand Gesture	Hand Posture and Gesture
RBF SOM	0.23 s	0.00069 s	0.0004 s
RBF K mean	0.033 s	0.00022 s	0.0002 s

Table 5: Execution time comparison between the two Recognition Mechanisms

RBF recognition using K-means has demonstrated significant superiority over RBF recognition using SOM regarding the average execution time for one image feature in

the testing phase. This is because, in K-means, each image test feature needs to be matched to the optimal cluster centers, which also represent features. On the other hand, in SOM, each image test feature needs to be matched to the optimal weights, which is computationally complex. This difference in matching processes contributes to the faster and more computationally efficient performance of K-means. Although the speed difference may depend on other parameters and the rapid advances in computing devices, our primary goal remains recognition efficiency.

In summary, the experimental results provided insights into the performance of various networks and factors in hand gesture recognition. The analysis of parameters such as beta size, cluster center selection, training iterations, and distance calculation methods contributes to a deeper understanding of the recognition capabilities of the RBF network. These findings pave the way for further improvements in the field.

4 Theoretical and Managerial Implications

The current study expands existing theories and provides new insights into the impact and efficiency of clustering algorithm performance on classification networks using the RBF network. Specifically, we examine the effect of the parameters accompanying each algorithm on the clustering quality. Additionally, our study demonstrates that both algorithms exhibit high quality in improving hand gesture recognition when utilizing features extracted from deep neural network training.

Furthermore, we highlight the significance of the method used to calculate the distance between adjacent nodes of the output layer in the SOM network, as it plays a crucial role in enhancing recognition and its overall quality. Our research also provides effective mathematical evidence to substantiate each case, affirming the practical application of our results.

4.1 Theoretical Implications

Our findings contribute to the theoretical underpinnings of clustering algorithms and their impact on classification networks within the RBF network framework. By exploring the influence of parameters on clustering quality, we advance existing theories and offer new insights into algorithm performance. This study enhances our understanding of the factors that determine the effectiveness of clustering algorithms in classification networks.

Moreover, our research sheds light on the importance of the method employed to calculate the distance between adjacent nodes in the SOM network's output layer. Our identification of the method's significant impact on recognition and its quality adds to the existing knowledge on optimizing SOM networks for improved recognition outcomes.

4.2 Managerial Implications

The managerial implications of our research are valuable for practitioners involved in hand gesture recognition and deep neural network applications. We demonstrate that the careful consideration of clustering algorithms and their accompanying parameters can enhance hand gesture recognition using features extracted from deep neural

network training. These findings provide actionable guidance to practitioners seeking to enhance the accuracy and performance of hand gesture recognition systems.

Additionally, our research offers effective mathematical evidence to support the identified cases and their implications. This evidence bolsters the practical application of our results, empowering practitioners to confidently implement our findings in real-world scenarios.

In conclusion, our study contributes to the theoretical understanding of clustering algorithm performance in classification networks, specifically within the RBF network. Furthermore, our findings hold significant managerial implications for practitioners involved in hand gesture recognition, offering insights into optimizing recognition systems using clustering algorithms and deep neural network training. The provided mathematical evidence further strengthens the practical applicability of our results. Future research should build upon these findings to advance the field and explore additional opportunities for improving recognition systems.

5 Conclusions

In conclusion, our research has demonstrated the significant impact of incorporating image clustering as a pre-training step in a standard neural network for hand gesture recognition in videos. We have shown that this approach improves accuracy, particularly for videos containing diverse image frames within the same category that can be effectively clustered together. The effectiveness of our approach was further supported by the recognition of static hand poses, where the similarity between images was more pronounced.

Our study has also highlighted the influence of design variations in deep neural networks on the performance of standard neural networks, specifically in terms of the extracted features. By exploring the training of the radial basis function (RBF) network using features derived from various deep neural network architectures, we aimed to enhance hand gesture recognition datasets. Through rigorous experimentation with both video and static image datasets, we have observed that fine-tuning a pre-trained network with a deeper design, such as the VGG_face network trained on larger datasets, resulted in significantly superior recognition performance using the RBF network.

To efficiently predict ideal cluster centers for highly similar hand gesture image features, we developed a method that leverages the self-organizing map (SOM) network. The SOM network, guided by the Hassanat distance function, effectively mitigates the impact of noisy predicted centers during weight updates. This facilitates the convergence of centers through the underlying image features, resulting in an increase in the formation of distinct and meaningful clusters within the predetermined grid size. Additionally, the selection of companion neighbors for the winning output neurons from the SOM network further optimizes the performance of the RBF network, which is proved by comparing it to K_means clustering.

While the RBF network-based K_means clustering demonstrates impressive speed in image evaluation, we emphasize that the efficiency of SOM clustering remains a decisive factor to consider. The SOM algorithm offers valuable advantages by effectively organizing input data into clusters while preserving their topological relationships. Therefore, careful consideration should be given to the efficiency and effectiveness of SOM clustering in image evaluation tasks.

However, it is important to acknowledge the limitations of our trained deep learning model on temporal shard frame image features (3D feature) for video recognition, specifically in the context of short video lengths for single gestures in the Chalearn 2014 dataset. Future research is recommended to investigate additional neural network architectures and expand the proposed training pipeline to include broader and deeper designs, enabling further improvements in recognition accuracy, especially for complex sign language sentences containing overlapping and challenging instances.

In summary, our research has contributed to the field of hand gesture recognition by demonstrating the importance of image clustering, the influence of deep neural network designs, and the effectiveness of RBF network based SOM network with the Hassanat distance function. These findings provide valuable insights for improving recognition performance and have the potential to enhance various applications in the field of computer vision.

In the future, we plan to investigate more neural networks and expand the training pipeline proposed in this work to include broader and deeper designs that can further improve recognition, especially for complex sign language sentences containing overlapping and more challenging instances of recognition.

Supplementary Materials

The code files used in this study are available on GitHub at https://github.com/MiraMe5/RBF_SOM

References

- [Akila et al., 2013] Akila, A., and E. Chandra. "Slope finder—A distance measure for DTW based isolated word speech recognition." *International Journal of Engineering and Computer Science* 2, no. 12 (2013): 3411-3417.
- [Bouchrika et al., 2018] Bouchrika, Tahani, OlfaJemai, Mourad Zaied, and Chokri Ben Amar. "Rapid and efficient hand gestures recognizer based on classes discriminator wavelet networks." *Multimedia Tools and Applications* 77 (2018): 5995-6016. <https://doi.org/10.1007/s11042-017-4510-7>
- [Brixy, 2017] Brixy, Mareva. Self-organising Map for handwritten number classification. 2017: <https://github.com/marevab/SOM>
- [Chase et al., 2008] Chase, L. D., & Chase, L. Euclidean Distance. College of Natural Resources, Colorado State University, Fort Collins, Colorado.(2008), USA, 824-146.
- [Ding et al., 2022] Ding, Yao, Zhili Zhang, Xiaofeng Zhao, Danfeng Hong, Wei Cai, Chengguo Yu, Nengjun Yang, and Weiwei Cai. "Multi-feature fusion: Graph neural network and CNN combining for hyperspectral image classification." *Neurocomputing* 501 (2022): 246-257.
- [Escalera et al., 2015] Escalera, Sergio, Xavier Baró, Jordi Gonzalez, Miguel A. Bautista, MeysamMadadi, Miguel Reyes, Víctor Ponce-López, Hugo J. Escalante, Jamie Shotton, and Isabelle Guyon. "Chalearn looking at people challenge 2014: Dataset and results." In *Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part I* 13, pp. 459-473. Springer International Publishing, 2015.
- [Gan et al., 2007] Gan, Guojun, Chaoqun Ma, and Jianhong Wu. *Data clustering: theory, algorithms, and applications*. Society for Industrial and Applied Mathematics, 2020.

- [Gerardo et al., 2020] Hernández, Gerardo, Erik Zamora, Humberto Sossa, Germán Téllez, and Federico Furlán. "Hybrid neural networks for big data classification." *Neurocomputing* 390 (2020): 327-340.
- [Hassanat, 2014] Hassanat, Ahmad Basheer. "Dimensionality invariant similarity measure." arXiv preprint arXiv:1409.0923 (2014).
- [Jaccard, 1901] Jaccard, Paul. "Étude comparative de la distribution florale dans une portion des Alpes et des Jura." *Bull Soc Vaudoise Sci Nat* 37 (1901): 547-579.
- [Krizhevsky et al., 2017] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet classification with deep convolutional neural networks." *Communications of the ACM* 60, no. 6 (2017): 84-90.
- [Krizhevsky et al., 2012] Krizhevsky, Alex. "Advances in neural information processing systems." (No Title) (2012): 1097.
- [Likas et al., 2003] Likas, Aristidis, Nikos Vlassis, and Jakob J. Verbeek. "The global k-means clustering algorithm." *Pattern recognition* 36, no. 2 (2003): 451-461.
- [Lin et al., 2020] Lin, Rendeng, Wenjuan Ren, Xian Sun, Zhanpeng Yang, and Kun Fu. "A hybrid neural network for fast automatic modulation classification." *IEEE Access* 8 (2020): 130314-130322.
- [Marin et al., 2014] Marin, Giulio, Fabio Dominio, and Pietro Zanuttigh. "Hand gesture recognition with leap motion and kinect devices." In *2014 IEEE International conference on image processing (ICIP)*, pp. 1565-1569. IEEE, 2014.
- [Marin et al., 2015] Marin, Giulio, Fabio Dominio, and Pietro Zanuttigh. "Hand gesture recognition with jointly calibrated leap motion and depth sensor." *Multimedia Tools and Applications* 75 (2016): 14991-15015.
- [McCormick, 2013] McCormick, Chris. "Radial basis function network (rbfn) tutorial." BERT eBook (2013).<https://mccormickml.com/2013/08/15/radial-basis-function-network-rbfn-tutorial/>
- [Modi et al., 2021] Modi, S., & Bohara, M. H. (2021, May). Facial emotion recognition using convolution neural network. In *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 1339-1344). IEEE.
- [Orlóci, 1967] Orlóci, Laszlo. "An agglomerative method for classification of plant communities." *The Journal of Ecology* (1967): 193-206.
- [Oza et al., 2022] Oza, Parita, Paawan Sharma, Samir Patel, and Pankaj Kumar. "Deep convolutional neural networks for computer-aided breast cancer diagnostic: a survey." *Neural Computing and Applications* 34, no. 3 (2022): 1815-1836.
- [Parkhi et al., 2015] Parkhi, Omkar, Andrea Vedaldi, and Andrew Zisserman. "Deep face recognition." In *BMVC 2015-Proceedings of the British Machine Vision Conference 2015*. British Machine Vision Association, 2015.
- [Simonyan and Zisserman, 2014] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).
- [Spielberg et al., 2019] Spielberg, Nathan A., Matthew Brown, Nitin R. Kapania, John C. Kegelman, and J. Christian Gerdes. "Neural network vehicle models for high-performance automated driving." *Science robotics* 4, no. 28 (2019): eaaw1975.

[Sumathi et al., 2021] Sumathi, S., and Ganesh Kumar Pugalendhi. "Cognition based spam mail text analysis using combined approach of deep neural network classifier and random forest." *Journal of Ambient Intelligence and Humanized Computing* 12 (2021): 5721-5731.

[Teyeb et al., 2021] Teyeb, Ines, Ahmed Snoun, OlfaJemai, and Mourad Zaied. "Fuzzy logic decision support system for hypovigilance detection based on CNN feature extractor and WN classifier." *Journal of Computer Science* 14, no. 11 (2018): 1546-1564.

[Triesch et al., 2002] Triesch, Jochen, and Christoph von der Malsburg. "Classification of hand postures against complex backgrounds using elastic graph matching." *Image and Vision Computing* 20, no. 13-14 (2002): 937-943.

[Vedaldi et al., 2010] Vedaldi, Andrea, and Brian Fulkerson. "VLFeat: An open and portable library of computer vision algorithms." In *Proceedings of the 18th ACM international conference on Multimedia*, pp. 1469-1472. 2010.

[Yao et al., 2023] Yao, Ding, Zhang Zhi-li, Zhao Xiao-feng, Cai Wei, He Fang, Cai Yao-ming, and Wei-Wei Cai. "Deep hybrid: multi-graph neural network collaboration for hyperspectral image classification." *Defence Technology* 23 (2023): 164-176.

[Yuan et al., 2021] Yuan, Chunmei, Yikun Yang, and Yang Liu. "Sports decision-making model based on data mining and neural network." *Neural Computing and Applications* 33 (2021): 3911-3924.

[Zerdoumi et al., 2018] Zerdoumi, Saber, AznulQalid Md Sabri, AmirrudinKamsin, Ibrahim AbakerTargio Hashem, Abdullah Gani, Saqib Hakak, Mohammed Ali Al-Garadi, and Victor Chang. "Image pattern recognition in big data: taxonomy and open challenges: survey." *Multimedia Tools and Applications* 77 (2018): 10091-10121.

[Zhao et al., 2021] Zhao, Gangming, Quanlong Feng, Chaoqi Chen, Zhen Zhou, and Yizhou Yu. "Diagnose like a radiologist: Hybrid neuro-probabilistic reasoning for attribute-based medical image diagnosis." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, no. 11 (2021): 7400-7416.