Title	Hand gesture recognition based on a Harris Hawks optimized Convolution Neural Network
Authors(s)	Gadekallu, Thippa Reddy, Srivastava, Gautam, Liyanage, Madhusanka, et al.
Publication date	2022-05
Publication information	Gadekallu, Thippa Reddy, Gautam Srivastava, Madhusanka Liyanage, and et al. "Hand Gesture Recognition Based on a Harris Hawks Optimized Convolution Neural Network." Elsevier, May 2022. https://doi.org/10.1016/j.compeleceng.2022.107836.
Publisher	Elsevier
Item record/more information	http://hdl.handle.net/10197/25916
Publisher's version (DOI)	10.1016/j.compeleceng.2022.107836

Hand Gesture Recognition based on a Harris Hawks Optimized Convolution Neural Network

Thippa Reddy Gadekallu[a], Gautam Srivastava[b,d,*], Madhusanka Liyanage[c], Iyapparaja M[a], Chiranji Lal Chowdhary[a], Srinivas Koppu[a], Praveen Kumar Reddy Maddikunta[a]

[a]*School of Information Technology, Vellore Institute of Technology, Vellore, India (email: {thippareddy.g, iyapparaja.m, chiranji.lal, srinukoppu, praveenkumarreddy}@vit.ac.in)*
[b]*Department of Mathematics and Computer Science, Brandon University, Manitoba, Canada.*
(email: srivastavag@brandonu.ca)
[c]*School of Computer Science, University Collage Dublin, Ireland and Centre for Wireless Communications,*
University of Oulu, Finland
(email: madhusanka@ucd.ie, madhusanka.liyanage@oulu.fi)
[d]*Research Centre of Interneural Computing, China Medical University, Taichung, Taiwan*

Abstract

Hand gestures are an effective means of communication, especially when we are communicating with people who cannot understand our language and also in human-computer interaction. Understanding the hand gestures is very important to ensure that the listeners understand what the speakers are trying to communicate. Even though several researchers have proposed deep learning-based models for hand gesture recognition, the hyper-parameter tuning of these models is relatively unexplored. In this work, Convolutional Neural Networks (CNN) are used to classify hand gesture images. To tune the hyper-parameters of the CNN, a recently developed metaheuristic algorithm, namely, the Harris Hawks Optimization (HHO) algorithm, is used. The comparative analysis proves that the proposed HHO-CNN model outperforms the existing models by attaining an accuracy of 100%.

Keywords: Hand Gesture Classification, Convolutional Neural Networks, Harris Hawks Optimization Algorithm, Image classification.

*Corresponding author

1. Introduction

Mobile and computing systems have evolved significantly over the last few decades. Along with that, the way humans interact with machines has also improved. These Human-Computer interactions (HCI) methods vary from simple keyboard inputs to advanced vision-based gesture recognition systems [1]. Hand gesture recognition is one of the most exciting and important HCI methods [2].Hand gesture recognition opens up a very interesting research domain as it can be used to enable communication for many applications such as mobile phones, smart TVs, robot controls, medical devices, access control systems, smart vehicles, and so on [3]. These applications can be divided into four main classes, i.e. 1). Medical systems and related technologies, 2). Disaster relief and crisis management 3). Human-robot interaction and 4). Entertainment [4]. Moreover, hand gesture recognition provides a natural means of interaction for humans, which is extremely user-friendly. Thus, the hand gesture is one of the most prominent and widely used computer input techniques.

Initially, wearable sensors that were attached to the hands were used for gesture recognition. Such wearable sensors can capture hand movements or finger bending, which can be converted into an electronic signal. Current vision-based hand gesture recognition systems can capture several meaningful inputs without any attached sensors. The advancement of computer vision techniques has fueled the design of wearables through hand gesture recognition systems [3]. As a result, research work related to hand gesture recognition based on vision has received a significant amount of attention during the last few decades [5, 6].

However, the development of an efficient vision-based hand gesture recognition system is a challenging task. Such, systems that are dependent on vision-based hand gesture recognition systems usually require combining knowledge of application-specific programming, algorithm development, and machine learning techniques. In general, it is challenging to deploy vision-based gesture recognition systems in practical environments due to challenges such as achieving the required robustness to achieve user-interface acceptability, application-specific requirements, Â stability, the accuracy of the camera sensor, and lens characteristics, scene, background, and lighting conditions. Also, hand gesture recognition systems have to address the challenges such as distance of communicating device, size, processing, and response speed. Several technological methods are developed to overcome these challenges. Especially, machine learning plays a vital role in mitigating these challenges. With the advancements in computational power, storage, and memory speeds, the training of ML models has become easier these days. Also, due to several advancements in ML, such as deep neural networks and convolutional neural networks, which give better results with more training data, classification of images has become easier these days.

Even though many researchers have worked on hand gesture recognition systems using advanced machine learning models, hyper-parameter tuning, which increases the efficiency of the models, has not been explored much in this domain. To achieve this objective of tuning hyperparameters of CNN, the HHO algorithm is employed in this work. HHO is a recently developed nature-inspired algorithm based on the hunting behaviour of the Harris Hawk. Due to its ability to converge quickly when compared with other meta-heuristic algorithms, HHO identifies the optimal solutions in complex problems. HHO hardly gets stuck in local minima, which acts as a great advantage in identifying the optimum parameters of the CNN. This will greatly enhance the performance of CNN. Apart from achieving accuracy, the training time of the CNN is also drastically reduced as HHO has a quick convergence time compared to the other meta-heuristic algorithms. In this work, an HHO based CNN model is used for the classification of hand gestures.

The main contributions of this work are given below.

- Choosing optimal parameters of convolutional neural networks using a recently proposed HHO algorithm.
- Reduction of the burden on the CNN by reducing the training time.

- 100% accuracy for hand gesture classification is attained.

The rest of the paper is organized as follows. Recent works on hand gesture classification are presented in Section 2. A detailed discussion on the proposed methodology and the related background is presented in Section 3. The results obtained are presented in Section 4, and the paper is concluded in Section 5.

2. Related Work

Several research works on hand gesture classifications are discussed in this section.

The authors in [7] proposed a double-channel convolutional neural network (DC-CNN) algorithm for enhancing the recognition rate of hand gesture movements. In this work, the authors have divided the whole work into three phases. First, they performed preprocessing, denoising, and edge detection against the hand gesture datasets. Next, CNN is fed with hand gesture images. Edge images with different weight parameters for classification are provided as input. The authors used Jochen Triesch (JT) and the NAO Camera hand posture (NC) datasets for experimental work. Based on different experiments, the recognition rate of different gesture recognition of the proposed method is 98.02% which is better than the existing algorithms such as Spatial Pyramid, Bottom-up CNN, Tiled CNN, and MPCNN. However, the work does not concentrate on the recognition of dynamic hand gesture data.

The authors in [8] presented a CNN-based model for classifying hand muscle shrinkages identified at the forearms. Here, the authors have designed an IoT-based forearm prosthesis band embedded with a lot of sensors for analyzing hand muscle movements based on gestures. The proposed system produced an accuracy of 98.15% for hand gesture recognition. Using CNN properties, the authors have proved that the proposed IoT-based model has yielded promising results for each hand gesture. However, the proposed hand prosthesis model consumes more time to recognize hand gesture movements.

Touch screen and hand gesture interaction systems are benefiting people who are visually impaired. In [9], the authors have developed a braille sketch with hand gesture-based motion sensors connected with mobile devices for overcoming the navigation issues of blind people. The proposed device accepted the hand finger motion as an input variable and calculates the x & y coordinates values, swiping speed, pixel rate based on the people inputs. Using an artificial neural network (ANN) with a crow search algorithm (CSA), the received inputs were analyzed for producing the expected outcome of blind people. The accuracy of the proposed model reaches maximum range for top and bottom gestures as 99.9%, 99.5%, left and right gestures as 95.5%, and 94.23%. From the experiments, the proposed model produced high precision with ideal execution for recognizing the hand motion of visually challenged people. However, the proposed model does not focus on the computation time of analyzing hand gesture data.

Another research work is proposed in [10] for recognizing hand gesture movements by using deep learning algorithms. In this work, the authors have segmented the hand gesture data based on the AdaBoost algorithm with the Haar feature. Here, by using the CamShift algorithm the human hand gesture images are separated from background noises. For experimental purposes, they have applied CNN against real-time hand gesture data for recognizing the actual need of people. Finally, results proved that the proposed CNN algorithm obtains a 98.3% accuracy rate for recognizing hand gesture data in real-time applications. However, the authors have used only 10 human gesture data for performing segmentation and recognition. However, the data is very limited data for analyzing the excellence of the proposed algorithm.

In [11], the authors proposed an improved 3D micro-Doppler feature synthesis model to recognize the hand gestures with Grad-CAM (GCAM) model. The Grad-CAM CNN model is used to analyze significant features of 3D gesture regions by suppressing non-essential features from noise regions. The authors have used two convolution layers that use significant features of azimuth, elevation angles from multi-channel micro-Doppler datasets. The demonstration results show that the proposed classification method achieved an accuracy of 96.61%. The proposed work can be extended to consider the temporal significant features of hand gestures for better performance. In another interesting work in [12], the authors have introduced a hybrid model based on shallow learning to recognize the tiniest hand motion gestures based on Unsupervised Frame Representation (UFR), CNN-LSTM, and Supervised Sequence Representation (SSR). In this work, the Range-Doppler Features (RDF) are fed to UFR. Then, SSR is used to capture the significant frame-level features to obtain robust subtle hand gesture recognition. Finally, the CNN classifier is used for classification. Experimental systems used range-doppler images of size 32×32 corresponding to 2750 subtle sequence gestures from 10 dissimilar subjects. The simulations of the proposed system achieved a classification of the accuracy of 87.69%. But, it is not suitable for gesture recognition in videos.

In [13], the authors proposed a 3D hand gesture detection model based on Hierarchically-Structured Convolutional Recurrent Neural Network (HCRNN). This system detects five fingers and the six branches of 3D positions of palm with fully connected Recurrent Neural Network (RNN). The adapted model consists of two modules. In the first module, the encoder takes input as depth image and transforms to significant features such as thumb, palm, middle, index, little and ring fingers. Then, the RNN takes input features from the encoder and predicts the 3D hand gestures. The experiments were performed on ICVL, MSRA, and NYU datasets with mean errors of 6.54, 7.70, and 9.37, respectively. However, the proposed system is not suitable for subtle hand gestures.

Even though many researchers have worked on recognizing the hand gestures using machine learning-based models, not much attention is paid to finding the optimal parameters of the classifiers that can significantly improve the classification performance and also reduce the training time of the classifiers. To address the aforementioned issue, a recently developed nature-inspired algorithm, the Harris-Hawk algorithm is used in this work to choose the optimal parameters of the CNN.

3. Preliminaries and Proposed Architecture

The meta-heuristic optimization algorithms can be effectively used for recognizing gestures. In this work, a recently developed algorithm, HHO, is used for hyper-parameter tuning of CNN for enhancing hand gesture recognition. The HHO algorithm has several advantages compared to its peer meta-heuristic algorithms such as fast convergence rate and avoiding the local optima to a greater extent. The working model of HHO based CNN to classify the hand gesture image dataset is depicted in Fig. 1. The dataset used in this work is gathered from the publicly available Kaggle platform.

Figure 1: Proposed architectural model for classifying the hand gestures.

The following are the steps involved in the proposed model:

- The dataset of hand gesture images is collected from Kaggle.
- To pre-process the data, one-hot encoding is applied.
- Apply Harris Hawks Optimization algorithm for tuning the hyper-parameters of the CNN to classify the hand gesture images.
- To prove the efficiency of the proposed HHO-CNN model, the results obtained are compared with other state-of-the-art meta-heuristic algorithms based on CNN models.

The rest of this section gives a brief introduction of the HHO algorithm, one-hot encoding, and CNN.

3.1. Mathematical Models of HHO

The Harris Hawk is a very intelligent bird, that attacks the prey with a surprise pounce or 7 killing strategies. Initially, the leader of the hawks' pack attempts to attack the prey. If the hawk is unable to catch the prey due to the prey's escaping behaviour, switching strategies are used with the help of other hawks to capture the escaped prey. The main feature of this collaborative technique is that it confuses the prey, and after a while, the prey becomes exhausted and easily caught. The main steps in the HHO algorithm are exploring the prey, followed by a surprise pounce and an attacking technique that is unique to the Harris Hawks.

3.1.1. Exploration phase:

Every Harris Hawk is considered as a candidate solution during the exploration phase. Depending on the intended prey, fitness values are calculated for available potential solutions in each iteration. Eq. 1 explains the exploration performances of Harris Hawks.

$$Y(t+1) = \begin{cases} Y_{\text{rand}}(t) \; - \; s_1 \; |Y_{\text{rand}}(t) \; - 2 \; s_2 Y(t)| \; p \geq 0.5 \\ (Y_{\text{rabbit}}(t) - Y_m(t)) - s_3 \; (LBound + s_4(UBound - LBound)) \; p < 0.5 \end{cases} \tag{1}$$

where $Y(t+1)$ represents the Hawk in second iteration t. $Y_{rabbit}(t)$ represents the position of the prey and $Y_{rand}(t)$ is used for the random solution, which is selected from the existing population. $Y(t)$ represents the Hawk's position vector in the existing iteration t, s_1, s_2, s_3, s_4 and p are randomly scaled factor within the range $[0, 1]$, which are updated in each iteration. Also, $LBound$ and $UBound$ are the lower bound and upper bound of variables. Y_m stands for the average of the solutions.

The positions of Hawks should be limited to the bounds $(UBound - LBound)$ based on the following conditions:

- Generate the solution by randomly selecting a hawk from the current population and other hawks.

- Generate the solution based on the location of the prey, randomly scaling factors and the average hawk positions.

The average solution of selected hawks is represented in Eq. 2.

$$Y_{\mathrm{m}}(t) = \frac{1}{N}\sum_{k=1}^{N} Y_k(t) \tag{2}$$

In Eq. 2, $Y_{\mathrm{m}}(t)$ indicates the average solution. N stands for every possible solution. $Y_k(t)$ infers the location of every solution in each iteration t. When the information from random hawks is used by the hawks, Rule 1 is followed to capture the prey. Another rule that can be used is to share the best solution among hawks by selecting the best hawk for the task.

3.1.2. Transition from exploration to exploitation

Using the HHO algorithm, the fugitive behaviour of prey energy might be reduced gradually. This prey energy ($Energy$) variation elucidates the transition of exploration to exploitation in HHO.

$$Energy = 2Energy_0\left(1 - \frac{t}{T}\right) \tag{3}$$

where $Energy_0$ denotes prey initial energy, T states the maximum iterations, and t stands for the present iteration in Eq. 3.

3.1.3. Exploitation phase

This phase uses four parameter sets for identifying the prey position and executing the attack against them. The following four possible parameter sets help the hawks to derive an attacking strategy against the identified prey.

- Hard besiege
- Soft besiege
- Hard besiege with progressive rapid dives
- Soft besiege with progressive rapid dives

Soft besiege

In the soft besiege phase, the prey uses random jumps to escape from the hawks with less amount of residual energy. At this point, Harris Hawks encircle the prey softly to direct the prey to lose its remaining energy and finally attack it by surprise pounce.

$$Y(t+1) = \Delta Y(t) - Energy \mid JY_{\text{rabbit}}(t) - Y(t) \mid \tag{4}$$

$$\Delta Y(t) = Y_{\text{rabbit}}(t) - Y(t) \tag{5}$$

In Eq. 4 and Eq. 5, $\Delta Y(t)$ and t represent the the difference between the position vectors and current location of the prey, respectively. Also, J denotes the jump power of the prey.

Hard besiege

In this case, the hawks encircle prey hardly and apply surprise pounce by updating the accurate position. Here, the prey is so exhausted and it has low escaping energy. In addition, the Harris Hawks encircle the intended prey to finally perform the surprise pounce. Fig. 2 depicts a simple example of this step with one hawk.

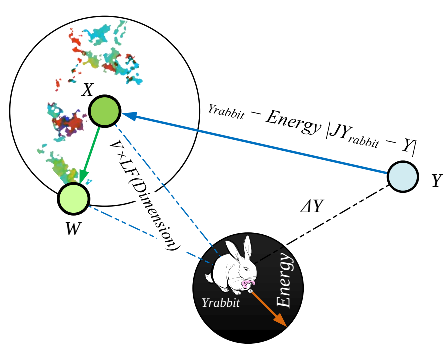


Figure 2: A simple case with one Hawk in hard besiege.

$$Y(t+1) = Y_{\text{rabbit}}(t) - Energy|\Delta Y(t)| \tag{6}$$

Soft besiege with progressive rapid dives

This stage is more intelligent than the previous stages, since the prey has enough energy to escape, and still it helps to make soft besiege for surprise pounce.

Fig. 3 depicts the overall vectors in the situation of soft besiege with successive quick dives, when the location history of LF-based leapfrog movement patterns is stored over several repetitions.

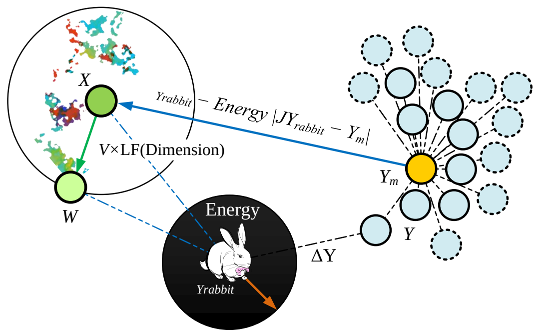


Figure 3: Overall vectors in the situation of soft besiege with successive quick dives.

One of the characters of hawks is that they catch the prey in competitive situations with a selection of accurate positions (Eq. 7).

$$X = Y_{\text{rabbit}}(t) - Energy \mid JY_{\text{rabbit}}(t) - Y(t) \mid \tag{7}$$

Here, possible results are compared with their previous dives to detect the best dive. In case, it is not good, hawks think about catching the prey by escalating with their behaviour of rapid dives, abruptly. By using levy flight (LF) based strategy in Eq. 8, their dive operation performed as:

$$W = X + V \times LF(Dimension) \tag{8}$$

where,

$Dimension$ = the dimension of solutions,

V = A vector of random size $1 \times dim$.

Eq. 9 below depicts the calculation of the Levy flight function:

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}, \sigma = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}} \tag{9}$$

where, u & v are random values and β is a default constant set.

$$Y(t+1) = \begin{cases} X & \text{if } F(X) < F(Y(t)) \\ W & \text{if } F(W) < F(Y(t)) \end{cases} \tag{10}$$

Finally, by using Eq. 10, the Hawks positions are updated with progressive rapid dives where X and W from Eq. 7 and Eq. 8

Hard besiege with progressive rapid dives

This step uses Eq. 11 for the Hawks to reduce their average location distance for reaching near the prey as prey does not have enough energy to escape.

$$Y(t+1) = \begin{cases} X & \text{if } F(X) < F(Y(t)) \\ W & \text{if } F(W) < F(Y(t)) \end{cases} \tag{11}$$

where, X is set as in Eq. 12, and W is updated as in Eq. 13.

$$Y = X_{\text{rabbit}}\,(t) - Energy \mid JY_{\text{rabbit}}(t) - Y_m(t) \mid \tag{12}$$

$$W = X + V \times LF(Dimension) \tag{13}$$

Fig. 4 describes the overall vectors in the situation of Hard Besiege with successive.

A complete HHO algorithm in pseudocode form is depicted in Algorithm 1.

3.2. One-hot encoding

The relationship among the categorical variables from the input datasets is always mismatched. Hence, the classification of those datasets by using the integer encoding technique will lead to producing unexpected prediction outcomes.

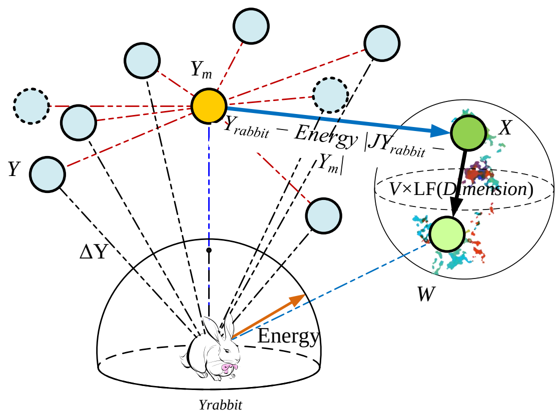


Figure 4: Overall vectors in the situation of hard besiege with successive quick dives.

One-hot encoding is one of the best methods for classifying those unrelated data. It helps to predict the output accurately utilizing adding a binary variable to each unique integer value in the input datasets.

In the HHO example, the range has 2 categorical values like hand closed for gesture and hand open for the gesture. In this example, an integer value of 0 is assigned for hand closed for gesture and hand open for gesture as the integer value of 1. This process is repeated for all the data of hand gestures. Next, using integer encoding, the mapping of binary vector length of 2 has been set for the assigned 2 integer values in the considered datasets, such as 'hand closed for gesture' is labelled with integer value 0, whereas 'hand opened for gesture' is labelled with integer value marked as 1. If our dataset sequence is hand open for gesture, hand closed for gesture, hand open for gesture, we could represent it in integer encoding as,$1, 0, 1$. The same could be represented after pre-processing with one hot encoding as $[1, 0], [0, 1], [1, 0]$.

3.3. Convolutional Neural Networks

CNN is one of the prominent deep learning algorithms for image classification. CNN has an inbuilt ability to learn the characteristics of input images. Hence, the preprocessing of CNN is comparatively lower than the other deep learning algorithms. It can effectively catch the spatial and temporal conditions

Algorithm 1: HHO Algorithm [14].

Inputs: Hyperparameters for CNN.
Outputs: Optimized hyperparameters.
Initialize the population $Y_i (i = 1, 2, \ldots, N)$
while *condition is not being met* **do**
    Determine the fitness rate of hawks.
    Y_{rabbit} is chosen as the best position
    for *every hawk* (Y_i) **do**
        Update the jump capacity J and initial energy $Energy_0$
        $\text{Energy}_0 = 2\text{rand}() - 1, \ \text{J} = 2(1 - \text{rand}())$
        Update $Energy$ using Eq. 3
    if $|Energy| \geq 1$ **then**
        Update the position vector using Eq. 1
    if $|Energy| < 1$ **then**
        if $s \geq 0.5$ *and* $|Energy| \geq 0.5$ **then**
            Update the position vector using Eq. 4
            else if $s \geq 0.5$ *and* $|Energy| < 0.5$ **then**
                Update the position vector using Eq. 6
            else if $s < 0.5$ *and* $|Energy| \geq 0.5$ **then**
                Update the position vector using Eq. 10
            else if $s < 0.5$ *and* $|Energy| < 0.5$ **then**
                Update the position vector using Eq. 11
return Y_{rabbit}

of a picture with the help of applicable filters. Max pooling helps to reimburse the information about the maximum coverage of images by the kernel, and the information about the average value of images covered by the kernel is derived by the average pooling method.

The three layers (convolutional (CL), pooling (PL), and fully connected layers (FCL)) of CNN help in classifying the input images. Here, CL helps to convolve the input images with different hyperparameters, and the result of it is passed to the subsequent layers. CL uses the loss function for regularizing the weight of the image. PL helps in segregating the input images into the group of non-overlapping rectangles. PL uses the RELU function for removing negative values of grouped images. FCL uses the softmax function for classifying the images as depicted in Fig. 5.

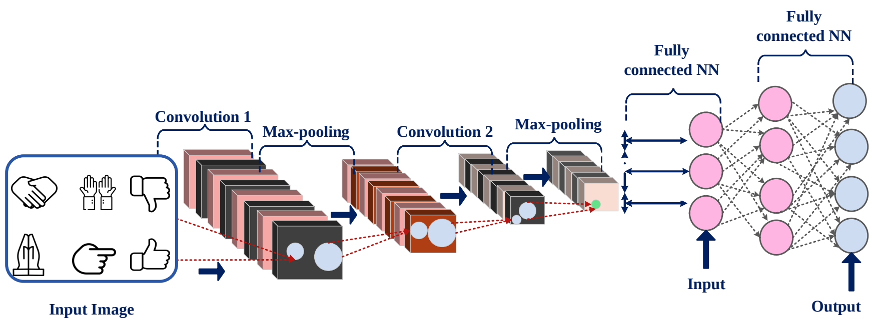


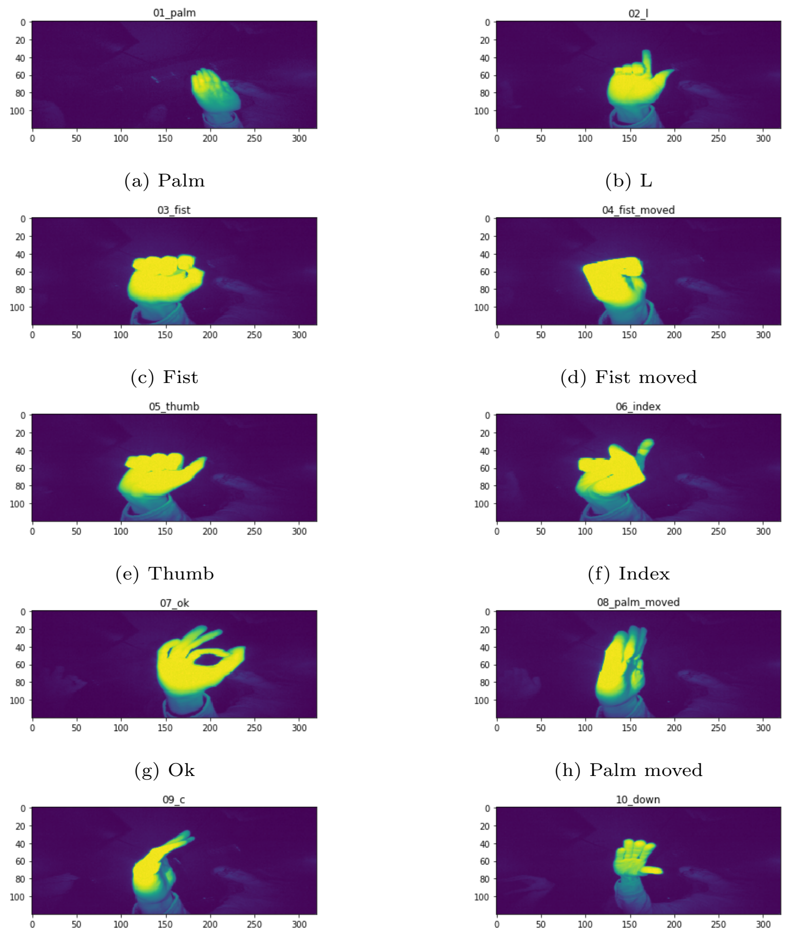
Figure 5: Fully connected CNN.

4. Results and Discussion

The proposed model is investigated on an open-source dataset collected from the Kaggle repository. Google Colab GPU is used in this work, which is based on the cloud framework developed by Google Inc. This framework consists of 25 GB RAM and 50 GB Hard Disk and python 3.7 programming language. The performance evaluation and the dataset description of the proposed system are discussed in the following subsections.

4.1. Experimental Setup

In the proposed work, 80% of the hand gesture image samples are used for training and the remaining 20% of the images are used for testing and validation. Sameples from the dataset are shown in Fig. 6. To select the hyper-parameters



(a) Palm (b) L (c) Fist (d) Fist moved (e) Thumb (f) Index (g) Ok (h) Palm moved (i) C (j) Down

Figure 6: Sample hand gesture images from the dataset.

of CNN, the HHO algorithm is used. The hyper-parameters chosen by HHO for the CNN model are shown in Table. 1.

4.2. Data Set Overview

The hand gesture datasets used for experiments which include 10 different folders of hand gestures like palm, thumb and down, etc, corresponding digits $(0-9)$ are collected from the public repository, Kaggle. Every folder has 2000 different hand gesture images. Some of the sample images from the dataset are

Table 1: HHO selection of hyper-parameters for CNN model.

Hyper - Parameter	**Description**
Number of epochs	4
Number of Dense Layers	2
Number of Convolutional Layers	3
Batch Size	128
Optimizer	Nadam
Loss Function	Categorical based Cross Entropy
Number of Pool Layers	3
Activation Function	Leaky Relu (Intermediate Layer)
Activation Function	Softmax (Output Layer)

shown in Fig. 6.

4.3. Experimental results and the performance evaluation of the proposed model

The CNN model has several parameters such as number of convolution layers, number of dense layers, activation function at each layer, activation function at the output layer, number of epochs, batch size, number of pool layers, loss function, etc. Identification of the right values for these parameters is crucial for the performance of the model. Also, choosing the optimal values reduces the burden on CNN by reducing the training time. To choose the optimal values for these parameters, the HHO algorithm is employed in this work.

Fig. 7 and Fig. 8 depict the experimental results of the proposed model, whereas the results obtained on comparison of the proposed method with the other CNN based models are depicted in Fig. 9, Fig. 10, and Fig. 11.

From Fig. 7, it can be observed that the proposed model has achieved 100% training and testing accuracy after 1 and 2 epochs respectively. Fig. 8 depicts that the loss percentage during training and the testing phases of the proposed model is 0% after 1 and 2 epochs respectively.

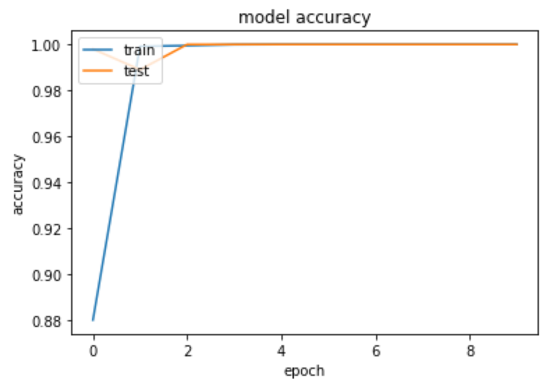


Figure 7: Epochs versus Accuracy.

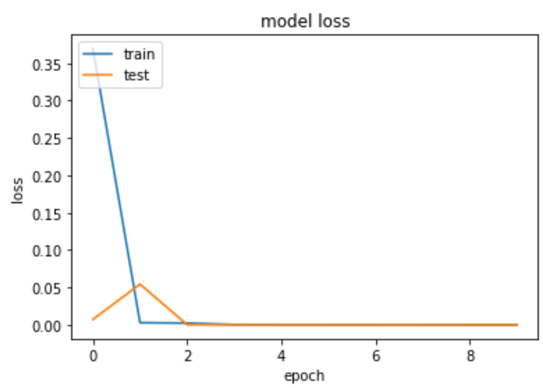


Figure 8: Epochs versus Loss.

The proposed model's (CNN+HHO) performance is compared with CNN integrated with several meta-heuristic algorithms such as whale optimization algorithm (WOA), gravitational search algorithm (GSA), cuckoo search algorithm (CSA), particle swarm optimization (PSO), genetic algorithm (GA), artificial bee colony (ABC), grey-wolf optimization (GWO).

Fig. 9 depicts that the proposed model outperforms the other variants of CNN based on accuracy measures. The training and testing accuracy of the proposed model is 100% each, which is better than the other models considered. The training and testing accuracy of WOA+CNN 97.2% and 98.2% respectively, GWO+CNN is 96.3% and 97.1%, PSO+CNN is 96.7% and 98.2%, respectively, GA+CNN is 96.8% and 97.5%, respectively, GSA+CNN is 97.2% and 98.1% respectively, ABC+CNN is 98.1% and 99.2%, respectively, CSA+CNN is 99.1% and 99.5%, respectively.

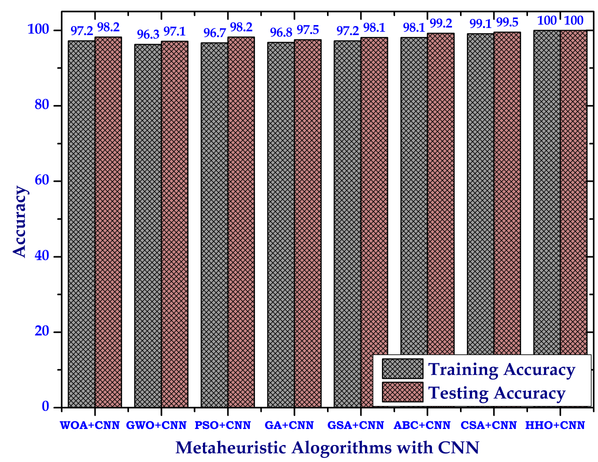


Figure 9: Comparative analysis of the proposed methodology with state-of-the-art models with respect to accuracy.

Fig. 10 depicts that the proposed model outperforms the other variants of CNN based on error rate measure. The proposed model achieved a training and testing error of 0% each, which is better than the other models considered. The training and testing accuracy of WOA+CNN 2.8% and 1.8%, respectively, GWO+CNN is 3.7% and 2.9%, PSO+CNN is 3.3% and 1.8%, respectively, GA+CNN is 3.2% and 2.5%, GSA+CNN is 2.8% and 1.9%, respectively, ABC+CNN is 1.9% and 0.8% respectively, CSA+CNN is 0.9% and 0.5%, respectively.

The comparative analysis of the proposed model on training time with the CNN integrated with other meta-heuristic algorithms considered is depicted in

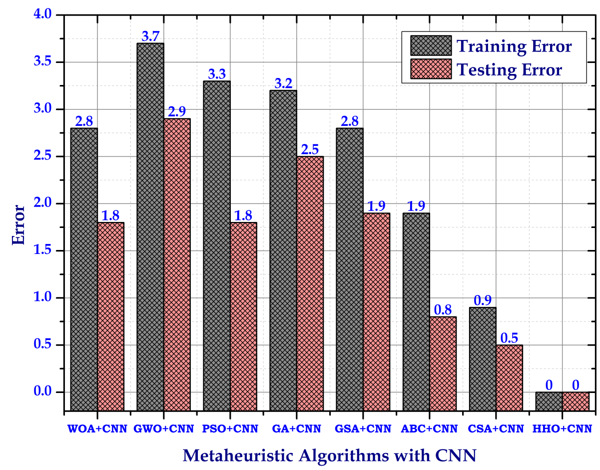


Figure 10: Comparative analysis of the proposed methodology with state-of-the-art models with respect to error rate.

Fig. 11. It can be observed from the figure that the proposed model, requires the least amount of time to train the model when compared with the other variants of CNN. The training time of the proposed model on the hand gesture dataset is 13 minutes, CSA+CNN is trained in 16 minutes, ABC+CNN is trained in 22 minutes, GSA+CNN is trained in 25 minutes, GA+CNN is trained in 26 minutes, PSO+CNN is trained in 24 minutes, GWO+CNN is trained in 23 minutes, and WOA+CNN is trained in 21 minutes respectively.

Table 2: Performance analysis of the proposed methodology with state-of-the-art models.

Metaheuristic Algorithm + CNN	Training and Testing Error		Training and Testing Accuracy		Training Time
	Training Error	Testing Error	Training Accuracy	Testing Accuracy	(Minutes)
WOA+CNN	2.8	1.8	97.2	98.2	21
GWO+CNN	3.7	2.9	96.3	97.1	23
PSO+CNN	3.3	1.8	96.7	98.2	24
GA+CNN	3.2	2.5	96.8	97.5	26
GSA+CNN	2.8	1.9	97.2	98.1	25
ABC+CNN	1.9	0.8	98.1	99.2	22
CSA+CNN	0.9	0.5	99.1	99.5	16
HHO+CNN	0	0	100	100	13

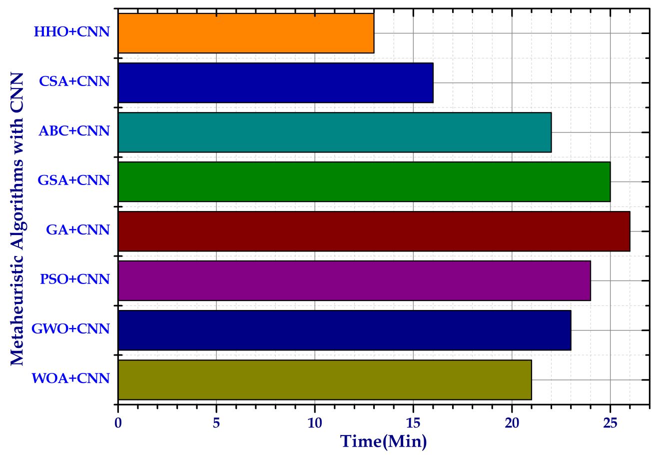


Figure 11: Comparative analysis of the proposed methodology with state-of-the-art models with respect to the training time.

Table. 2 summarizes the comparative analysis of the performance of the proposed model with the other CNN-based models. Table. 3 depicts the comparative analysis of the proposed work with recent works on hand gesture recognition. From the table, it can be observed that the proposed method outperforms the existing works in [15] in terms of accuracy and loss. Even though [16] attained an accuracy of 100% and a loss of 0%, the training time is 15 minutes. The proposed method performs better than the work in [16] by attaining a training time of 13 minutes.

Table 3: Comparison with existing works.

Ref. No	**Accuracy**	**Loss**	**Training Time (Min)**
[15]	98	2	-
[16]	100	0	15
Proposed	100	0	13

4.4. Discussion

From the result analysis, it is evident that the proposed HHO+CNN model attains better results when compared with the other variants of CNN. The HHO+CNN model is proved to be superior to the other models to classify the hand gesture dataset concerning accuracy and loss measures, as well as it is proved to be trained in a less amount of time. The HHO chooses optimal parameters of the CNN due to its fast convergence rate and not getting trapped in the local optima. Compared to the existing state-of-the-art, the proposed model is trained in less amount of time as the convergence rate of the HHO is better than the existing models.

The following objectives are met by the proposed HHO-CNN model in this work.

- Optimal hyper-parameter tuning of the CNN parameters is achieved.
- The proposed model achieved 100% accuracy and 0% error rate.
- The burden on the model is reduced due to tuning of the hyper-parameters and hence the training time of the proposed model is optimized.

5. Conclusion and Future Work

Tuning of the hyper-parameters plays a very important role in optimizing the training of CNN on images. Nature-inspired metaheuristic algorithms can be used effectively to choose the optimal parameters of the CNN that can enhance the performance of the CNN and also reduce the training time of the CNN. In this work, the HHO algorithm is used for the hyper-parameter tuning of CNN to classify the hand gesture images. In this work, firstly the hand gesture dataset is gathered from the Kaggle repository. Then, the images are pre-processed using the one-hot encoding method. The preprocessed images are then fed to the proposed HHO-CNN model to classify the hand gesture images. The proposed HHO-CNN achieved an accuracy of 100% and an error rate of 0%. The experimental results prove the superiority of the proposed model over the other CNN-based models. In the future, the proposed model can be enhanced to classify hand gestures in real-time.

References

[1] S. Mitra, T. Acharya, Gesture recognition: A survey, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 37 (3) (2007) 311–324.

[2] A. R. Sarkar, G. Sanyal, S. Majumder, Hand gesture recognition systems: a survey, International Journal of Computer Applications 71 (15).

[3] S. S. Rautaray, A. Agrawal, Vision based hand gesture recognition for human computer interaction: A survey, Artificial intelligence review 43 (1) (2015) 1–54.

[4] J. P. Wachs, M. Kölsch, H. Stern, Y. Edan, Vision-based hand-gesture applications, Communications of the ACM 54 (2) (2011) 60–71.

[5] A. Chaudhary, J. L. Raheja, K. Das, S. Raheja, Intelligent approaches to interact with machines using hand gesture recognition in natural way: a survey, arXiv preprint arXiv:1303.2292.

[6] N. A. Ibraheem, R. Z. Khan, Survey on various gesture recognition technologies and techniques, International journal of computer applications 50 (7).

[7] X. Y. Wu, A hand gesture recognition algorithm based on dc-cnn, Multimedia Tools and Applications (2019) 1–13.

[8] S. Tam, M. Boukadoum, A. Campeau-Lecours, B. Gosselin, A fully embedded adaptive real-time hand gesture classifier leveraging hd-semg and deep learning, IEEE Transactions on Biomedical Circuits and Systems 14 (2) (2019) 232–243.

[9] S. M. Aslam, S. Samreen, Gesture recognition algorithm for visually blind touch interaction optimization using crow search method, IEEE Access 8 (2020) 127560–127568.

[10] J.-H. Sun, T.-T. Ji, S.-B. Zhang, J.-K. Yang, G.-R. Ji, Research on the hand gesture recognition based on deep learning, in: 2018 12th International Symposium on Antennas, Propagation and EM Theory (ISAPE), IEEE, 2018, pp. 1–4.

[11] C. Du, L. Zhang, X. Sun, J. Wang, J. Sheng, Enhanced multi-channel feature synthesis for hand gesture recognition based on cnn with a channel and spatial attention mechanism, IEEE Access 8 (2020) 144610–144620.

[12] A. D. Berenguer, M. C. Oveneke, M. Alioscha-Perez, A. Bourdoux, H. Sahli, et al., Gesturevlad: Combining unsupervised features representation and spatio-temporal aggregation for doppler-radar gesture recognition, IEEE Access 7 (2019) 137122–137135.

[13] C.-h. Yoo, S.-w. Ji, Y.-g. Shin, S.-w. Kim, S.-j. Ko, Fast and accurate 3d hand pose estimation via recurrent neural network for capturing hand articulations, arXiv preprint arXiv:1911.07424.

[14] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, Future Generation Computer Systems 97 (2019) 849–872.

[15] R. Ramya, K. Srinivasan, Real time palm and finger detection for gesture recognition using convolution neural network, in: Human Behaviour Analysis Using Intelligent Systems, Springer, 2020, pp. 1–19.

[16] T. R. Gadekallu, M. Alazab, R. Kaluri, P. K. R. Maddikunta, S. Bhattacharya, K. Lakshmanna, M. Parimala, Hand gesture classification using a novel cnn-crow search algorithm, Complex & Intelligent Systems (2021) 1–14.

Thippa Reddy Gadekallu (Senior Member IEEE) is currently working as Associate Professor in School of Information Technology and Engineering, VIT, Vellore, Tamil Nadu, India. He has more than 14 years of experience in teaching. He has published more than 100 international/national publications. Currently, his areas of research include Machine Learning, Internet of Things, Deep Neural Networks, Blockchain, Computer Vision.

Gautam Srivastava (Senior Member, IEEE) received the B.Sc. degree from Briar Cliff University, USA, in 2004, and the M.Sc. and Ph.D. degrees from the University of Victoria, Canada, in 2006 and 2011, respectively. He is currently working as Associate Professor, Brandon University, Canada. His area of interest is data mining, big data, security.

Madhusanka Liyanage is currently an Assistant Professor and Director of Graduate Research at the School of Computer Science, University College Dublin, Ireland. He is also acting as a Docent/Adjunct Professor at the University of Oulu, Finland, and Honorary Adjunct Professor of the University of Ruhuna, Sri Lanka. His research interests includes 5G/6G, Blockchain, Network security, Network Slicing and Machine Learning.

Iyapparaja M is an Associate Professor in the School of Information Technology & Engineering at VIT Vellore, He received a B.E. (CSE),M.E. (CSE), and Ph.D. from Anna University Chennai. His research interests are federated learning, machine learning, and cyber security.

Chiranji Lal Chowdhary is an Associate Professor in the School of Information Technology & Engineering at VIT Vellore, where he has been since 2010. He received a B.E. (CSE) from MBMEC Jodhpur, and M. Tech. (CSE) from the MSRIT, Bangalore. He received his Ph.D. from the VIT University Vellore. His research interests span both computer vision and image processing.

Srinivas Koppu working as Associate Professor in SITE, VIT, India. He received the Phd, M. Tech and B. Tech (CSE) from, VIT, IIIT Allahabad, JNTU Hyder- abad, respectively, India. His Research interest also includes Robotics, IoT, Medical Image Processing.

Praveen Kumar Reddy Maddikunta received B.Tech in CSE from JNT University, M.Tech and Phd in CSE from VIT, Vellore, India. He is currently working as an Assistant Professor Senior in SITE, VIT. His Research interest also includes IoT, ML.