Intelligent human action recognition using an ensemble model of evolving deep networks with swarm-based optimization.

ZHANG, L., LIM, C.P. and YU, Y.

2021

Intelligent Human Action Recognition Using an Ensemble Model of Evolving Deep Networks with Swarm-based Optimization

Li Zhang[1], Chee Peng Lim[2], and Yonghong Yu[3]

[1]Computational Intelligence Research Group
Department of Computer and Information Sciences
Faculty of Engineering and Environment
University of Northumbria
Newcastle, NE1 8ST, UK

[2]Institute for Intelligent Systems Research and Innovation
Deakin University
Waurn Ponds, VIC 3216, Australia

[3]College of Tongda
Nanjing University of Posts and Telecommunications
Nanjing, China

Email: li.zhang@northumbria.ac.uk; chee.lim@deakin.edu.au; yuyh@njupt.edu.cn

Abstract.
Automatic interpretation of human actions from realistic videos attracts increasing research attention owing to its growing demand in real-world deployments such as biometrics, intelligent robotics, and surveillance. In this research, we propose an ensemble model of evolving deep networks comprising Convolutional Neural Networks (CNNs) and bidirectional Long Short-Term Memory (BLSTM) networks for human action recognition. A swarm intelligence (SI)-based algorithm is also proposed for identifying the optimal hyper-parameters of the deep networks. The SI algorithm plays a crucial role for determining the BLSTM network and learning configurations such as the learning and dropout rates and the number of hidden neurons, in order to establish effective deep features that accurately represent the temporal dynamics of human actions. The proposed SI algorithm incorporates hybrid crossover operators implemented by sine, cosine, and tanh functions for multiple elite offspring signal generation, as well as geometric search coefficients extracted from a three-dimensional super-ellipse surface. Moreover, it employs a versatile search process led by the yielded promising offspring solutions to overcome stagnation. Diverse CNN-BLSTM networks with distinctive hyper-parameter settings are devised. An ensemble model is subsequently constructed by aggregating a set of three optimized CNN-BLSTM networks based on the average prediction probabilities. Evaluated using several publicly available human action data sets, our evolving ensemble deep networks illustrate statistically significant superiority over those with default and optimal settings identified by other search methods. The proposed SI algorithm also shows great superiority over several other methods for solving diverse high-dimensional unimodal and multimodal optimization functions with artificial landscapes.

Keywords: Swarm Intelligence, Evolutionary Algorithm, Deep Hybrid Neural Network, Ensemble Classifier, and Human Action Recognition.

1. INTRODUCTION

Automatic understanding of human behaviours from video contents is a challenging task owing to dynamic background, viewpoint variations, rotations, camera motion, as well as small inter-class and large intra-class variations. Effective video encoding and representation methods play an important role in informing human action classification. The Convolutional Neural Networks (CNNs) show great superiority in deep feature

learning for image classification tasks. In addition, the Long Short-Term Memory (LSTM) networks are capable of extracting temporal dynamics in video sequences [1, 2]. As a variant of LSTM, the bidirectional LSTM (BLSTM) network is equipped with two hidden LSTM layers of opposite directions for sequential analysis. In comparison with that of LSTM, the learning mechanism of BLSTM provides additional context to the network by adopting both the past and future states simultaneously to facilitate time series prediction. Therefore, the combination of BLSTM and CNN is employed in this research, in view of their enhanced capabilities in video representation. However, the identification of optimal network hyper-parameters such as the number of hidden neurons, and the learning and dropout rates, is vital in determining network capabilities in extracting complex sequential patterns. This is a bottleneck in deploying BLSTM models to a new domain. As a result, a swarm intelligence (SI)-based algorithm is proposed in this research to optimize the model and learning hyper-parameters of BLSTM networks.

Specifically, we propose an ensemble of evolving CNN-BLSTM networks with optimal hyper-parameter identification for human action classification. In order to extract effective discriminative features, an ImageNet pre-trained GoogLeNet model is used for deep feature extraction from video frames. A BLSTM network is then employed to learn the temporal sequential patterns of the extracted deep frame features. The proposed SI algorithm is subsequently used to identify the optimal settings of the learning and dropout rates, as well as the number of hidden neurons in the BLSTM networks. The proposed SI algorithm incorporates hybrid crossover operators based on sine, cosine and tanh functions to generate promising offspring leaders. A parametric hyper-plane surface for search coefficient assignment is also established. A versatile search process led by the yielded promising offspring solutions, the mean vector of these elite signals as well as the swarm leader is exploited to overcome stagnation. Diverse BLSTM networks with distinctive optimal hyper-parameter settings are devised by the proposed algorithm. An ensemble model is subsequently constructed by incorporating a set of three optimized CNN-BLSTM networks, in which the average of their prediction probabilities is exploited to reach the final prediction. Owing to the employment of dynamic crossover operators and three-dimensional (3D) super-ellipse search coefficients, the proposed SI algorithm provides significant capabilities in devising the ensemble CNN-BLSTM networks with optimal settings for video action classification. Figure 1 illustrates the system architecture.

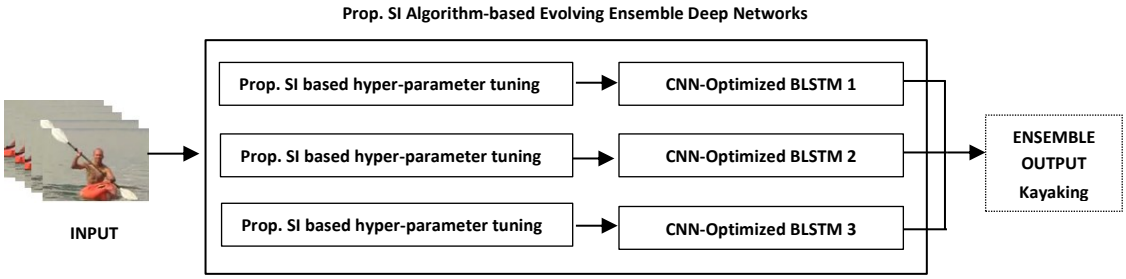

Figure 1 The overall system architecture

The contributions of this research are summarized as follows.

- We propose an ensemble of evolving CNN-BLSTM networks for human action classification. In addition, a new SI algorithm is devised for optimal hyper-parameter selection. This is because the network structure and learning configurations, such as the learning and dropout rates and the number of hidden neurons, are important for extracting temporal dynamics from the deep features established in a BLSTM network.
- The proposed SI algorithm incorporates hybrid crossover operators implemented by sine, cosine and tanh formulae for multiple elite offspring signal generation, as well as search coefficients extracted from a 3D super-ellipse eigenvector. Specifically, two remote swarm leaders are identified at the initial stage of the search process. Crossover operations based on sine, cosine and tanh functions are subsequently conducted to yield multiple elite offspring signals with the above two remote swarm leaders as the parent chromosomes. The resulting elite solutions are, therefore, capable of exploring the search space between the two remote leaders in different directions and scales, in order to increase search diversity. In comparison with single leader guided monotonous search operations, the proposed search process is a versatile one, where the promising offspring leaders, the mean vector of these elite signals, and the global best solution are randomly selected to overcome stagnation during the search process. Moreover, instead of using uniform or Gaussian distributions, the proposed model uses a 3D geometric hyper-plane surface for search coefficient generation, in order to accelerate convergence.

The empirical results indicate the statistical superiority of the proposed SI algorithm over other baseline methods for optimal hyper-parameter identification in BLSTM networks.

- To overcome the bias issue of individual classifiers, we construct an ensemble model consisting of three optimized CNN-BLSTM networks by averaging their prediction probabilities. Since the base CNN-BLSTM networks have different optimized hyper-parameter settings, they possess great diversity and flexibility to enhance the classification performance through the ensemble approach.
- Evaluated using several action data sets such as KTH, UCF50 and UCF101, the proposed SI algorithm indicates significant capabilities in identifying the optimal model settings for temporal sequential information extraction in BLSTM networks. Our evolving deep ensemble CNN-BLSTM networks illustrate statistically significant superiority over those with default and optimal settings identified by other classical and advanced search methods. The proposed SI algorithm also outperforms a number of classical and advanced search methods significantly for solving diverse high-dimensional unimodal and multimodal benchmark functions.

The paper is organized as follows. Section 2 introduces classical search algorithms and their advanced variants for solving diverse optimization problems, as well as the related state-of-the-art studies on human action recognition. The proposed SI algorithm with adaptive crossover operators and super-ellipse coefficients is presented in Section 3. Section 4 introduces hyper-parameter identification for formulating the ensemble deep networks. A comprehensive evaluation is provided in Section 5. The concluding remarks and future research directions are presented in Section 6.

2. RELATED WORK

In this section, we introduce classical search methods and their advanced variants for solving diverse optimization problems. State-of-the-art related studies on human action recognition are also discussed.

2.1 Evolutionary Algorithms and Their Variant Models

Owing to the capability of achieving global optimality, evolutionary algorithms are effective for solving diverse optimization problems, such as feature selection [3-6], job scheduling [7], cluster centroid improvement [8, 9], parameter tuning [10], and deep architecture generation [11, 12]. Sundaramurthy and Jayavel [4] proposed a hybrid model incorporating Particle Swarm Optimization (PSO) with Grey Wolf Optimization (GWO) for discriminative feature selection, while the C4.5 Decision Tree (DT) was used for Rheumatoid Arthritis (RA) classification. The PSO method was responsible for the initial search of the optimal feature subsets, while the GWO model carried out a subsequent more refined feature selection procedure. In comparison with feature optimization using individual PSO and GWO models, the proposed hybrid model achieved a better RA classification performance. Shao et al. [13] proposed a PSO variant for optimal hyper-parameter identification for the LSTM networks in nickel price prediction. Adaptive acceleration coefficients based on sine functions, as well as an adaptive decreasing inertia weight, were proposed to increase search diversification. Their PSO variant optimized the time step and the numbers of neurons in two LSTM layers for the regression analysis. The yielded LSTM model outperformed a manually-designed LSTM network, AutoRegressive Integrated Moving Average (ARIMA), and the PSO-devised LSTM network. Chitradevi and Prabha [14] employed PSO, Cuckoo Search (CS), GA, and GWO for the segmentation of brain sub-regions, i.e. White Matter (GM), Corpus Callosum (CC), Grey Matter (WM), and Hippocampus (HC). These segmented regions were used as the inputs of a deep CNN model for Alzheimer disease (AD) diagnosis. The empirical results indicated that the multiple-leader guided GWO method identified the most effective sub-regions and illustrated a better performance pertaining to image segmentation in comparison with other search methods.

Sun et al. [12] employed the Genetic Algorithm (GA) for deep CNN architecture generation. Statistical measures and a variable-length gene encoding strategy were used to encode the initialization weights and network architectures, respectively. The Gaussian distribution was used to decode the network weights based on the statistical measures. The GA operators, including a slacked binary tournament selection and different types of mutation operations, were applied for offspring generation. Their model outperformed several state-of-the-art manually designed networks in performance evaluation using various data sets, including the Rectangle Images, MNIST and its variants. A performance predictor based on Random Forest (RF) was also developed by Sun et al. [15] to alleviate the heavy computational cost of evaluating deep CNNs in deep architecture generation. Deep networks with ResNet and DenseNet blocks were produced using a GA operator. The obtained network configurations and their corresponding accuracy rates were used for RF training. Each base model of RF was trained using a feature subspace. Good results with improved computational costs were achieved in large-scale image classification tasks.

An enhanced chaotic PSO (CPSO) method was proposed by Zhang and Xin [16] for the identification of the optimal number of hidden neurons in LSTM for short-term traffic flow prediction. A pre-processing procedure based on a classification algorithm, namely Asym-Gentle Adaboost with Cost-sensitive Support Vector Machine (AGACS), was firstly performed to separate outliers from normal samples. The identified normal samples were subsequently used for hyper-parameter fine-tuning using CPSO in combination with LSTM. Moreover, a logistic map was used in the CPSO model for both population initialization and chaotic perturbation, in order to accelerate convergence and overcome stagnation. The CPSO-devised LSTM model achieved a better performance in traffic flow prediction, as compared with those from the Deep Belief Network (DBN), Recurrent Neural Network (RNN) and CNN models. Tsai et al. [17] conducted optimal network topology identification of the Multiple-layer Perceptron (MLP) using Simulated Annealing (SA) for the prediction of the number of bus passengers. The SA model was used to optimize the numbers of neurons of each of the three hidden layers in the MLP network. The SA-based MLP model outperformed other regression methods, including Support Vector Regressor (SVR), RF, and eXtreme gradient boosting (XGBoost) models in passenger number prediction tasks.

A cosine-annealing PSO (COSPSO) model was proposed by Fielding and Zhang [11] to generate deep CNN architectures for image classification. The search parameters were yielded by exploiting the cosine functions. The PSO variant was used to identify the optimal numbers of five distinctive types of convolutional blocks in a CNN model. The optimized network achieved superior results with respect to the CIFAR-10 and CIFAR-100 data sets. Emotion recognition based on electroencephalogram (EEG) and blood volume pulse (BVP) signals was performed by Nakisa et al. [18]. Differential evolution (DE) was used to optimize the LSTM hyper-parameters, including the batch size and number of hidden neurons. The DE-optimized LSTM model outperformed those yielded by PSO, random search, Tree-of-Parzen-Estimators (TPE), and SA, in emotion identification. Two PSO variants with adaptive and random acceleration coefficients based on non-linear circle, sine, and spiral functions were proposed by Tan et al. [10]. The models were embedded into the MLP for discriminative feature selection as well as hyper-parameter (i.e. learning rate and regularization strength) fine-tuning pertaining to a deep CNN model for skin cancer detection. Besides the adoption of adaptive and random coefficients, their PSO variants embedded sub-dimension-based search, multiple global promising solutions, and scattering procedures to overcome stagnation. Their models outperformed a number of advanced PSO and Firefly Algorithm (FA) variants in discriminative feature selection and hyper-parameter identification for melanoma classification. Yan et al. [19] used the LSTM networks for missing value prediction in data samples of water quality. The PSO algorithm was used to optimize the learning rate and the time step of the LSTM network. The yielded LSTM model outperformed other regressors, i.e. a manually-crafted LSTM, Bayesian Network, DT and Backpropagation Neural Network (BPNN), in missing value prediction.

Kang et al. [20] developed a non-inertial opposition-based PSO algorithm (NOPSO) for undertaking various unimodal and multimodal benchmark functions. A rotationally invariant semi-autonomous PSO model with directional diversity was proposed by Santos et al. [21] for solving the CEC2017 artificial landscapes. Another PSO variant, namely dynamic multi-swarm differential learning PSO, was proposed by Chen et al. [22]. It outperformed several classical search methods (such as DE) and PSO variants (such as Comprehensive Learning PSO) for solving 41 numerical benchmark functions. There are other studies in the literature pertaining to deep architecture generation and hyper-parameter fine-tuning using PSO and GA methods, such as Zhang and Lim [23], Sun et al. [24], Nayak et al. [25], and Tan et al. [26, 27].

2.2 Human Action Recognition

A variety of state-of-the-art studies have been proposed for human action recognition, owing to the growing demand in real-world deployments, e.g. security and surveillance. A two-stream architecture was proposed by Simonyan and Zisserman [28]. It comprised two separate CNNs, i.e. spatial and temporal CNNs, for appearance and motion-based action recognition, respectively. Specifically, the individual video frames and multi-frame optical flow were used as the inputs of the two-stream CNNs, respectively. Several variations of the optical flow inputs, such as optical flow and trajectory stackings, were explored. The fusion of both networks was performed based on the softmax scores at the classification level using either averaging or a linear SVM. The SVM ensemble strategy performed the best in several human action data sets. Wang et al. [29] employed a spatio-temporal pyramid network to fuse and reinforce the extracted spatial and temporal cues for human action recognition. Hierarchical fusion strategies in conjunction with a spatio-temporal compact bilinear operator embedded with attention mechanisms were proposed to effectively capture element-wise spatio-temporal interactions and attended features. The video-level representation was formed by aggregating the features from spatial, temporal, and attention streams. The model produced superior performances in several human action data sets. Owing to the expensive computational cost in estimating optical flow, Zhang et al. [30] performed real-time human action recognition using motion vectors, which can be extracted from video clips with a significantly lower cost. A deep learning scheme consisting of two modules was proposed, i.e. (1) video encoder

for image and motion vector extraction, and (2) a two-stream architecture as in [28], where the motion vector instead of optical flow was used as the input of a temporal CNN model. To overcome imprecision and noise in the motion vectors, four transfer learning mechanisms, i.e. initialization transfer, supervision transfer and their combination, as well as deeply connected transfer, were proposed. The aim was to enhance the training process of the motion vector CNN based on the refined and subtle motion details extracted by the optical flow CNN. The model achieved promising performances, and it was significantly faster than those of many existing optical flow-based methods.

Ullah et al. [31] conducted human action recognition using bi-directional LSTM in conjunction with AlexNet. The ImageNet pre-trained AlexNet was used to extract the deep features from every sixth frame of the video clips. Then, a BLSTM model was used to extract the temporal sequential information from the frame features. The BLSTM model was constructed by stacking two LSTM layers with respect to the forward and backward passes, respectively. The model showed great superiority over other methods in human action recognition using HMDB51, UCF101, and YouTube 11 data sets. Singh et al. [32] proposed a multi-stream CNN model incorporated with a BLSTM network for fine-grained human action detection. In comparison with action recognition where a single label is assigned to a video sequence, this human action detection method required the assignment of each label pertaining to each frame. Their work employed motion and appearance CNNs with VGG architectures dedicated to the full and cropped person-centric frames, respectively, for feature extraction. A fully-connected projection layer was used to yield a joint representation based on the outputs of the above multi-stream CNNs. The long-term temporal dynamics of the outputs of the projection layer were analysed using the BLSTM network for the subsequent human action prediction. Evaluated using the MPII Cooking 2 and the MERL Shopping data sets, their model showed enhanced performances.

Dai et al. [33] proposed a two-stream LSTM network with visual attention mechanisms for human action recognition. Specifically, their end-to-end deep network comprised both 'temporal' and 'spatial-temporal' feature streams. The 'temporal' stream employed a temporal attention mechanism to identify the visual saliency in each optical flow image, while in the 'spatial-temporal' stream, the convolutional layers integrated with LSTM were employed to capture the spatial-temporal dynamics. In the latter, a spatial-temporal attention model was embedded, in order to allocate distinctive attention weights to the outputs of each extracted feature map. A joint optimization layer was used to fuse the attention feature vectors from the two streams. The combined losses from both streams and a third stream with the classifier trained using the fused feature vector were calculated to produce the final classification. Evaluated using UCF Sports, UCF11 and jHMDB data sets, this two-stream LSTM deep network outperformed other state-of-the-art methods. Khan et al. [34] performed human action recognition using feature fusion and selection based on hand-crafted histogram of oriented gradients (HoG) and deep CNN features. Pre-processing methods such as a 3D median filter and HSV colour transformation were firstly applied to increase the image contrast. A motion-based saliency detection method, consisting of motion and geometric feature extraction and Chi2 distance calculation, was subsequently applied to identify the region of interest (ROI). The ROI was used as the input for hand-crafted HOG feature extraction, while the deep features of the overall frame were extracted using AlexNet. Feature fusion was conducted based on both extracted feature vectors. The entropy-based feature selection was applied to identify the most discriminative features from the fused vector, while a multi-class SVM (M-SVM) was adopted for video classification. Their method achieved high classification accuracy rates as compared with those of existing studies in several human action data sets.

Li et al. [35] proposed a VideoLSTM model for human action recognition. A motion-based attention scheme and convolutions in the soft-attention LSTM network were used for action localization and classification. Yang et al. [36] employed the attention-again model for human action classification. A CNN was used as the encoder for spatial feature extraction, while a LSTM network with two hidden layers and soft attention mechanisms was used as the decoder. da Silva and Marana [37] conducted human action recognition based on the extraction and encoding of 2D poses. The OpenPose framework was used to extract 2D poses in each frame. Then the poses were projected into a proposed straight line parameter space, where the spatio-temporal features such as trajectories and angles were extracted using a Bag-of-Poses mechanism. The model achieved enhanced performances in the evaluation of the KTH and Weizmann data sets. A biologically inspired visual mechanism with a dual fast and slow feature interaction was proposed by Yousefi and Loo [38] for human action recognition. Yu et al. [39] conducted video-based human action classification using Hierarchical Generative Adversarial Networks (HiGAN). Their model employed both low-level and high-level conditional GAN models to transfer knowledge learned from the image domain to the video domain by constructing a domain-invariant feature representation. The experimental studies indicated that the image classifiers trained with such domain-invariant features showed impressive performance for video classification. Other state-of-the-art studies in human action recognition include a deep 3D Residual ConvNet (Res3D) [40] and compressed video action

recognition (CoViAR) [41]. Comprehensive reviews on action classification are also provided by Jegham et al. [1], Rodríguez-Moreno et al. [2], Singh et al. [42], and Zhang et al. [43].

3. THE PROPOSED SI ALGORITHM

In this research, we propose an evolving ensemble deep learning model that consists of multiple CNN-BLSTM networks for human action recognition. Identifying the optimal model hyper-parameters, such as the learning and dropout rates as well as the number of hidden neurons, is crucial in ensuring the capability of extracting effective sequential patterns. Since this is a bottleneck in deploying BLSTM models to a new domain, a hybrid learning SI algorithm is proposed to optimize the relevant hyper-parameters. Specifically, the proposed SI algorithm incorporates diverse elite signals yielded by nonlinear functions and search coefficients generated using a super formula to increase search diversity. Hybrid learning based on distinctive adaptive formulae between two remote swarm leaders is performed to produce promising offspring solutions for expanding the search territory in different scales and directions. In comparison with single leader guided monotonous operations in classical methods, the proposed SI algorithm employs multiple position updating operations led by diverse elite indicators, as an attempt to overcome stagnation. Algorithm 1 illustrates the pseudo-code of the proposed SI algorithm.

After the population initialization step, the global best solution and a remote second leader with a competitive fitness score and low position proximity are identified. A set of adaptive sine, cosine and tanh oriented functions is used to produce the crossover factors for hybrid offspring leader generation. Specifically, this hybrid leader generation process employs two strategies, i.e. random and adaptive crossover operations, with the abovementioned two remote swarm leaders as the parent chromosomes. The random and adaptive operations are selected randomly in each dimension for generating the promising offspring signals. Therefore, these resulting elite indicators are capable of exploring the search space between the two remote swarm leaders in a dynamic and nonlinear manner to increase search diversity. Subsequently, the algorithm randomly selects any of the yielded hybrid offspring leaders, the mean vector of these promising indicators, as well as the global best solution, for position updating. Instead of using uniform or Gaussian distributions, a super-ellipse parametric hyper-plane surface is employed for search coefficient generation, in order to accelerate convergence. We introduce each key proposed search mechanism comprehensively in the following sub-sections.

Algorithm 1: Pseudo-Code of the Proposed SI Algorithm

Start
Initialize a swarm randomly;
Evaluate the population;
Sort the swarm based on fitness values and identify $gBest$;
Calculate a 3D parametric surface using Equations (2)-(8);
While (Stopping criterion is not satisfied)
{
    Identify a second leader with a similar fitness score but low position correlation to g_{best};
    For each individual **do** {
        Generate the search coefficient using the above yielded 3D parametric surface;
        Generate the offspring leader 1 using the dimension-based operation in Equation (9);
        Generate the offspring leader 2 using the dimension-based operation in Equation (11);
        Generate the offspring leader 3 using the dimension-based operation in Equation (13);
        Calculate the mean position of the above three offspring leaders;
        Randomly select one of the following operations for position updating;
        1. Conduct position updating as defined in Equation (1) using g_{best};
        2. Conduct position updating as defined in Equation (10) using offspring leader 1;
        3. Conduct position updating as defined in Equation (12) using offspring leader 2;
        4. Conduct position updating as defined in Equation (14) using offspring leader 3;
        5. Conduct position updating as defined in Equation (15) using the mean position of the three offspring leaders;
    } **End For**
    Sort the overall swarm based on the fitness values and identify $gBest$;
} **Until** (Stagnation)
Output $gBest$;
End

3.1 The Proposed Search Operation following a Geometric Parametric Surface

In this research, we propose a new position updating strategy to guide the search process. The particle movement follows a 3D nature-inspired parametric surface for performing global exploration. Equation (1) defines the proposed position updating operation.

$$x_i^{t+1} = x_i^t + h_i^t \times (g_{best} - x_i^t) \tag{1}$$

where x_i^{t+1} and x_i^t represent the individual x_i in the $(t+1)$-th and the t-th iterations, respectively. The swarm leader g_{best} is obtained after fitness evaluation of the overall swarm, while h_i^t denotes the search coefficient extracted from a parametric surface simulating curves and shapes observed in nature. Equations (2)-(8) define the generated geometric hyperplane surface [44].

$$\delta(u) = |cos(\tfrac{7u}{4})|^8 + |sin(\tfrac{7u}{4})|^4 \qquad u = [-\pi: 0.05: \pi] \tag{2}$$

$$\gamma(v) = |cos(\tfrac{7v}{4})|^8 + |sin(\tfrac{7v}{4})|^4 \qquad v = [-\pi/2: 0.05: \pi/2] \tag{3}$$

$$\varepsilon = |\delta|^{-\frac{1}{2}} \tag{4}$$

$$\sigma = |\gamma|^{-\frac{1}{2}} \tag{5}$$

$$x = c \times \varepsilon \times \sigma \times cos(u) \times cos(v) \tag{6}$$

$$y = c \times \varepsilon \times \sigma \times \sin(u) \times cos(v) \tag{7}$$

$$z = c \times \sigma \times \sin(v) \tag{8}$$

where ε and σ denote the radiuses, while x, y, and z represent the coordinates of the resulting 3D geometric hyper-plane. The growing factor, c, determines the scale of the produced hyper-plane surface. Based on trial-and-error, we assign $c = 2$ in this study. Figure 2 illustrates the parametric surface defined by Equations (2)-(8).

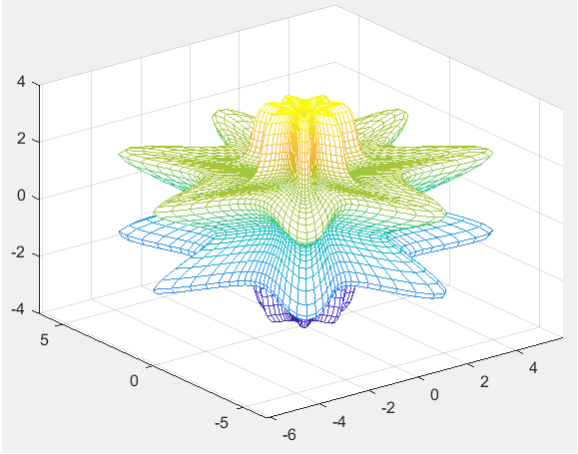

Figure 2 The parametric surface produced using Equations (2)-(8)

As indicated in Figure 2, the generated curve surface depicts a stretching geometric shape. The ranges of x, y and z dimensions are [-6.1476, 5.9029], [-6.2833, 6.2946], and [-3.6465, 3.6307], respectively. In each iteration, we randomly select a point from the generated 3D surface in Figure 2 for each individual. The maximum value among the x, y, and z coordinates of this randomly selected point is used as the search coefficient, h_i^t, in Equation (1). The aim is to enable a comparatively larger search step towards the g_{best} solution, therefore accelerating convergence while following the produced 'flower-like' geometric shape for position updating. The growing size of the generated surface is determined by parameter c, which can affect the size of the search step in position updating. Since we assign a distinctive but comparatively large search coefficient produced by the abovementioned process for each individual, the proposed operation is able to entail different search scales and directions towards the g_{best} solution, therefore diversifying the search process.

3.2 Crossover Operations for Hybrid Leader Generation

After the identification of the global best solution, we extract a second swarm leader that has a competitive fitness score but with the lowest position proximity to the best leader. Besides using the global best solution to

guide the overall swarm (as discussed in Section 3.1), the crossover operators are used to generate a set of hybrid elite signals, in order to overcome premature stagnation. These offspring leaders are yielded using sine, cosine and tanh oriented crossover probabilities, and the aforementioned two swarm leaders serve as the parent chromosomes. These offspring solutions are able to exploit the distinctive search regions between the two best leaders in different directions and scales, therefore avoiding stagnation.

3.2.1 The first proposed crossover strategy

Equation (9) defines the first hybrid optimal leader generation operation. It includes two strategies, i.e. adaptive and random crossover actions. Specifically, the adaptive crossover operation is applied to the randomly selected sub-dimensions of the generated offspring. For the remaining sub-dimensions of the offspring, the random crossover operation is adopted.

$$L_1 = \begin{cases} \cos(1.5 \times t/iters) \times s_{sec} + \sin(1.5 \times t/iters) \times \sin(1.5 \times t/iters) \times \tanh(1.5 \times t/iters) \times g_{best}, if\ d \in [1, rand_{dims}] \\ \cos(1.5 \times rand) \times s_{sec} + \sin(1.5 \times rand) \times \sin(1.5 \times rand) \times \tanh(1.5 \times rand) \times g_{best}, \quad otherwise \end{cases} \tag{9}$$

where g_{best} and s_{sec} represent the parent chromosomes of the global best solution and the remote second leader, respectively. L_1 denotes the yielded hybrid leader 1, while $rand$ indicates a randomly generated value between 0 and 1. In addition, both parameters t and $iters$ specify the current and the maximum numbers of iterations, respectively. As shown in Equation (9), the crossover probabilities are implemented using sine, tanh, and cosine formulae. Figure 3 depicts the search trajectories produced by $cos(m)$ and $sin(m) \times sin(m) \times tanh(m)$, respectively, where $m \in [0, 1.5]$ is used to represent $1.5 \times t/iters$ and $1.5 \times rand$ in Equation (9). As shown in Figure 3, they indicate the increasing and decreasing mild shifts between the global best and the second remote leaders for generating the leader offspring.

Referring to Equation (9), the first part indicates the adaptive method. This operation assigns adaptive crossover factors, i.e. the decreasing and increasing weights, to both parent chromosomes in randomly selected sub-dimensions over iterations. As such, the procedure starts with a greater focus on the second leader before switching the attention to the global best solution. The gradients of the search transitions between the two swarm leaders are illustrated by two adaptive courses shown in Figure 3. Instead of purely using the adaptive weightings for offspring generation, the second crossover operation defined in the second part of Equation (9) inserts randomness and assigns randomly generated values from both curves in Figure 3 for offspring generation. This proposed leader generation mechanism is conducted for each individual in each iteration, in order to produce diverse bespoke optimal signals. The yielded offspring leader is subsequently used to lead the search process, as defined in Equation (10).

$$x_i^{t+1} = x_i^t + h_i^t \times (L_1 - x_i^t) \tag{10}$$

where L_1 is the hybrid offspring leader yielded by Equation (9), and h_i^t denotes the search coefficient based on Equations (2)-(8). The search action in Equation (10) is able to explore the regions between the two remote best leaders in an adaptive and dynamic way by following a geometric surface to overcome local optima traps.

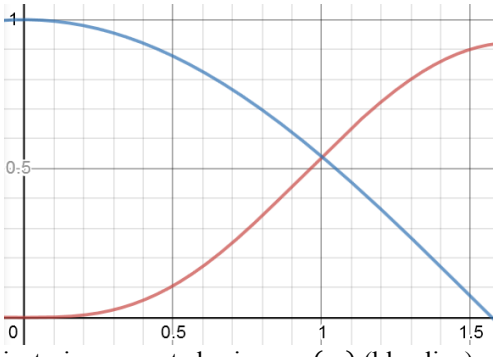

Figure 3 The adaptive search trajectories generated using $\cos(m)$ (blue line) and $\sin(m) \times \sin(m) \times \tanh(m)$ (red line) respectively, where $m \in [0, 1.5]$

3.2.2 The second proposed crossover strategy

To further diversify the search process, another crossover strategy for hybrid leader generation is formulated, as defined in Equation (11).

$$L_2 = \begin{cases} \cos(1.5 \times t/iters) \times \cos(1.5 \times t/iters) \times s_{sec} + \sin(1.5 \times t/iters) \times \sin(1.5 \times t/iters) \times g_{best}, if\ d \in [1, rand_{dims}] \\ \cos(1.5 \times rand) \times \cos(1.5 \times rand) \times s_{sec} + \sin(1.5 \times rand) \times \sin(1.5 \times rand) \times g_{best}, \quad otherwise \end{cases} \tag{11}$$

where L_2 indicates the hybrid offspring leader 2. As indicated in Equation (11), the crossover factors are implemented using adaptive and random sine and cosine formulae. Figure 4 illustrates the trajectories produced by $cos(m) \times cos(m)$ and $sin(m) \times sin(m)$, respectively, where $m \in [0, 1.5]$ is used to represent $1.5 \times t/iters$ and $1.5 \times rand$ in Equation (11). Again two search mechanisms, i.e. the adaptive and random operations, are employed in Equation (11) for generating the leader offspring by exploiting the search region between the global best solution and the remote second leader.

Figure 4 The adaptive search trajectories generated using $\cos(m) \times \cos(m)$ (blue line) and $\sin(m) \times \sin(m)$ (red line) respectively, where $m \in [0, 1.5]$

As indicated in Figure 4, the yielded curves are comparatively steeper and show faster transitions between the two best leaders, in comparison with those produced using Equation (9) and illustrated in Figure 3. Therefore, the offspring elite signal generation is dominated by the second swarm leader at the initial stage, while the impact of the global best solution is increased more radically over the iterations. Moreover, both the adaptive and random crossover operations defined in Equation (11) are taken into account for offspring production. The resulting indicators extend the search territory by exploring the distinctive search regions between the two remote leaders, as compared with those exploited in Equation (9).

The offspring solution is subsequently used to lead the search procedure defined in Equation (12). A distinctive super-ellipse search coefficient, h_i^t, is also extracted from the parametric super-ellipse surface defined in Equations (2)-(8) for each individual in the swarm. Based on Equation (12), each individual is equipped with different degrees of momentum and energy to explore the bespoke promising regions, in order to overcome stagnation.

$$x_i^{t+1} = x_i^t + h_i^t \times (L_2 - x_i^t) \tag{12}$$

3.2.3 The third proposed crossover strategy

Another offspring leader generation mechanism is also defined in Equation (13). This new crossover operation is implemented using another set of sine and cosine formulae. Figure 5 signifies the trajectories produced by $cos(m) \times cos(1.5 \times m)$ and $sin(m) \times cos(0.1 \times m)$ respectively, where $m \in [0, 1.5]$ is used to represent $1.5 \times t/iters$ and $1.5 \times rand$ in Equation (13). Again, the crossover operations follow the adaptive scheme in randomly selected sub-dimensions with the rest of the elements manufactured using the random mechanism. As indicated in Figure 5, the adaptive trajectories illustrate a comparatively early crossing with the sharpest diminishing and increasing momentum in comparison with those defined in Equations (9) and (11). This leads to more drastic switching of the search emphasis between the two swarm leaders. In other words, the effect of the second leader is decreasing sharply whereas the impact of the global best solution is increasing drastically during the search process. This phenomenon enables the newly generated leader to explore wider search regions in comparison with those based on Equations (9) and (11). The generated offspring leader is used to lead the search process, as defined in Equation (14), where the search coefficient h_i^t also follows a parametric surface.

$$L_3 = \begin{cases} \cos(1.5 \times t/iters) \times \cos(1.5 \times 1.5 \times t/iters) \times s_{sec} + \sin(1.5 \times t/iters) \times \cos(0.1 \times 1.5 \times t/iters) \times g_{best}, if\ d \in [1, rand_{dims}] \\ \cos(1.5 \times rand) \times \cos(1.5 \times 1.5 \times rand) \times s_{sec} + \sin(1.5 \times rand) \times \cos(0.1 \times 1.5 \times rand) \times g_{best}, \quad otherwise \end{cases} \tag{13}$$

$$x_i^{t+1} = x_i^t + h_i^t \times (L_3 - x_i^t) \tag{14}$$

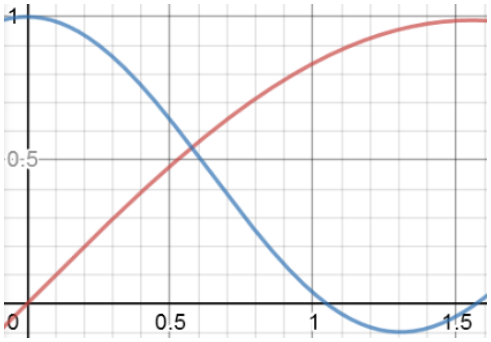

Figure 5 The adaptive search trajectories generated using $\cos(m) \times \cos(1.5 \times m)$ (blue line) and $\sin(m) \times \cos(0.1 \times m)$ (red line) respectively, where $m \in [0, 1.5]$

Besides the search operation led by each offspring solution, the three hybrid offspring leaders produced thus far are averaged to indicate the promising elite region in the search space, which is used to guide the search process as defined in Equation (15).

$$x_i^{t+1} = x_i^t + h_i^t \times (mean(L_1{+}L_2{+}L_3) - x_i^t) \tag{15}$$

where $mean(L_1{+}L_2{+}L_3)$ denotes the mean position of the three generated offspring leaders. Each individual in the swarm moves towards this elite indicator by following a super-ellipse parametric hyper-plane surface.

The proposed leader generation mechanisms produce a variety of distinctive optimal signals and enable the search process to effectively balance between diversification and intensification, in an attempt to overcome stagnation. Diversified geometric super-ellipse coefficients are assigned to each individual, in order to equip it with different accelerated search steps to move towards the elite signals. These proposed search mechanisms account for the significant capabilities of the proposed SI algorithm in overcoming local optima traps and achieving global optimality. It is used to optimize the hyper-parameters of deep BLSTM networks for human action recognition. We introduce the proposed ensemble CNN-BLSTM networks with optimal hyper-parameter identification in detail in the next section.

4. THE PROPOSED ENSEMBLE MODEL OF DEEP NETWORKS FOR HUMAN ACTION RECOGNITION

We employ a hybrid CNN-BLSTM model, i.e., a CNN encoder and a BLSTM decoder, for video action classification. In the literature, hybrid CNN-LSTM networks have been used for time series analysis, computer vision, and language processing tasks in recent years, including energy consumption prediction [45], visual question answering [46], and image description generation [47, 48]. As such, this hybrid architecture is adopted in this research. Instead of using LSTM as the decoder, we employ the BLSTM network because of its superior capabilities in video interpretation. In comparison with the existing video action recognition methods which employ spatial and temporal cues separately, the proposed hybrid model embeds CNN and BLSTM networks in series to capture temporal dynamics between the deep spatial features. Moreover, many existing studies rely heavily on computationally expensive pre-processing steps, such as motion information extraction using optical flow, and show significant restrictions to tackle real-time video classification tasks. In contrast, the proposed CNN-BLSTM model does not depend on optical flow extraction, and is capable of performing video action classification in real time.

Specifically, a pre-trained CNN model extracts deep spatial features from the video frames, while a BLSTM network analyses the temporal sequential cues from the extracted features of the video frames. The proposed SI algorithm is subsequently employed to optimize the learning rate, dropout rate, and the number of hidden neurons of the BLSTM network. Diverse optimized BLSTM models with a variety of distinctive learning configurations are yielded. An ensemble model is constructed by combining three optimized CNN-BLSTM networks for human action classification.

We first employ the ImageNet pre-trained GoogLeNet to extract the spatial feature vectors from video frames, owing to its efficient feature description capabilities. Before conducting feature transformation, several pre-

processing steps are performed, which include centre-cropping where we crop the longest edges of a video and re-size it to the input size (i.e. 224×224) of GoogLeNet. We also extract 30 frames per second from each video clip for deep feature extraction, in order to better capture the detailed and granular variations between frames for sequential analysis. The 'activations' function associated with the last pooling layer of the GoogLeNet is used to extract the feature vectors from the sequences of the video frames. The obtained sequences of feature vectors are then used as inputs to the BLSTM network for human action classification.

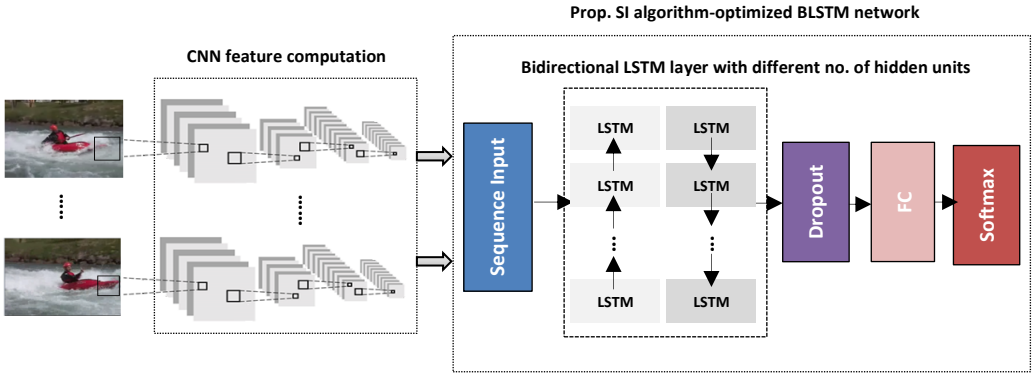

Figure 6 The proposed CNN-BLSTM architecture for each base model

In comparison with the unidirectional LSTM network, the BLSTM decoder is able to employ both forward and backward dependencies in deep frame features for classification purposes. Specifically, it embeds two LSTM layers of opposite directions to learn the temporal sequential dynamics from the deep spatial features. The information from the backward and forward passes is used simultaneously to inform the output layer. Based on the existing and empirical studies, it shows superior performances over those of unidirectional LSTM networks in time series analysis [31, 49]. The employed CNN-BLSTM architecture in this research for each base model is illustrated in Figure 6.

As shown in Figure 6, we design the BLSTM model with the following topology, i.e. a sequence input layer, a bidirectional LSTM layer, a dropout layer, a fully connected layer, a softmax layer, and a prediction layer. Particularly, the BLSTM layer is used to capture the bidirectional temporal dependencies from the sequences of extracted deep feature vectors. As we know, the network configurations, such as the number of hidden neurons of the BLSTM layer and the dropout rate of the dropout layer, as well as the training settings such as the learning rate, play important roles in determining the network capabilities in sequential information extraction. As such, an automated hyper-parameter selection process is developed using the proposed SI algorithm. The detailed search ranges of these hyper-parameters are provided in Table 1.

The number of hidden units of the BLSTM layer indicates the amount of dynamics and dependencies stored between the time steps. If the number of hidden neurons is too small, the model may not be able to capture the complex dynamic dependencies between the deep frame features effectively. If the number of hidden neurons is too large, the model is more likely to suffer from overfitting by capturing noisy, excessive, and redundant information. On the other hand, the dropout layer randomly ascribes zero to the input elements with a given dropout mask. Specifically, the dropout factor acts as a control threshold that determines the likelihood of any input elements to be dropped out. Such a dropout operation is able to change the network topology, and it plays an important role in preventing the issue of overfitting. A reasonable dropout rate leads to an optimal trade-off between network diversity and convergence, while a smaller setting is likely to result in restricted model diversity. Other network hyper-parameters, such as the learning rate, also have a significant impact on the learning behaviours. A larger learning rate is more likely to result in suboptimal solutions, while a smaller setting may lead to a less efficient learner with a slow convergence rate. In order to obtain an effective network and learning configuration, we utilize the proposed SI algorithm to identify the optimal settings of the hyper-parameters of BLSTM.

Table 1 Optimized hyper-parameters

Optimized factors	Search ranges
Learning rate	[5e-5, 5e-4]
Dropout rate	[0.1, 0.7]
The number of hidden units of the BLSTM layer	[400, 1800]

The following parameter settings are used for optimal hyper-parameter identification, i.e. dimension=3 and the maximum number of function evaluations=population (15) × a maximum number of iterations (10). For each data set, we employ the officially published training and test splits for model evaluation. The published training set has also been further divided into training and validation sets for hyper-parameter selection.

The proposed SI model initializes a swarm where each element of an individual is used to represent each of the optimized hyper-parameters. The proposed search operations are used to guide each individual in the search process pertaining to optimal parameter settings. For the fitness evaluation, the hyper-parameters recommended by each individual are used to establish the BLSTM network and its associated learning configuration. The BLSTM network is then trained and evaluated using the newly split training and validation sets for fitness evaluation. The resulting deep network with the most optimal hyper-parameter settings is indicated by the global best solution. It is further fine-tuned using the combined training and validation sets with larger numbers of training epochs, before it is evaluated with the unseen test set.

Moreover, video sequences that are comparatively longer than the typical cases can embed a large amount of padding in the training process. This can result in a negative impact on the classification performance. As such, temporal augmentation techniques such as random cropping or subsampling of the consecutive frames are employed to tackle the long video sequences. In addition, we employ the Adam optimizer for training the BLSTM network. The L2Regularization term is used in the loss function to prevent overfitting. This is achieved through assigning additional penalties on the layer parameters during optimization. We employ a mini-batch size of 16 and shuffle the video sequences every epoch in the training process. The input video frames are re-sized to 224×224, in order to match the input size of the encoder (GoogLeNet).

The hyper-parameter optimization process is conducted for a series of 10 runs to produce 10 optimized BLSTM networks, each with a distinctive network and learning configuration. These optimized networks are re-trained using three larger numbers of epochs (e.g. 15, 20, and 30) to yield 30 distinctive BLSTM networks using the combined training and validation sets. Next, a number of ensemble classification models are constructed. Each contains a set of three CNN-BLSTM networks with different optimized settings as the base classifiers. In other words, three optimized CNN-BLSTM networks are included in one ensemble model where Figure 6 illustrates the structure of each optimized base network. These base networks possess different layer structures and learning configurations, in order to enhance the ensemble model diversity. Finally, the average of the prediction probabilities is used to produce the final prediction result based on the outputs of the base networks. A comprehensive evaluation of the proposed ensemble model is provided in the next section.

5. EVALUATION

We evaluate the efficiency of the proposed ensemble model of evolving CNN-BLSTM networks using three human action data sets, i.e. KTH, UCF50 and UCF101. KTH [50] is a constrained data set, which has a total of 2,391 video sequences from 25 subjects. The video sequences are categorized into six actions, i.e. walking, jogging, running, boxing, hand-waving, and hand-clapping. We utilize the official training, validation, and test splits [50, 51] from two different sets of 8 subjects and the remaining 9 subjects, respectively, in model evaluation. The data set has a spatial resolution of 160×120. Unlike the KTH data set, both UCF50 [52] and UCF101 [53] are obtained from YouTube. They are realistic human action clips with a frame resolution of 320×240. Specifically, a total of 6,618 video samples from 50 classes are included in UCF50, while UCF101 contains 13,320 videos from 101 classes. For both UCF50 and UCF101, the video clips of each action class are divided into 25 groups, with 4 to 7 clips in each group. The clips in one group have similar characteristics such as the same actors or scenes. Following other related studies, we employ the leave-one-group-out cross-validation method for UCF50, and the first official training and test split for UCF101, in model evaluation. Example video frames from the KTH, UCF50 and UCF101 data sets are illustrated in Figure 7.

We employ the original PSO algorithm [54] and 9 modified PSO models for performance comparison, i.e. Genetic PSO (GPSO) [55], Autonomous Group PSO (AGPSO) [56], a Bare-bones PSO variant (BBPSOV) [3], PSO with genetic and random mutation operators (GMPSO) [57], Dynamic Neighbourhood Learning PSO (DNLPSO) [58], COSPSO [11], a modified PSO (MPSO) with time-varying acceleration coefficients and an adaptive inertia weight factor [25], Enhanced Leader PSO (ELPSO) [59], and a PSO variant (PSOVA) with elliptical coefficients [23]. As illustrated in Table 2, the parameter settings of these baseline methods are based on the original studies, while the experimental settings of our model are obtained via trial-and-error. An ensemble model of CNN-BLSTM networks with the default hyper-parameter settings is also employed for performance comparison, where learning rate=1e-4, dropout rate=0.1, and the number of hidden units=400. As mentioned earlier, the ensemble networks with the default and optimal settings identified by each search method re-size the input frames to 224×224, in order to match the input size of the encoder network.

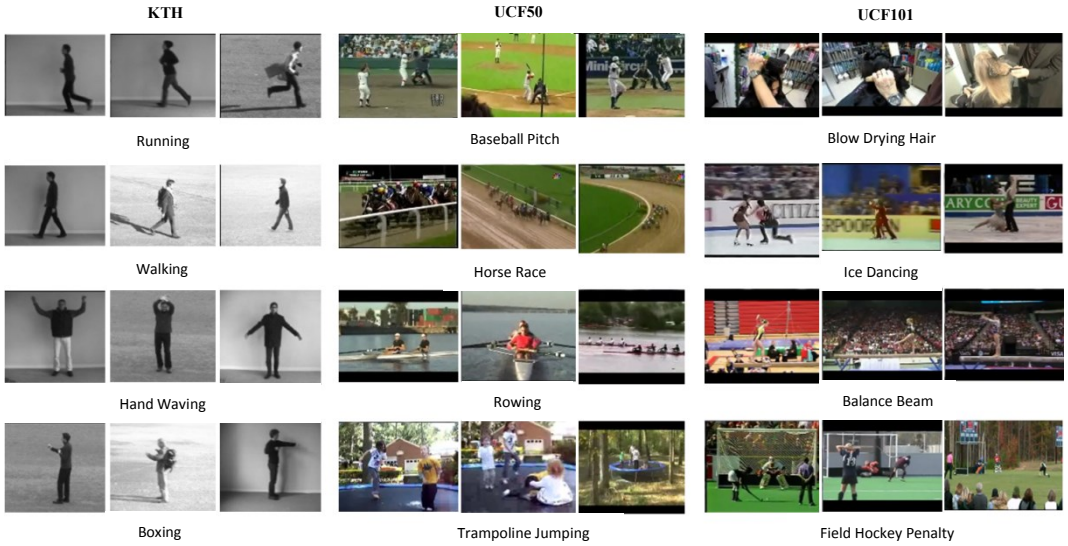

Figure 7 Example video frames from the KTH, UCF50 and UCF101 data sets

Table 2 Parameter settings of the proposed and the baseline methods

Method	Parameter setting
PSO [54]	maximum velocity=0.6, inertia weight=0.5, acceleration coefficients c_1= c_2= 1.5
GPSO [55]	maximum velocity=0.6, inertia weight=0.9, acceleration parameters c_1 = 2.6, c_2 = 1.5
AGPSO [56]	maximum velocity=0.6, adaptive decreasing c_1 and increasing c_2 over generations, inertia weight=0.9−(0.9−0.4)×(t)/(MaxGeneration), where t and MaxGeneration represent the current and maximum iteration numbers, respectively.
BBPSOV [3]	attraction and evading coefficients generated using the logistic map
MPSO [25]	time-varying acceleration coefficients and an adaptive inertia weight factor
COSPSO [11]	cosine annealing oriented search parameters
ELPSO [59]	c_1=c_2=2, standard deviation of Gaussian mutation=1, scale parameter of Cauchy mutation=2, scale factor of DE-based mutation=1.2, inertia weight=0.9−(0.9−0.4)×(t−1)/(MaxGeneration−1), where t and MaxGeneration represent the current and maximum iteration numbers, respectively.
GMPSO [57]	maximum velocity=0.6, inertia weight=0.5, acceleration constants c_1 = c_2 = 1.5, standard deviation of Gaussian distribution=1, scaling factor of Cauchy distribution=2, crossover probability=0.6, mutation probability=0.05
DNLPSO [58]	c_1=c_2=1.49445, refreshing gap=3, regrouping period=5, inertia weight=0.9−(0.9−0.4)×(t−1)/(MaxGeneration−1), where t and MaxGeneration represent the current and maximum iteration numbers, respectively.
PSOVA [23]	maximum velocity=0.6, inertia weight=0.6, with elliptical search parameters
Prop. model	search coefficients generated by a 3D super-ellipse parametric hyper-plane surface, crossover factors for the two remote leaders produced by nonlinear sine, cosine and tanh functions

5.1 Evaluation Using the KTH Data Set

We first evaluate the proposed ensemble model of CNN-BLSTM networks using the KTH data set. All the 2,391 video sequences are employed for evaluation. The proposed SI algorithm is used to identify the optimal settings of the learning and dropout rates as well as the number of hidden neurons of the BLSTM network. The official training-test split of the data set [50, 51] is adopted in this research, where the training, validation, and test sets comprise video sequences from two different sets of 8 subjects and the remaining 9 subjects, respectively. We employ the training and validation sets in the learning stage for hyper-parameter search, where the classification accuracy of the validation set is used as the fitness score.

The following experimental settings are employed, i.e., maximum number of function evaluations=population (15) × maximum number of iterations (10). A training epoch of 10 is employed during fitness evaluation. The optimized network with the most optimal settings is selected based on the global best solution. A series of 10 runs is performed for each search method in order to devise a number of optimized CNN-BLSTM networks with distinctive optimal hyper-parameters and the BLSTM layer structures.

We then re-train each devised base network using the combined training and validation sets in three larger numbers of training epochs, i.e. 15, 20 and 30. A total of 30 base CNN-BLSTM networks with distinctive hyper-

parameters are thus obtained. We aggregate a set of three base networks into an ensemble model based on the average of prediction probabilities. The mean classification performance of the yielded 10 ensemble models is provided in Table 3 for performance comparison.

Table 3 The mean accuracy rates of the KTH data set over 10 runs

Models	Methodologies	Accuracy rates	Rank sum
Prop. SI Algorithm-based Ensemble	Prop. SI algorithm + ensemble CNN-BLSTM model	0.963	n/a
PSO-based Ensemble	PSO + ensemble CNN-BLSTM model	0.9413	+
GPSO-based Ensemble	GPSO + ensemble CNN-BLSTM model	0.9452	+
BBPSOV-based Ensemble	BBPSOV + ensemble CNN-BLSTM model	0.9508	+
AGPSO-based Ensemble	AGPSO + ensemble CNN-BLSTM model	0.9341	+
MPSO-based Ensemble	MPSO + ensemble CNN-BLSTM model	0.9236	+
COSPSO-based Ensemble	COSPSO + ensemble CNN-BLSTM model	0.9389	+
GMPSO-based Ensemble	GMPSO + ensemble CNN-BLSTM model	0.9523	+
ELPSO-based Ensemble	ELPSO + ensemble CNN-BLSTM model	0.9259	+
DNLPSO-based Ensemble	DNLPSO + ensemble CNN-BLSTM model	0.9398	+
PSOVA-based Ensemble	PSOVA + ensemble CNN-BLSTM model	0.9483	+
Ensemble Model with Default Settings	Ensemble CNN-BLSTM model with default parameter settings	0.9218	+

As illustrated in Table 3, the proposed ensemble model of CNN-BLSTM networks yields the best mean accuracy rate of 96.3%. This is followed by the ensemble deep networks devised by the GMPSO, BBPSOV and PSOVA models with the mean accuracy rates of 95.23%, 95.08% and 94.83%, respectively. Since the optimized base CNN-BLSTM learners yielded by each search method are equipped with distinctive learning and network settings, their resulting ensemble models demonstrate significant diversity in comparison with the one with default settings.

The Wilcoxon rank sum test is conducted to ascertain the statistical difference in performance of the proposed SI algorithm over the baseline methods in hyper-parameter fine-tuning. The symbols, i.e. '+', '=' and '-', indicate whether our ensemble model is statistically better, the same, or worse than those yielded by the baseline methods. As indicated in Table 3, our devised ensemble model shows statistically better performance over those of all the baseline methods with all the *p*-values lower than 0.05, as indicated by the '+' symbol.

Table 4 The mean optimal hyper-parameters and accuracy rates over 10 runs for the KTH data set

		Accuracy	Learning rate	Dropout rate	Hidden units
Prop. SI	mean	0.963	0.00018	0.2815	864
MPSO	mean	0.9236	0.0001	0.1310	632
ELPSO	mean	0.9259	0.0003	0.2979	779
PSO	mean	0.9413	0.00014	0.3005	817
GPSO	mean	0.9452	0.00022	0.4373	945
COSPSO	mean	0.9389	0.00013	0.3879	1039
GMPSO	mean	0.9523	0.00027	0.3227	1075
PSOVA	mean	0.9483	0.0001	0.2242	1113.7
BBPSOV	mean	0.9508	0.00028	0.3827	1114
DNLPSO	mean	0.9398	0.00015	0.2483	1236
AGPSO	mean	0.9341	0.0001	0.2805	1302

Table 4 shows the mean results of the identified hyper-parameters over a set of 10 runs. In comparison with the settings identified by all the baseline methods, the proposed SI algorithm identifies a moderate mean learning rate, a moderate mean dropout rate, and a moderate mean number of hidden neurons. The empirical results indicate that such moderate model settings offer the required flexibilities in capturing the long-term temporal dependencies between deep frame features for human action classification. The deep networks constructed by the BBPSOV and GMPSO models yield competitive performances, which illustrate larger numbers of hidden neurons in combination with larger learning and dropout rates to balance between convergence and diversity. DNLPSO and AGPSO recruit larger numbers of hidden neurons and smaller learning and dropout rates, which are highly susceptible to overfitting. MPSO and ELPSO devise the BLSTM layers with the smallest numbers of hidden neurons, leading to restricted capabilities in capturing the temporal sequential dependencies and compromising the performance.

The confusion matrix of the proposed ensemble model of CNN-BLSTM networks is provided in Table 5. As shown in Table 5, the action categories of boxing, hand-clapping and walking are recognised with perfect accuracy (100%), while the remaining ones, i.e., hand-waving, jogging, and running, are identified with good

accuracy rates of more than 90%. Owing to the similarity in actions, such as jogging and running, both categories have been misclassified as each other in some cases. The clips of hand-waving have also been misclassified as the instances of hand-clapping occasionally. Overall, the devised ensemble model shows great superiority in identifying different action categories in the KTH data set.

In addition, Table 6 shows a comparison with the related studies. Since the existing studies have used different training and test sets, as well as different input sizes, Table 6 serves as an approximate performance comparison between the proposed model and the baseline methods. As mentioned earlier, our model re-sizes the video frames to 224×224 in order to match the input size of the encoder network, while other existing studies employ a variety of input sizes to balance between performance and computational cost. Among the baseline methods, Jaouedi et al. [61] and Zhang et al. [66] achieved the best performances. Specifically, Jaouedi et al. [61] used the same input size (224×224) as in this research with three methods, i.e. (1) Gaussian mixture model (GMM)+Kalman filter (KF)+k-Nearest Neighbours (KNN), (2) Gated Recurrent Neural Network (GRNN), and (3) GMM+KF+GRNN, for action classification. Moreover, their best results were obtained using the GMM+KF+GRNN method. GMM and KF were first used for background subtraction and bounding box extraction, respectively, and a simple GRNN model with one hidden Gated Recurrent Unit (GRU) layer was used for feature extraction and classification. Since GRU is a simplified LSTM unit, it has comparatively limited learning capabilities in capturing complex temporal dynamics from the frame features. In addition, their ROI extraction procedures based on GMM and KF are prone to noise, leading to limited capabilities in tackling regional proposal generation in realistic videos with complex scenes (e.g. as those in UCF101). Their GMM+KF+GRNN model achieved an accuracy rate of 96.3%, while the other two methods, i.e. GMM+KF+KNN and GRNN, produced accuracy rates of 71.1% and 86%, respectively. Instead of using the official training-test split of the KTH data set [50, 51], their work employed an aspect ratio of 75-25 for forming the training and test sets. In comparison with our model, a comparatively larger training set and a smaller test set have been used for evaluation. Furthermore, our model outperforms another top performer, i.e. Zhang et al. [66], where global silhouette and local optical flow have been used for action classification with a comparatively larger input size, i.e. 514×670. In short, our ensemble model of evolving CNN-BLSTM networks depicts competitive results, and it is among the top performers for human action classification with the KTH data set.

Table 5 The confusion matrix of the proposed ensemble network for the KTH data set

	boxing	**handclap.**	**handwav.**	**jogging**	**Running**	**walking**
boxing	1	0	0	0	0	0
handclap.	0	1	0	0	0	0
handwav.	0	0.0556	0.9444	0	0	0
jogging	0		0	0.9167	0.0833	0
running	0	0	0	0.0833	0.9167	0
walking	0	0	0	0	0	1

Table 6 Performance comparison for the KTH data set

Studies	**Methodology**	**Input size**	**Accuracy rates**
Babu et al. [60]	Meta-cognitive radial basis function network	160×120	0.9044
Jaouedi et al. [61]	GMM+KF+KNN (75-25 for training and test split)	224×224	0.711
Jaouedi et al. [61]	GRNN (75-25 for training and test split)	224×224	0.86
Jaouedi et al. [61]	GMM+KF+GRNN (75-25 for training and test split)	224×224	0.963
Fu et al. [62]	Sparse coding-based space-time video representation	-	0.9433
Liu et al. [51]	Hierarchical clustering multi-task learning	-	0.943
Latah [63]	3D CNN+SVM (70-30 for training and test split)	80×60	0.9034
Zhang et al. [64]	3D CNN	-	0.9167
Rodriguez et al. [65]	Maximum a posteriori (MAP) adapted GMM+simplex-Hidden Markov Model (SHMM)	160×120	0.942
Zhang et al. [66]	Global silhouette+local optical flow	514×670	0.95
Yousefi and Loo [38]	A dual fast and slow feature interaction	200×142	0.9007
Naidoo et al. [67]	Spatial-temporal analysis and bag of visual words (80-20 for training and test split)	-	0.82
Dasari and Chen [68]	MPEG compact descriptors for visual search feature trajectories with Fisher Vector (FV) encoding	-	0.875
Tong et al. [69]	Deep nonnegative matrix factorization	-	0.9396
Najar et al. [70]	Unsupervised learning of finite full covariance multivariate generalized GMMs	-	0.9197
Leyva et al. [71]	Spatio-temporal binary feature descriptor	-	0.9305
This research	**Prop. SI algorithm-optimized evolving ensemble deep networks**	224×224	**0.963**

5.2 Evaluation Using the UCF50 Data Set

In this evaluation, all the 6,618 video clips in the UCF50 data set [52] are used in the experiments. In order to compare with related studies, we employ the leave-one-group-out cross-validation method reported in [52] for evaluation. As an example, at the training stage, we employ the video clips from randomly selected 24 out of 25 groups from each action class for optimal hyper-parameter selection. The remaining group from each action class is used for evaluating the resulting model.

We further divide the learning samples from 24 groups using an aspect ratio of 80-20 into the training and validation sets, respectively. The accuracy rate of the validation set is used as the fitness score. The deep CNN-BLSTM network with the most optimal configuration is selected based on the global best solution. We combine the training and validation sets into a larger set, i.e. the video clips from 24 groups of each action class, to re-train the identified network. Three optimized base networks are aggregated based on the averages of prediction probabilities. The yielded ensemble model is assessed with the remaining unseen group for performance comparison. We subsequently select another set of 24 groups for training with the remaining group for evaluation. As such, the mean results from both experimental studies are used for performance comparison.

The experimental settings of the optimal network hyper-parameter selection are as follows, i.e. maximum number of function evaluations=population (15) × maximum number of iterations (10). For each search method, a series of 10 runs is performed for each pair of the training and test settings using the above process, in order to devise a number of optimized CNN-BLSTM networks with distinctive optimal hyper-parameters and the BLSTM layer structures. To balance between performance and computational efficiency, we employ a comparatively smaller learning epoch=3 at the training stage, in comparison with the training epoch of 10 for the KTH data set. Each optimized base model is subsequently trained using the original training set (i.e. the clips of 24 groups from each class) with three larger numbers of training epochs, i.e. 25, 30 and 50.

We aggregate three CNN-BLSTM networks into an ensemble model based on the average of prediction probabilities to yield the final output. The mean classification performance of various ensemble models with respect to both test groups with a series of 10 runs for each group is provided in Table 7.

Table 7 The mean accuracy rates for the UCF50 data set

Models	Methodologies	Accuracy rates	Rank sum
Prop. SI Algorithm-based Ensemble	Prop. SI algorithm + ensemble CNN-BLSTM model	0.9222	n/a
PSO-based Ensemble	PSO + ensemble CNN-BLSTM model	0.8856	+
GPSO-based Ensemble	GPSO + ensemble CNN-BLSTM model	0.8872	+
BBPSOV-based Ensemble	BBPSOV + ensemble CNN-BLSTM model	0.9004	+
AGPSO-based Ensemble	AGPSO + ensemble CNN-BLSTM model	0.8720	+
MPSO-based Ensemble	MPSO + ensemble CNN-BLSTM model	0.8665	+
COSPSO-based Ensemble	COSPSO + ensemble CNN-BLSTM model	0.8911	+
GMPSO-based Ensemble	GMPSO + ensemble CNN-BLSTM model	0.8856	+
ELPSO-based Ensemble	ELPSO + ensemble CNN-BLSTM model	0.8642	+
DNLPSO-based Ensemble	DNLPSO + ensemble CNN-BLSTM model	0.8732	+
PSOVA-based Ensemble	PSOVA + ensemble CNN-BLSTM model	0.8833	+
Ensemble Model with Default Settings	Ensemble CNN-BLSTM model with default parameter settings	0.8599	+

As illustrated in Table 7, the ensemble model of evolving deep networks constructed by the proposed SI algorithm achieves the best mean accuracy rate of 92.22%. As compared with other baseline search methods, the proposed SI algorithm employs diverse accelerated search strategies led by multiple leaders to overcome stagnation. Our optimized base networks with distinctive model and learning settings indicate great superiority over those yielded by other search methods. The ensemble model generated by BBPSOV shows a competitive performance of 90.04%, owing to the adoption of attractiveness and evading actions with chaotic search coefficients in its search strategies. The COSPSO, GPSO, GMPSO, and PSO devised ensemble networks also illustrate good performance with accuracy rates of 89.11%, 88.72%, 88.56% and 88.56%, respectively. In addition, the ensemble model yielded by each search method possesses significant diversity and illustrates better performance than that of the ensemble model with default settings.

We conduct the Wilcoxon rank sum test to indicate the statistical significance of the proposed SI algorithm against other baseline methods. As indicated in Table 7, our devised ensemble model shows statistically better results over those of the baseline methods with all the *p*-values lower than 0.05, as indicated by the '+' symbol.

Table 8 The mean optimal hyper-parameters and accuracy rates over 20 runs for the UCF50 data set

		Accuracy	Learning rate	Dropout rate	Hidden units
Prop. SI	mean	0.9222	0.00028	0.3452	839.3
MPSO	mean	0.8665	0.0001	0.1053	649.8
GPSO	mean	0.8872	0.0001	0.3382	699.5
BBPSOV	mean	0.9004	0.0002	0.4644	754
COSPSO	mean	0.8911	0.00032	0.5134	932
PSO	mean	0.8856	0.00018	0.4305	942.2
GMPSO	mean	0.8856	0.0001	0.2356	948.3
PSOVA	mean	0.8833	0.00023	0.3195	979.1
DNLPSO	mean	0.8732	0.00026	0.4154	1024
AGPSO	mean	0.872	0.0001	0.3979	1110.6
ELPSO	mean	0.8642	0.00023	0.3418	1130.5

Table 8 shows the mean hyper-parameters identified over 20 runs of each search method for both test groups (with 10 runs in each test group). Owing to the employment of larger training and validation sets in comparison with those of KTH, we can observe different search behaviours with respect to each search method. The proposed SI algorithm and the BBPSOV model identify moderate mean numbers of hidden neurons, moderate or larger mean learning rates and dropout rates in comparison with those from other search methods. Such network settings illustrate competitive capabilities in capturing the bidirectional long-term dependencies of deep features between the time steps. The identified moderate or large mean dropout rates of both models lead to a reasonable trade-off between network diversity and convergence speed. The MPSO and GPSO constructed models have the smallest hidden network configurations, which show limitations in capturing the delicate temporal dependencies, therefore their suboptimal performance. On the contrary, DNLPSO, AGPSO and ELPSO identify the largest numbers of hidden neurons, which are susceptible to preserving noisy, excessive and redundant motion details, therefore their inferior classification results.

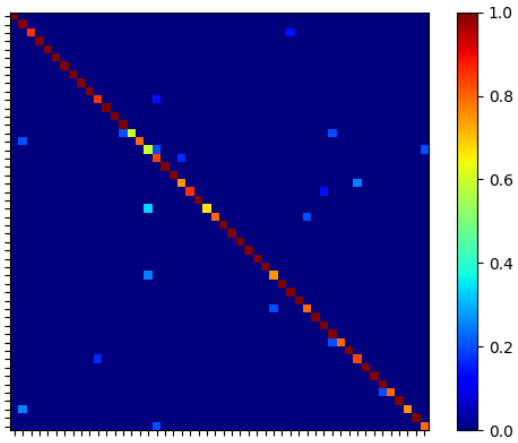

Figure 8 The confusion matrix of the proposed ensemble model for UCF50

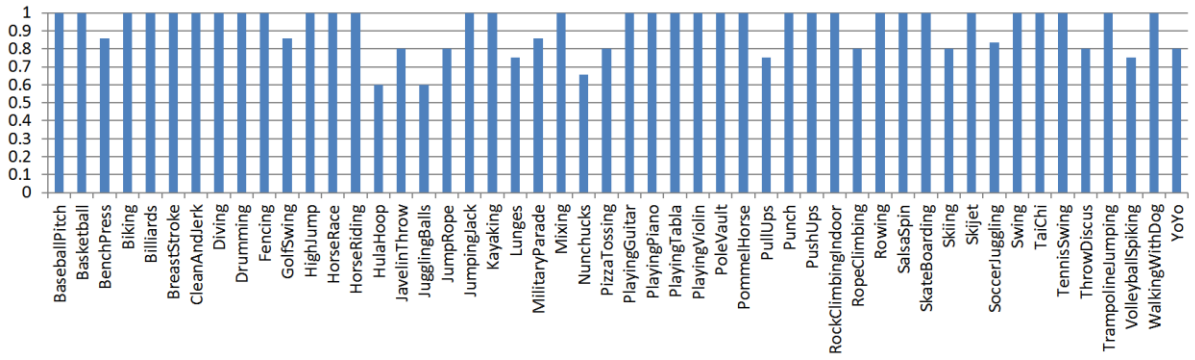

Figure 9 The mean accuracy rate for each class in UCF50

The confusion matrix of our devised ensemble model for UCF50 is illustrated in Figure 8, while the detailed mean accuracy rates are illustrated in Figure 9. As indicated in Figure 9, the devised ensemble network achieves good accuracy rates for the UCF50 data set. Specifically, it achieves over 80% accuracy rates for 44 action categories, and a total of 33 classes have an accuracy rate of 100%. There are only three classes, i.e. JugglingBalls, HulaHoop and Nunchucks with accuracy rates lower than 70%. As an example, as indicated in Figure 10, owing to a high degree of similarity in the motions performed as well as the shared spatial background details, the JugglingBalls actions are misidentified as YoYo. In addition, the HulaHoop videos are misclassified as SkateBoarding while the Nunchucks actions are misrecognised as JugglingBalls occasionally. There are also other classes that show a high degree of inter-class similarity, such as RopeClimbing and PullUps, where the former is misidentified as the latter in some cases, and vice versa. Overall, our devised deep ensemble network produces superior performance for the classification of most action categories in UCF50. Owing to the identification of sufficient number of hidden neurons, our ensemble model captures the temporal relationships effectively from the spatial features, leading to enhanced discriminative capabilities for most action categories.

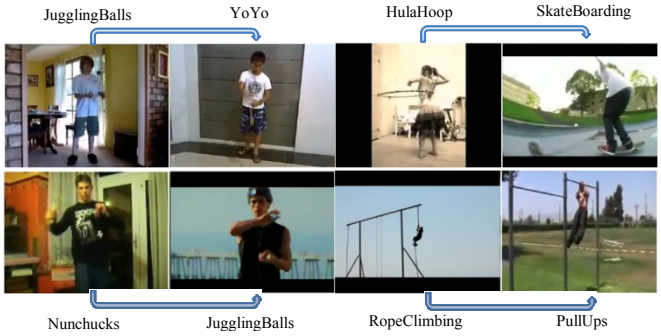

Figure 10 Examples of misclassification cases for UCF50

Table 9 illustrates a comparison with related studies for UCF50. Since different training and test sample sizes, input frame resolutions, as well as data augmentation techniques, were adopted in related methods, Table 9 serves as a rough comparison. Moreover, many existing studies employed the original video frame size of 320×240 (e.g. Kantorov and Laptev [73] and Uijlings et al. [76]) or a modified frame size of 256×256 (e.g. Liu et al. [77]). Comparatively, our proposed model uses a comparatively smaller frame resolution of 224×224. As indicated in Table 9, our ensemble networks devised by the proposed SI algorithm depict a competitive performance in comparison with those of the existing state-of-the-art studies, providing an alternative solution for solving human action classification problems.

Table 9 Performance comparison for the UCF50 data set

Studies	Methodology	Input size	Accuracy rates
Reddy and Shah [52]	Moving and stationary pixels+motion features	-	0.769
Wang et al. [72]	Dense trajectories and motion boundary descriptors	-	0.856
Dasari and Chen [68]	MPEG compact descriptors for visual search feature trajectories with FV encoding	-	0.548
Kantorov and Laptev [73]	MPEG flow+FV	320×240	0.822
Narayan and Ramakrishnan [74]	Object trajectories+causality descriptors	-	0.894
Luo et al. [75]	Combined linear dynamical systems and cuboids based distances with class dependent weights	-	0.8474
Uijlings et al. [76]	HOG+histograms of optical flow (HOF)+motion boundary histograms (MBH)	320×240	0.818
Liu et al. [77]	Convolutional neural random fields	256×256	0.865
Wang et al. [78]	HOF+MBH+FV+automatic human detection	-	0.917
Wilson and Mohan [79]	Sparsity-based coherent and noncoherent dictionaries+ action bank features	-	0.663
Zhang et al. [80]	Dynamical information + background information	-	0.8276
Zhang et al. [81]	A maximum margin part (MMP)-based method+a recursive part elimination	-	0.924
Leyva et al. [71]	Spatio-temporal binary feature descriptor (STB)+FV	-	0.8305
This research	**Prop. SI algorithm-optimized evolving ensemble deep networks**	224×224	**0.9222**

5.3 Evaluation Using the UCF101 Data Set

The UCF101 data set [53] is employed for further evaluation of the proposed model. We utilize all the 101 classes with a total of 13,320 videos for human action classification. The first official training-test split of the

UCF101 data set is adopted in our experiment, where the video clips of the first 7 groups from each class are used for evaluation while those in the remaining 18 groups from each class are used for training. An aspect ratio of 80-20 is employed to further divide the training set for learning and validation of hyper-parameters. The training experimental set-up of UCF50 is adopted in this experiment, i.e. maximum number of function evaluations=population (15) × maximum number of iterations (10), with maximum learning epoch=3. For each search method, a series of 10 runs is conducted to produce the base CNN-BLSTM models with distinctive networks and learning hyper-parameters. Each optimized base model is subsequently trained using the original training set (i.e. the video clips of 18 groups from each class) with larger numbers (i.e. 25, 30, and 50) of training epochs. Then, an ensemble model incorporating three base optimized CNN-BLSTM networks is constructed where the average of prediction probabilities is used to produce the final output. Table 10 presents the mean classification results of the ensemble models with respect to each search method.

Table 10 The mean accuracy rates for the UCF101 data set over 10 runs

Models	Methodologies	Accuracy rates	Rank sum
Prop. SI Algorithm-based Ensemble	Prop. SI algorithm + ensemble CNN-BLSTM model	0.8478	n/a
PSO-based Ensemble	PSO + ensemble CNN-BLSTM model	0.8239	+
GPSO-based Ensemble	GPSO + ensemble CNN-BLSTM model	0.8364	+
BBPSOV-based Ensemble	BBPSOV + ensemble CNN-BLSTM model	0.8399	+
AGPSO-based Ensemble	AGPSO + ensemble CNN-BLSTM model	0.8353	+
MPSO-based Ensemble	MPSO + ensemble CNN-BLSTM model	0.8319	+
COSPSO-based Ensemble	COSPSO + ensemble CNN-BLSTM model	0.8389	+
GMPSO-based Ensemble	GMPSO + ensemble CNN-BLSTM model	0.8242	+
ELPSO-based Ensemble	ELPSO + ensemble CNN-BLSTM model	0.8299	+
DNLPSO-based Ensemble	DNLPSO + ensemble CNN-BLSTM model	0.8308	+
PSOVA-based Ensemble	PSOVA + ensemble CNN-BLSTM model	0.8369	+
Ensemble Model with Default Settings	Ensemble CNN-BLSTM model with default parameter settings	0.8131	+

As illustrated in Table 10, the developed ensemble model achieves the best mean accuracy rate of 84.78% for the classification of 101 classes in UCF101. The proposed SI algorithm employs distinctive promising global indicators with dynamic accelerated search steps that follow a 3D geometric surface to overcome stagnation and achieve global optimality. The empirical results indicate the efficiency and robustness of the identified hyper-parameters of our optimized base networks in comparison with those from other search methods. The devised ensemble models of BBPSOV, COSPSO, and PSOVA also show reasonable performance with accuracy rates of 83.99%, 83.89% and 83.69%, respectively. In addition, the ensemble model constructed by each search method indicates significant diversity to overcome biases and variance of the base networks, whereas the ensemble model with default settings depicts constrained model flexibility owing to the adoption of base learners with similar settings. The superiority of our optimized ensemble model is also evidenced by the statistical test. As shown in Table 10, our result distributions are significantly different from those of the baseline methods with all the *p*-values lower than 0.05, as indicated by the '+' symbol.

Table 11 The mean optimal hyper-parameters and accuracy rates over 10 runs for the UCF101 data set

		Accuracy	Learning rate	Dropout rate	Hidden units
Prop. SI	mean	0.8478	0.0001	0.2372	802
MPSO	mean	0.8319	0.0001	0.1634	575
GPSO	mean	0.8364	0.00023	0.2711	610.33
BBPSOV	mean	0.8399	0.00023	0.2431	642
COSPSO	mean	0.8389	0.00025	0.2954	830
PSOVA	mean	0.8369	0.0002	0.3893	862
AGPSO	mean	0.8353	0.00017	0.4761	939.33
ELPSO	mean	0.8299	0.00017	0.1894	960
PSO	mean	0.8239	0.0002	0.2355	972.67
DNLPSO	mean	0.8308	0.0005	0.3949	1204
GMPSO	mean	0.8242	0.0001	0.1058	1551

The mean hyper-parameters identified over a series of 10 runs are provided in Table 11. The devised deep networks by the proposed model possess comparatively smaller learning rates with medium numbers of hidden neurons and dropout rates. Such network topologies show superior capabilities in capturing sufficient discriminative temporal patterns of the deep features without suffering from a constrained or excessive storage of dependencies between time steps. In contrast, the DNLPSO and GMPSO models select comparatively larger numbers of hidden neurons, while MPSO and GPSO yield smaller numbers of hidden neurons in the BLSTM layers. The empirical results indicate that the networks identified by DNLPSO and GMPSO are more likely to

suffer from overfitting by capturing excessive information between the video sequences. On the other hand, the networks yielded by MPSO and GPSO are constrained owing to the limited learning capabilities.

We further conduct a theoretical comparison between the proposed SI algorithm and the baseline methods. In GPSO [55], the PSO algorithm is integrated with a crossover operator to further enhance the swarm diversity. In AGPSO [56] and COSPSO [11], adaptive linear and nonlinear search coefficients are incorporated into the PSO algorithm to accelerate convergence, while both time-varying acceleration coefficients and the inertia weight factor are embedded in MPSO [25]. DNLPSO [58] employs the neighbouring historical best experiences for position updating. ELPSO [59] uses a series of distribution probabilities and mutation mechanisms to further enhance the global best solution, while GMPSO [57] enhances the swarm leader by adopting both genetic operators and random walk strategies.

The above PSO variants and the original PSO algorithm conduct a monotonous search operation guided by single leader for position updating. Therefore, they are more likely to be trapped in local optima. On the other hand, besides using the logistic map as the search parameters, BBPSOV [3] employs a mean vector of the local and global promising solutions, as well as a mean position of the local and global worst particles, to guide the attractiveness and evading actions, respectively. In particular, it assigns fixed and equal weightings to the local and global best solutions throughout the search iterations pertaining to the mean leader vector generation. Therefore, the model is subject to a less efficient trade-off between exploitation and exploration during the search course. A similar case is also applied to PSOVA [23]. In PSOVA, besides using the swarm leader and a randomly selected neighbouring elite solution for position updating, it employs a mean vector of all the neighbouring fitter solutions to guide the search process. Fixed and equal weights are assigned to the neighbouring promising solutions throughout the search course pertaining to the mean leader vector generation, therefore a less efficient trade-off between intensification and diversification.

In comparison with these baseline methods, the proposed SI algorithm employs a versatile search process led by multiple distinctive promising indicators to overcome stagnation. Instead of assigning fixed weights to the parent chromosomes for hybrid leader generation as in BBPSOV and PSOVA, the proposed SI algorithm employs dynamic weight factors, i.e. adaptive and random weights based on sine, cosine and tanh formulae, to yield elite offspring signals, resulting in a balance between intensification and diversification. Instead of using adaptive and chaotic search parameters, e.g. as in AGPSO and BBPSOV, respectively, the proposed SI algorithm equips each individual with distinctive super-ellipse geometric search coefficients extracted from a 3D hyper-plane surface, in order to exploit the optimal regions with different forces and scales in accelerating convergence. In this regard, when the search process guided by the global best solution stagnates, the operations led by each of the yielded elite signals and the mean of these promising solutions are able to drive the search out of local optima. Therefore, the proposed model outperforms other baseline methods in achieving global optimality. The empirical results indicate that competitive network configurations are formulated, which illustrate statistically better performance than those from the baseline methods for human action classification.

Table 12 The mean computational costs (in seconds) of one fitness evaluation for the proposed method and 10 baseline methods

	Prop. SI	MPSO	GPSO	BBPSOV	COSPSO	GMPSO	PSOVA	DNLPSO	AGPSO	ELPSO	PSO
KTH	124.7138	88.5894	144.0221	214.1905	165.5687	170.0996	223.1243	240.5063	282.2271	114.1644	119.2525
UCF50	351.3564	292.687	270.052	299.1994	430.4293	468.6627	487.1719	516.2275	593.7988	598.3891	461.7416
UCF101	484.9916	365.8315	399.376	427.9021	558.7587	1461.791	605.5556	920.4534	652.8815	697.4448	689.2222

We subsequently conduct computational efficiency comparison between the proposed model and 10 baseline methods for hyper-parameter search during the training stage. Since fitness evaluation is the most time-consuming component of each search method, we provide the mean computational cost of one fitness evaluation with respect to each method in Table 12. The experiments are conducted using one NVIDIA GTX 1080Ti consumer GPU. Specifically, in one fitness evaluation, the devised BLSTM network is trained and tested with the training and validation data sets for each problem. As indicated in Table 12, on average, the proposed model illustrates a comparatively lower mean computational cost for one fitness evaluation as compared with those from most of the baseline methods. The computational efficiency of our devised networks is attributed to the identification of moderate BLSTM layer settings, which result in moderate network complexity. In contrast, the deep networks devised by MPSO have the smallest numbers of hidden neurons. This results in comparatively lighter settings of the BLSTM layers, therefore lower the computational cost, but at the expense of a constrained storage of temporal dependencies. On the other hand, PSOVA, GMPSO, DNLPSO and AGPSO identify larger numbers of hidden neurons in the BLSTM networks, resulting in higher degrees of network complexity and higher mean computational costs. Overall, the proposed model achieves an efficient trade-off between

performance and computational cost. In addition, its computational costs are better than those from most of the baseline methods for optimal hyper-parameter search with all the test data sets.

Since we employ video frames as the inputs for human action classification, we select related studies that use video frames (i.e. RGB data only) as the inputs for performance comparison. Table 13 illustrates a comparison with several existing studies for the UCF101 data set. Since a variety of network architectures and different training and test strategies and data augmentation techniques have been reported, Table 13 serves as an approximate performance comparison. Many of the existing studies [28, 83, 86, 35, 87, 88, 36] employed the same input size (i.e. 224×224) as in our experiment, owing to the fact that it is widely used for many CNN encoders (e.g. VGG-16, VGG-19, GoogLeNet, ResNet-50 and ResNet-152). A slightly larger input size of 227×227 was adopted by Donahue et al. [85] and Ullah et al. [31], because of the requirement of their encoder networks (such as AlexNet). Other studies, e.g. Karpathy et al. [82] and Varol et al. [89], used input sizes of 170×170 and 112×112, respectively, to balance between performance and computational cost. Besides using 224×224 at the training stage, He et al. [87] employed multiple input scales at the test stage to further enhance performance.

As indicated in Table 13, with RGB images as inputs, CNN-based methods have been popularly used for human action classification, e.g., shallow [28] and deep [83] ConvNets, multiresolution CNNs [82], ResNet-50 [87], ResNet-152 [87], VGG-16 [88], and CNN with long-term temporal convolutions (LTC-CNN) [89]. On top of using CNNs, other studies added LSTM networks, which include a ConvLSTM network [84], long-term Recurrent Convolutional Network (LRCN) [85], soft attention LSTM [86], and VideoLSTM [35]. However, the networks in most of the above related studies using CNN-LSTM architectures employed a single unidirectional LSTM layer with a fixed number of hidden units, which could constrain the performance. Note that in Ng et al. [84], although the model employed a slightly smaller input size of 220×220, their encoder network was pre-trained using 1 million sport videos. Our CNN encoder and those of other studies were pre-trained using ImageNet only. Moreover, several CNN-LSTM networks were also proposed by Yang et al. [36], which embedded two or three LSTM layers as well as attention mechanisms. In their so-called attention-again model, the attention mechanisms were used multiple times in the LSTM layers, in order to achieve better performance. Ullah et al. [31] employed AlexNet as the encoder and the BLSTM network as the decoder for human action classification. Their study employed an aspect ratio of 60-20-20 for forming the training, validation, and test sets, instead of using the official training and test split, resulting in a comparatively smaller test size. Another limitation of such ad-hoc split is that it allocates videos with similar characteristics from the same group of each action into both training and test sets, leading to bias in performance.

Table 13 Performance comparison with related studies using RGB images only for the UCF101 data set

Studies	Methodologies	Input size	Accuracy rates
Karpathy et al. [82]	Slow fusion CNNs	170×170	0.654
Simonyan and Zisserman [28]	ConvNet (CNN-M-2048)	224×224	0.728
Simonyan and Zisserman [83]	A deep ConvNet	224×224	0.774
Ng et al. [84]	LSTM with 30 Frame Unroll	220×220	0.775
Donahue et al. [85]	LRCN networks	227×227	0.829
Sharma et al. [86]	Soft attention LSTM	224×224	0.77
Li et al. [35]	VideoLSTM	224×224	0.796
He et al. [87]	ResNet-50	224×224 (& other scales)	0.823
He et al. [87]	ResNet-152	224×224 (& other scales)	0.834
Feichtenhofer et al. [88]	VGG-16	224×224	0.826
Varol et al. [89]	CNN with long-term temporal convolutions (LTC-CNN)	112×112	0.824
Yang et al. [36]	ConvLSTM (three LSTM layers)	224×224	0.817
Yang et al. [36]	ConvLSTM (two LSTM layers)	224×224	0.824
Yang et al. [36]	ConvLSTM+attention	224×224	0.841
Yang et al. [36]	Attention-again model (GoogLeNet)	224×224	0.854
Ullah et al. [31]	AlexNet-BLSTM (using own split of 60-20-20 with a smaller test set)	227×227	0.9121
Leyva et al. [71]	STB+FV	-	0.716
This research	**Prop. SI algorithm-optimized evolving ensemble deep networks**	224×224	**0.8478**

Our ensemble networks compare favourably with most of the related studies using the RGB images as the inputs. They incorporate diverse base models with a variety of optimal settings to achieve enhanced performance. Firstly, our ensemble model of deep networks takes the advantage of the BLSTM layer to preserve and process information from both the past and the future for human action classification, in comparison with

unidirectional LSTM layers used in the existing studies [84, 85, 86, 35]. Comparing with manually-designed fixed model and training settings used in the existing studies, our ensemble model of networks exploits the interactive effects of the associated hyper-parameters and employs diverse optimal network topologies and learning settings for addressing the classification problem. This leads to an effective preservation of bidirectional discriminative sequential and motion dependencies for improving the performance, while overcoming the overfitting and underfitting issues. Indeed, as indicated in the empirical results, owing to the employment of optimized numbers of hidden units in BLSTM layers, our ensemble model discovers the temporal relationships effectively. Moreover, by combining base models with distinctive network and training settings, our ensemble learning mechanism is able to increase model diversity for boosting the classification performance.

Next, we conduct human action classification using both video frames and motion information as the inputs. Motivated by the two-stream model [28] and some related studies [84, 88, 89], where both RGB data and motion information (such as optical flow) are used as the inputs for classification, we combine our ensemble model with a motion CNN to further enhance performance in this additional experimental study. The optical flow images are used as the inputs of our motion CNN because they provide extra motion details of the human actions for classification.

Specifically, the TV-L1 optical flow images in [28] (derived using the method in [90]) are employed in this research. We fine-tune the ImageNet pre-trained ResNet101 model using the optical flow images with 30 epochs for human action classification. At the training stage, we randomly select 16 video clips in each mini-batch. A stack of optical flow images is further randomly chosen in each video clip. During the test stage, a set of 19 optical flow images is uniformly selected in each video clip. The majority voting strategy is used to yield the video-level prediction based on the prediction results of the 19 optical flow images. As illustrated in Table 14, the ResNet101 model obtains an accuracy rate of 76.55% for UCF101. The prediction results of our ensemble model and the ResNet101 network are then combined by taking the averages of prediction probabilities. Since the proposed ensemble model embeds spatial and temporal dynamics extracted from the video sequences, while the motion ResNet101 model is trained using new optical flow images, both networks illustrate significant complementary characteristics for performance enhancement. The integration of both models yields an accuracy rate of 92.09%. The corresponding confusion matrix is provided in Figure 11.

Table 14 The integration of the proposed ensemble CNN-BLSTM model and motion ResNet101

Model	Accuracy rate
The proposed SI-devised ensemble CNN-BLSTM network	0.8478
Motion ResNet101 (optical flow)	0.7655
Ensemble using the average of probabilities	0.9209

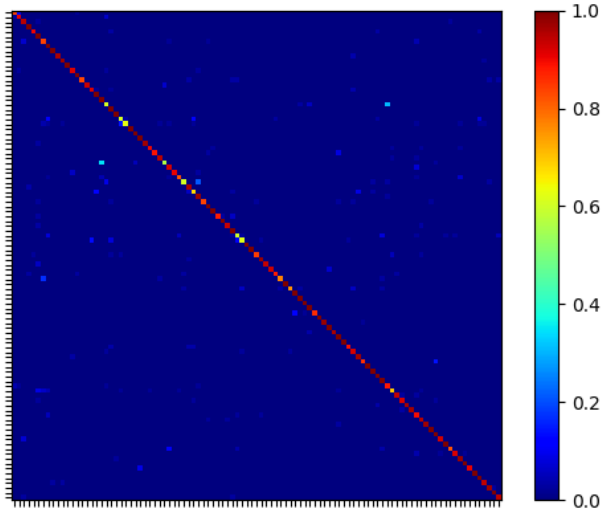

Figure 11 The confusion matrix of the proposed ensemble model integrated with motion ResNet101 for the UCF101 data set

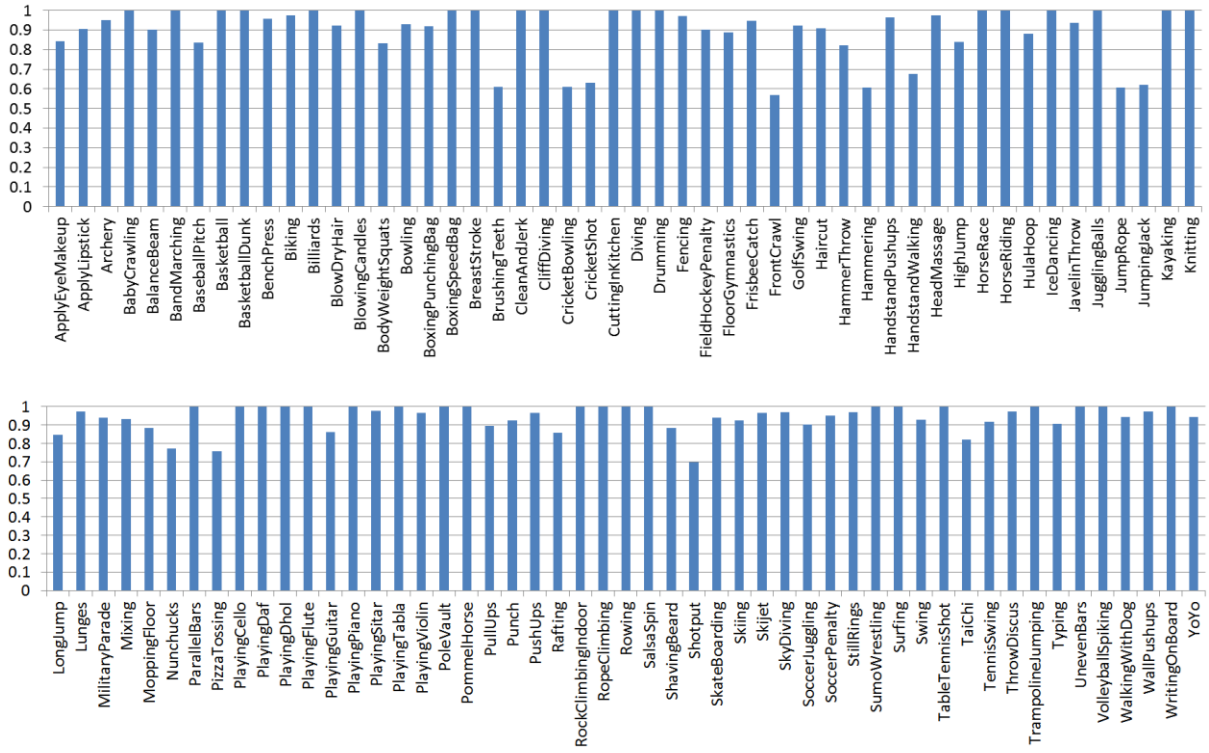

Figure 12 The mean classification accuracy rate for each class in UCF101

Figure 12 illustrates the detailed mean classification accuracy rates of UCF101 from our integrated network. The integrated model is able to achieve good classification accuracy for most of the action categories. Specifically, a total of 90 classes are recognised with over 80% accuracy rates, among which 39 classes illustrate a perfect accuracy rate of 100%. There is only one class, FrontCrawl, with an accuracy rate lower than 60%. As an example, FrontCrawl tends to be misclassified as BreastStroke. As indicated in Figure 13, both categories have extremely similar background details (e.g. swimming pools), which become the dominating spatial features and result in misclassification. Similarly, owing to the fine-grained actions and shared spatial background details, the JumpRope actions are misclassified as those of JumpingJack and vice versa, while BrushingTeeth actions are misclassified as those of ShavingBeard. HandstandWalking actions are also misidentified as those of HandstandPushups, occasionally. Overall, our integrated network achieves good classification results for most action categories in UCF101. Again, owing to the employment of optimized numbers of hidden units, it discovers the temporal relationships effectively. By incorporating with additional optical flow information, our model demonstrates enhanced discriminative capabilities for action classification.

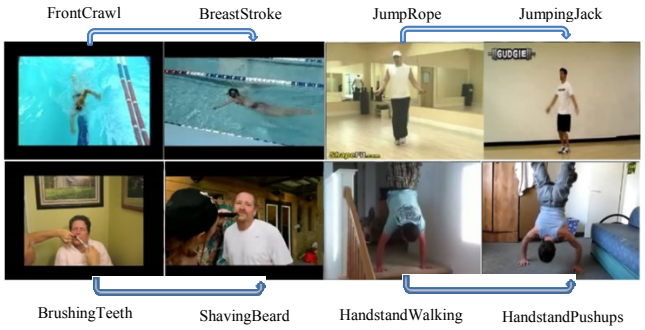

Figure 13 Examples of misclassification cases for UCF101

Table 15 shows the performance comparison with related studies using both RGB and motion inputs for the UCF101 data set. As shown in Table 15, the two-stream model which includes spatial and temporal CNNs was proposed by Simonyan and Zisserman [28], while various variants of the two-stream method were used in other related studies, e.g. different types of fusion of the spatial and temporal CNNs [88], the integration of the two-stream method and LSTM [84], the employment of motion-based attention based on optical flow and convolutional operations for spatial information storage [35], the implementation of long-term temporal convolutions in space-time CNNs [89], the adoption of motion vectors instead of optical flow as the inputs to the

temporal CNN [30, 91], and the proposal of a lightweight generator network to extract more refined Discriminative Motion Cue (DMC) representations [92]. Note that most of the existing studies [28, 88, 35, 30, 91, 92] employed the same input size, i.e. 224×224, as that of this research.

In comparison with these related studies, our integrated network achieves comparable performance. The devised ensemble model is equipped with base CNN-BLSTM networks with diverse optimal network configurations, and it delivers an impressive performance. The integration of the motion ResNet101 model and our ensemble model offers significant complementary capabilities. In short, the proposed ensemble model of evolving deep networks and its integration with a motion ResNet101 model depict superior performances, and serve as alternative solutions for human action classification.

Table 15 Performance comparison with related studies using both RGB and motion inputs for the UCF101 data set

Studies	Methodologies	Input Size	Accuracy rates
Simonyan and Zisserman [28]	Two-Stream CNNs (RGB+Flow)	224×224	0.88
Ng et al. [84]	Two-Stream+LSTM (RGB+Flow)	220×220	0.886
Feichtenhofer et al. [88]	Two-Stream (VGG-16) (RGB+Flow)	224×224	0.906
Li et al. [35]	VideoLSTM (RGB+Flow)	224×224	0.892
Varol et al. [89]	LTC-CNN (RGB+Flow)	112×112	0.917
Zhang et al. [30]	Enhanced motion vector CNN (EMV-CNN) (RGB+Motion Vector)	224×224	0.864
Zhang et al. [91]	Deeply transferred motion vector CNN (DTMV-CNN) (RGB+Motion Vector)	224×224	0.875
Shou et al. [92]	DMC-Net (ResNet-18) (Compressed Videos+Flow)	224×224	0.909
This research	**Prop. SI algorithm-optimized evolving ensemble deep networks + motion ReseNet 101**	224×224	**0.9209**

5.4 Evaluation Using Benchmark Functions

To further indicate the efficiency of the proposed SI algorithm for solving high-dimensional optimization problems, we employ a set of unimodal and multimodal benchmark functions for performance comparison. The employed test functions include Dixon-Price, Rotated Hyper-Ellipsoid, Rosenbrock, Sphere, Sum of Different Powers, Sum Squares, Ackley, Griewank, and Powell. The first six artificial landscapes are unimodal functions with single global optimum, while the remaining ones are multimodal functions with multiple local minima. The definitions of these benchmark functions are provided in Jordehi [59], Zhang et al. [93] and Pandit et al. [94].

Besides using the aforementioned baseline models, we use a set of additional search methods, i.e. the original FA and its variant models, for performance comparison, as these new methods illustrate significant performances for solving diverse unimodal and multimodal benchmark functions. Specifically, the new baseline methods include the original FA [95], FA with random attraction (RaFA) [96], a modified FA (MFA) [97], FA with neighbourhood attraction (NaFA) [98], and a repulsive FA (RFA) [94]. Table 16 shows the parameter settings of these additional baseline methods, which are extracted from their original studies.

Table 16 Parameter settings of the additional baseline methods

Method	Parameter setting
FA [95]	initial attractiveness=1.0, absorption coefficient=1.0, Levy's index=1.5, randomization parameter=0.5
NaFA [98]	absorption coefficient=1.0, Levy's index=1.5, and randomization parameter $\alpha(t+1) = \left(\frac{1}{9000}\right)^{\frac{1}{t}} \times \alpha(t)$ where t is the current iteration.
RaFA [96]	absorption coefficient=1.0, Levy's index=1.5, randomization parameter=0.5, using Cauchy distribution for global best enhancement.
MFA [97]	initial attractiveness=1.0, absorption coefficient=1.0, Levy's index=1.5, randomization parameter=0.5
RFA [94]	attractiveness=1.0, randomization parameter=0.25, absorption coefficient=1.0, repulsive force factor=0.6, repulsive immune threshold=0.1

We employ a dimension of 200 to assess the model efficiency in tackling these high-dimensional unimodal and multimodal benchmark problems. To ensure a fair comparison, a maximum number of function evaluations=4,000 is used as the termination criterion of each search method. This moderate experimental setting is able to better indicate the convergence rate of each search method. A series of 30 runs is performed, in order to eliminate any random factors. The mean, minimum, maximum, and standard deviation results over 30 runs are used as the main performance criteria for comparison. Table 17 depicts the detailed results, with the best ones highlighted in bold.

Based on Table 17, the proposed SI algorithm illustrates significant capabilities in tackling these high-dimensional optimization problems. It yields the best results for all the test functions, except for Dixon-Price, where RFA [94] has the best mean of global minimum. For unimodal functions, i.e. Rosenbrock, Sphere and Sum Squares, the proposed SI algorithms achieves the global optimality with the best mean fitness scores of '0', as compared with those of all the classical search methods and advanced PSO and FA variants. The statistical Wilcoxon rank sum test is conducted. As shown in Table 18, since nearly all the *p*-values are lower than 0.05, our results show statistically significant differences to those of the baseline methods in nearly all test cases. The only exception is Dixon-Price, where our model and RFA have similar performance distributions. Besides obtaining the best results in nearly all test cases, the proposed SI algorithm illustrates a significantly faster convergence rate than those of all 15 baseline methods.

The superior performance of the proposed SI algorithm is attributed to the injection of diverse position updating strategies guided by multiple elite signals with accelerated super-ellipse coefficients. These characteristics provide competitive advantages of the proposed SI algorithm over the monotonous operations of the baseline methods with fixed, random and adaptive search steps. In short, the devised mechanisms, which include adaptive crossover operators based on sine, cosine and tanh functions as well as 3D super formula oriented search coefficients, account for the significant superiority of the proposed SI algorithm over the baseline methods in solving diverse unimodal and multimodal benchmark functions. They are also useful for devising the ensemble model of evolving deep networks as shown in the experiments of human action classification.

Table 17 Evaluation results of various benchmark functions with dimension=200

		Prop. SI	PSOVA	BBPSOV	GPSO	AGPSO	DNLPSO	ELPSO	GMPSO	COSPSO	MPSO	RaFA
Ackley	mean	**8.88E-16**	1.69E+01	2.05E+01	1.98E+01	1.82E+01	1.65E+01	1.70E+01	5.04E+00	1.86E+01	1.82E+01	1.73E+01
	min	**8.88E-16**	1.63E+01	2.03E+01	1.95E+01	1.77E+01	1.42E+01	1.66E+01	3.64E+00	1.82E+01	1.75E+01	1.68E+01
	max	**8.88E-16**	1.75E+01	2.06E+01	2.01E+01	1.88E+01	1.89E+01	1.73E+01	7.08E+00	1.88E+01	1.89E+01	1.79E+01
	std	**0.00E+00**	2.98E-01	7.44E-02	1.52E-01	3.02E-01	1.27E+00	2.08E-01	7.02E-01	1.73E-01	2.83E-01	2.85E-01
Dixon	mean	**2.21E+00**	4.13E+05	1.29E+08	7.68E+07	7.01E+06	6.87E+06	1.94E+07	2.28E+05	2.66E+07	8.59E+07	1.68E+06
	min	**2.50E-01**	1.71E+05	1.12E+08	5.66E+07	3.29E+06	1.22E+06	1.87E+06	5.27E+04	1.77E+07	6.14E+07	7.20E+05
	max	**1.05E+01**	6.34E+05	1.43E+08	1.07E+08	1.16E+07	4.05E+07	2.42E+07	4.44E+05	3.32E+07	1.24E+08	2.52E+06
	std	**2.71E+00**	1.12E+05	7.35E+06	1.06E+07	1.94E+06	9.06E+06	4.04E+06	1.03E+05	3.69E+06	1.89E+07	3.97E+05
Griewank	mean	**1.96E-15**	8.86E-05	5.28E+03	3.61E+03	1.09E+03	9.47E+02	1.62E+03	2.39E+00	2.07E+03	3.34E+03	5.14E+02
	min	**0.00E+00**	5.98E-08	4.94E+03	3.18E+03	9.13E+02	4.71E+02	1.34E+03	1.65E+00	1.84E+03	2.60E+03	3.84E+02
	max	**5.88E-14**	4.64E-04	5.59E+03	4.05E+03	1.32E+03	1.58E+03	1.88E+03	3.83E+00	2.35E+03	4.20E+03	6.39E+02
	std	**1.07E-14**	1.25E-04	1.42E+02	2.36E+02	1.05E+02	3.15E+02	1.14E+02	5.25E-01	1.16E+02	3.41E+02	5.88E+01
Rothyp	mean	**2.07E-16**	9.05E+05	2.44E+07	1.66E+07	4.31E+06	3.41E+06	7.52E+06	2.65E+04	9.05E+06	1.55E+07	2.24E+06
	min	**0.00E+00**	5.53E+05	2.28E+07	1.33E+07	2.78E+06	1.73E+06	6.32E+06	1.35E+04	7.78E+06	1.21E+07	1.88E+06
	max	**6.20E-15**	1.27E+06	2.56E+07	1.89E+07	5.30E+06	5.60E+06	8.36E+06	5.71E+04	1.05E+07	1.89E+07	2.69E+06
	std	**1.13E-15**	1.70E+05	8.81E+05	1.15E+06	6.21E+05	9.93E+05	4.79E+05	1.04E+04	6.60E+05	1.74E+06	2.13E+05
Rosenb	mean	**0.00E+00**	3.01E+05	2.00E+07	1.20E+07	1.79E+06	8.23E+05	2.53E+06	2.14E+04	4.54E+06	4.87E+06	1.18E+06
	min	**0.00E+00**	1.52E+05	1.65E+07	8.30E+06	1.28E+06	1.60E+05	1.60E+06	1.02E+04	3.37E+06	3.65E+06	8.90E+05
	max	**0.00E+00**	4.62E+05	2.20E+07	1.56E+07	2.82E+06	2.86E+06	3.19E+06	5.47E+04	6.00E+06	6.09E+06	1.38E+06
	std	**0.00E+00**	6.97E+04	1.31E+06	1.67E+06	4.04E+05	7.02E+05	3.23E+05	1.07E+04	5.80E+05	6.28E+05	1.35E+05
Sphere	mean	**0.00E+00**	6.29E+01	1.55E+03	1.06E+03	3.16E+02	2.33E+02	4.72E+02	4.60E+01	6.10E+02	9.46E+02	1.44E+02
	min	**0.00E+00**	4.67E+01	1.39E+03	9.48E+02	2.39E+02	1.03E+02	4.00E+02	3.19E+01	5.38E+02	7.58E+02	1.07E+02
	max	**0.00E+00**	9.57E+01	1.68E+03	1.22E+03	3.74E+02	4.53E+02	5.24E+02	6.76E+01	6.78E+02	1.08E+03	1.74E+02
	std	**0.00E+00**	1.07E+01	6.31E+01	7.39E+01	3.91E+01	9.92E+01	3.02E+01	8.02E+00	3.58E+01	7.41E+01	1.80E+01
Sumpow	mean	**7.37E-25**	4.18E-05	2.18E+00	2.16E+00	6.48E-04	1.23E-03	3.22E-02	3.98E-06	1.66E-01	8.09E+00	8.83E-06
	min	**0.00E+00**	5.64E-07	1.45E+00	7.65E-01	4.68E-07	1.41E-08	8.47E-04	4.33E-12	5.10E-02	4.60E+00	9.98E-08
	max	**2.21E-23**	2.67E-04	3.00E+00	3.00E+00	8.22E-03	2.44E-02	9.72E-02	3.12E-05	3.87E-01	1.39E+01	4.21E-05
	std	**4.04E-24**	5.90E-05	4.08E-01	5.66E-01	1.59E-03	4.51E-03	2.53E-02	8.78E-06	8.66E-02	2.67E+00	1.09E-05
Sumsqu	mean	**0.00E+00**	5.77E+03	1.52E+05	1.02E+05	2.74E+04	2.10E+04	4.59E+04	4.07E+03	5.57E+04	9.58E+04	1.40E+04
	min	**0.00E+00**	3.79E+03	1.35E+05	9.07E+04	2.23E+04	9.23E+03	3.55E+04	2.66E+03	5.16E+04	7.50E+04	1.14E+04
	max	**0.00E+00**	7.69E+03	1.66E+05	1.20E+05	3.37E+04	3.90E+04	5.43E+04	6.56E+03	6.12E+04	1.22E+05	1.71E+04
	std	**0.00E+00**	1.07E+03	7.52E+03	6.85E+03	3.51E+03	8.05E+03	3.49E+03	8.40E+02	2.14E+03	9.94E+03	1.42E+03
Powell	mean	**8.55E-12**	1.76E+03	2.61E+05	1.47E+05	1.71E+04	1.19E+04	4.01E+04	1.50E+03	4.78E+04	1.46E+05	7.02E+03
	min	**0.00E+00**	6.53E+02	1.79E+05	1.07E+05	1.22E+04	2.91E+03	2.70E+04	6.76E+02	3.65E+04	7.57E+04	5.36E+03
	max	**2.57E-10**	2.56E+03	3.19E+05	2.30E+05	2.44E+04	2.53E+04	5.14E+04	2.68E+03	5.55E+04	2.38E+05	9.39E+03
	std	**4.68E-11**	4.58E+02	3.34E+04	2.74E+04	3.13E+03	6.12E+03	5.93E+03	5.42E+02	3.79E+03	3.92E+04	9.59E+02

		Prop. SI	MFA	NaFA	RFA	FA	PSO
Ackley	mean	**8.88E-16**	2.07E+01	1.52E+01	1.03E-02	1.31E+01	1.76E+01
	min	**8.88E-16**	2.07E+01	1.44E+01	9.33E-03	1.19E+01	1.69E+01
	max	**8.88E-16**	2.07E+01	1.63E+01	1.13E-02	1.46E+01	1.81E+01
	std	**0.00E+00**	1.08E-14	5.00E-01	6.02E-04	6.99E-01	3.29E-01
Dixon	mean	2.21E+00	1.16E+08	1.49E+06	**1.02E+00**	8.45E+05	1.93E+06
	min	**2.50E-01**	1.16E+08	5.86E+05	1.02E+00	4.44E+05	1.19E+06
	max	1.05E+01	1.16E+08	2.28E+06	**1.03E+00**	1.48E+06	3.29E+06
	std	2.71E+00	**1.36E-08**	3.82E+05	3.00E-03	2.56E+05	5.41E+05
Griewank	mean	**1.96E-15**	5.09E+03	4.94E+02	3.01E-03	2.86E+02	5.55E+02
	min	**0.00E+00**	5.09E+03	3.64E+02	2.33E-03	1.66E+02	4.25E+02
	max	**5.88E-14**	5.09E+03	6.30E+02	3.68E-03	3.79E+02	6.92E+02

	std	**1.07E-14**	2.36E-12	6.93E+01	3.17E-04	4.65E+01	7.68E+01
Rothyp	mean	**2.07E-16**	2.34E+07	2.05E+06	9.32E-01	1.25E+06	2.22E+06
	min	**0.00E+00**	2.34E+07	1.50E+06	7.67E-01	8.26E+05	1.51E+06
	max	**6.20E-15**	2.34E+07	2.79E+06	1.16E+00	1.79E+06	2.79E+06
	std	**1.13E-15**	1.20E-09	2.81E+05	1.21E-01	2.79E+05	3.39E+05
Rosenb	mean	**0.00E+00**	2.51E+07	5.66E+05	1.99E+02	2.43E+05	7.76E+05
	min	**0.00E+00**	2.51E+07	3.22E+05	1.99E+02	1.51E+05	4.22E+05
	max	**0.00E+00**	2.51E+07	1.19E+06	1.99E+02	3.91E+05	1.31E+06
	std	**0.00E+00**	1.28E-08	1.80E+05	5.11E-02	6.03E+04	2.08E+05
Sphere	mean	**0.00E+00**	1.48E+03	1.37E+02	6.26E-05	8.98E+01	1.64E+02
	min	**0.00E+00**	1.48E+03	9.65E+01	4.73E-05	5.92E+01	1.25E+02
	max	**0.00E+00**	1.48E+03	1.59E+02	7.91E-05	1.16E+02	2.50E+02
	std	**0.00E+00**	3.43E-13	1.53E+01	8.45E-06	1.66E+01	2.60E+01
Sumpow	mean	**7.37E-25**	1.02E+00	2.83E-06	8.38E-13	1.01E-05	1.08E-04
	min	**0.00E+00**	1.02E+00	3.09E-07	2.72E-14	7.80E-07	2.87E-06
	max	**2.21E-23**	1.02E+00	9.88E-06	4.23E-12	4.80E-05	5.14E-04
	std	**4.04E-24**	9.68E-16	2.30E-06	9.79E-13	1.18E-05	1.16E-04
Sumsqu	mean	**0.00E+00**	1.48E+05	1.21E+04	6.43E-03	8.03E+03	1.39E+04
	min	**0.00E+00**	1.48E+05	8.62E+03	5.36E-03	5.99E+03	1.07E+04
	max	**0.00E+00**	1.48E+05	1.63E+04	8.16E-03	1.00E+04	1.79E+04
	std	**0.00E+00**	5.82E-11	2.24E+03	6.89E-04	9.98E+02	1.56E+03
Powell	mean	**8.55E-12**	2.21E+05	5.44E+03	1.11E-03	3.66E+03	5.52E+03
	min	**0.00E+00**	2.21E+05	3.35E+03	8.25E-04	1.86E+03	2.94E+03
	max	**2.57E-10**	2.21E+05	7.36E+03	1.40E-03	5.20E+03	9.62E+03
	std	4.68E-11	**2.81E-11**	1.21E+03	1.44E-04	7.75E+02	1.61E+03

Table 18 The Wilcoxon rank sum test results of all benchmark functions with dimension=200

	PSOVA	**BBPSOV**	**GPSO**	**AGPSO**	**DNLPSO**	**ELPSO**	**GMPSO**	**COSPSO**	**MPSO**	**RaFA**	**MFA**	**NaFA**	**RFA**	**PSO**	**FA**
Ackley	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.69E-14	1.21E-12	1.21E-12	1.21E-12	1.21E-12
Dixon	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	7.69E-12	3.02E-11	1.86E-01	3.02E-11	3.02E-11
Griewank	1.72E-12	1.72E-12	1.72E-12	1.72E-12	1.72E-12	1.72E-12	1.72E-12	2.37E-12	2.37E-12	1.72E-12	6.84E-13	1.72E-12	1.72E-12	1.72E-12	1.72E-12
rothyp	1.72E-12	1.72E-12	1.72E-12	1.72E-12	1.72E-12	1.72E-12	1.72E-12	1.21E-12	1.21E-12	1.72E-12	9.45E-14	1.72E-12	1.72E-12	1.72E-12	1.72E-12
Rosenbrock	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	2.54E-13	1.21E-12	1.21E-12	1.21E-12	1.21E-12
Sphere	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	4.17E-13	1.21E-12	1.21E-12	1.21E-12	1.21E-12
sumpow	1.44E-11	1.44E-11	1.44E-11	1.44E-11	1.44E-11	1.44E-11	1.44E-11	4.11E-12	4.11E-12	1.44E-11	1.36E-11	1.44E-11	1.44E-11	1.44E-11	1.44E-11
sumsqu	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	8.70E-14	1.21E-12	1.21E-12	1.21E-12	1.21E-12
powell	5.22E-12	5.22E-12	5.22E-12	5.22E-12	5.22E-12	5.22E-12	5.22E-12	7.88E-12	7.88E-12	5.22E-12	3.71E-13	5.22E-12	5.22E-12	5.22E-12	5.22E-12

6. CONCLUSIONS

In this research, we have proposed an ensemble model of evolving CNN-BLSTM networks with optimal hyper-parameter identification using a new SI algorithm for human action classification based on video clips. The proposed SI algorithm incorporates diverse elite signals yielded by sine, cosine and tanh oriented crossover operators, as well as search coefficients following a parametric hyper-plane surface. It employs a versatile search process guided by multiple promising global signals to overcome the limitations of single-leader guided classical search methods. The proposed algorithm shows superior capabilities in identifying the optimal network settings for temporal sequential pattern extraction in BLSTM networks over a set of 10 classical and advanced search methods. Owing to the employment of diverse base CNN-BLSTM networks with distinctive optimized network settings, the resulting ensemble model illustrates significant diversity to further enhance performance. Evaluated using three publicly available human action data sets, the proposed ensemble model of evolving CNN-BLSTM networks produces statistically significant superiority over those with default and optimal settings identified by other classical and advanced search methods in human action recognition. The proposed SI algorithm also outperforms a number of PSO and FA variants with statistical significance in handling diverse high-dimensional unimodal and multimodal artificial landscapes. In short, the proposed search strategies of dynamic crossover operators and super-ellipse search coefficients account for the superior capabilities of the proposed SI algorithm in achieving global optimality with fast convergence rates for solving optimal hyper-parameter identification and numerical benchmark functions.

In future work, we will evaluate the proposed ensemble model in real-time human action recognition applications, e.g. for real-world surveillance and health monitoring. The proposed network architectures will also be evaluated using other image and video understanding tasks, e.g. visual question answering [46] and video summary [85]. Moreover, we aim to evaluate the proposed SI algorithm for hyper-parameter tuning and deep architecture generation with residual and dense connectivity in other computer vision tasks, such as large-scale object detection and classification [11, 12, 24] as well as image description generation [47, 48, 99].

CONFLICT OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] I. Jegham, A.B. Khalifa, I. Alouani and M.A. Mahjoub. (2020). Vision-based human action recognition: An overview and real world challenges. *Forensic Science International: Digital Investigation*, 32, p.200901.

[2] I. Rodríguez-Moreno, J.M. Martínez-Otzeta, B. Sierra, I. Rodriguez and E. Jauregi. (2019). Video activity recognition: State-of-the-art. *Sensors*, 19(14), p.3160.

[3] W. Srisukkham, L. Zhang, S.C. Neoh, S. Todryk and C.P. Lim. (2017). Intelligent Leukaemia Diagnosis with Bare-Bones PSO based Feature Optimization. *Applied Soft Computing*, 56 (2017) 405-419.

[4] S. Sundaramurthy and P. Jayavel. (2020). A hybrid Grey Wolf Optimization and Particle Swarm Optimization with C4. 5 approach for prediction of Rheumatoid Arthritis. *Applied Soft Computing*, p.106500.

[5] T.Y. Tan, L. Zhang, L., S.C. Neoh and C.P. Lim. (2018). Intelligent skin cancer detection using enhanced particle swarm optimization. *Knowledge-based systems*, 158, pp.118-135.

[6] K. Mistry, L. Zhang, S.C. Neoh, C.P. Lim and B. Fielding. (2017). A micro-GA Embedded PSO Feature Selection Approach to Intelligent Facial Emotion Recognition. *IEEE Transactions on Cybernetics*. PP (99) 1-14.

[7] C.J. Tan, S.C. Neoh, C.P. Lim, S. Hanoun, W.P. Wong, C.K. Loo, L. Zhang and S. Nahavandi. (2019). Application of an evolutionary algorithm-based ensemble model to job-shop scheduling. *Journal of Intelligent Manufacturing*, 30(2), pp.879-890.

[8] S.C. Neoh, W. Srisukkham, L. Zhang, S. Todryk, B. Greystoke, C.P. Lim, A. Hossain and N. Aslam. (2015). An Intelligent Decision Support System for Leukaemia Diagnosis using Microscopic Blood Images. *Scientific Reports*. 5 (14938) 1-14. Nature Publishing Group.

[9] H. Xie, L. Zhang, C.P. Lim, Y. Yu, C. Liu, H. Liu, and J. Walters. (2019). Improving K-means clustering with enhanced Firefly Algorithms. *Applied Soft Computing*, 84, p.105763.

[10] T.Y. Tan, L. Zhang and C.P. Lim. (2019). Intelligent skin cancer diagnosis using improved particle swarm optimization and deep learning models. *Applied Soft Computing*, p.105725.

[11] B. Fielding and L. Zhang. (2018). Evolving Image Classification Architectures with Enhanced Particle Swarm Optimisation. *IEEE Access*. Vol 6. 68560–68575.

[12] Y. Sun, B. Xue, M. Zhang and G.G. Yen. (2020). Evolving deep convolutional neural networks for image classification. *IEEE Transactions on Evolutionary Computation*, 24(2), pp.394-407.

[13] B. Shao, M. Li, Y. Zhao and G. Bian. (2019). Nickel Price Forecast Based on the LSTM Neural Network Optimized by the Improved PSO Algorithm. *Mathematical Problems in Engineering*, 2019.

[14] D. Chitradevi and S. Prabha. (2020). Analysis of brain sub regions using optimization techniques and deep learning method in Alzheimer disease. *Applied Soft Computing*, 86, p.105857.

[15] Y. Sun, H. Wang, B. Xue, Y. Jin, G.G. Yen and M. Zhang. (2020). Surrogate-assisted evolutionary deep learning using an end-to-end random forest-based performance predictor. *IEEE Transactions on Evolutionary Computation*, 24(2), pp.350-364.

[16] Y. Zhang and D. Xin. (2020). Dynamic Optimization Long Short-Term Memory Model Based on Data Preprocessing for Short-Term Traffic Flow Prediction. *IEEE Access*, 8, pp.91510-91520.

[17] C.W. Tsai, C.H. Hsia, S.J. Yang, S.J. Liu and Z.Y. Fang. (2020). Optimizing hyperparameters of deep learning in predicting bus passengers based on simulated annealing. *Applied Soft Computing*, 88, p.106068.

[18] B. Nakisa, M.N. Rastgoo, A. Rakotonirainy, F. Maire and V. Chandran. (2018). Long short term memory hyperparameter optimization for a neural network based emotion recognition framework. *IEEE Access*, 6, pp.49325-49338.

[19] J. Yan, X. Chen, Y. Yu and X. Zhang. (2019). Application of a Parallel Particle Swarm Optimization-Long Short Term Memory Model to Improve Water Quality Data. *Water*, 11(7), p.1317.

[20] L. Kang, R.S. Chen, W. Cao and Y.C. Chen. (2020). Non-inertial opposition-based particle swarm optimization and its theoretical analysis for deep learning applications. *Applied Soft Computing*, 88, p.106038.

[21] R. Santos, G. Borges, A. Santos, M. Silva, C. Sales and J.C. Costa. (2020). A rotationally invariant semi-autonomous particle swarm optimizer with directional diversity. *Swarm and Evolutionary Computation*, p.100700.

[22] Y. Chen, L. Li, H. Peng, J. Xiao and Q. Wu. (2018). Dynamic multi-swarm differential learning particle swarm optimizer. *Swarm and Evolutionary Computation*, 39, pp.209-221.

[23] L. Zhang and C.P. Lim. (2020). Intelligent optic disc segmentation using improved particle swarm optimization and evolving ensemble models. *Applied Soft Computing*, p.106328.

[24] Y. Sun, B. Xue, M. Zhang, G.G. Yen and J. Lv. (2020). Automatically Designing CNN Architectures Using the Genetic Algorithm for Image Classification. *IEEE Transactions on Cybernetics*.

[25] D.R. Nayak, R. Dash and B. Majhi. (2018). Discrete ripplet-II transform and modified PSO based improved evolutionary extreme learning machine for pathological brain detection. *Neurocomputing*. 282 (2018) 232–247.

[26] T.Y. Tan, L. Zhang, C.P. Lim, B. Fielding, Y. Yu and E. Anderson. (2019). Evolving Ensemble Models for Image Segmentation Using Enhanced Particle Swarm Optimization. *IEEE Access*. 7, 34004-34019.

[27] T.Y. Tan, L. Zhang and C.P. Lim. (2020). Adaptive melanoma diagnosis using evolving clustering, ensemble and deep neural networks. *Knowledge-Based Systems*, 187, p.104807.

[28] K. Simonyan and A. Zisserman. (2014). Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems* (pp. 568-576).

[29] Y. Wang, M. Long, J. Wang and P.S. Yu. (2017). Spatiotemporal pyramid network for video action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (pp. 1529-1538).

[30] B. Zhang, L. Wang, Z. Wang, Y. Qiao and H. Wang. (2018). Real-time action recognition with deeply transferred motion vector cnns. *IEEE Transactions on Image Processing*, 27(5), pp.2326-2339.

[31] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad and S.W. Baik. (2017). Action recognition in video sequences using deep bi-directional LSTM with CNN features. *IEEE Access*, 6, pp.1155-1166.

[32] B. Singh, T.K. Marks, M. Jones, O. Tuzel and M. Shao. (2016). A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1961-1970).

[33] C. Dai, X. Liu and J. Lai. (2020). Human action recognition using two-stream attention based LSTM networks. *Applied Soft Computing*, 86, p.105820.

[34] M.A. Khan, M. Sharif, T. Akram, M. Raza, T. Saba and A. Rehman. (2020). Hand-crafted and deep convolutional neural network features fusion and selection strategy: an application to intelligent human action recognition. *Applied Soft Computing,* 87, p.105986.

[35] Z. Li, K. Gavrilyuk, E. Gavves, M. Jain and C.G. Snoek. (2018). Videolstm convolves, attends and flows for action recognition. *Computer Vision and Image Understanding*, 166, pp.41-50.

[36] H. Yang, J. Zhang, S. Li, J. Lei and S. Chen. (2018). Attend it again: Recurrent attention convolutional neural network for action recognition. *Applied Sciences*, 8(3), p.383.

[37] M.V. da Silva and A.N. Marana. (2020). Human action recognition in videos based on spatiotemporal features and bag-of-poses. *Applied Soft Computing*, p.106513.

[38] B. Yousefi and C.K. Loo. (2018). A dual fast and slow feature interaction in biologically inspired visual recognition of human action. *Applied Soft Computing,* 62, pp.57-72.

[39] F. Yu, X. Wu, Y. Sun and L. Duan. (2018). Exploiting images for video recognition with hierarchical generative adversarial networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*.

[40] D. Tran, J. Ray, Z. Shou, S.F. Chang and M. Paluri. (2017). Convnet architecture search for spatiotemporal feature learning. *arXiv preprint arXiv:1708.05038*.

[41] C.Y. Wu, M. Zaheer, H. Hu, R. Manmatha, A.J. Smola and P. Krähenbühl. (2018). Compressed video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6026-6035).

[42] R. Singh, A. Sonawane and R. Srivastava. (2019). Recent evolution of modern datasets for human activity recognition: a deep survey. *Multimedia Systems*, pp.1-24.

[43] H.B. Zhang, Y.X. Zhang, B. Zhong, Q. Lei, L. Yang, J.X. Du and D.S. Chen. (2019). A comprehensive survey of vision-based human action recognition methods. *Sensors,* 19(5), p.1005.

[44] J. Gielis. (2003). A generic geometric transformation that unifies a wide range of natural and abstract shapes. *American Journal of Botany*, 90(3), pp.333-338.

[45] T.Y. Kim, and S.B. Cho. (2019). Predicting residential energy consumption using CNN-LSTM neural networks. *Energy*, 182, pp.72-81.

[46] M. Alizadeh and B. Di Eugenio. (2020). Augmenting Visual Question Answering with Semantic Frame Information in a Multitask Learning Approach. In *Proceedings of IEEE 14th International Conference on Semantic Computing* (ICSC) (pp. 37-44). IEEE.

[47] P. Kinghorn, L. Zhang and L. Shao. (2018). A region-based image caption generator with refined descriptions. *Neurocomputing*. 272 (2018) 416-424.

[48] P. Kinghorn, L. Zhang and L. Shao. (2019). A Hierarchical and Regional Deep Learning Architecture for Image Description Generation. *Pattern Recognition Letters*. 119 (2019) 77-85.

[49] V. Makarenkov, L. Rokach and B. Shapira. (2019). Choosing the right word: Using bidirectional LSTM tagger for writing support systems. *Engineering Applications of Artificial Intelligence*, 84, pp.1-10.

[50] C. Schuldt, I. Laptev and B. Caputo. Recognizing human actions: a local SVM approach. (2004). In *Proceedings of the 17th International Conference on Pattern Recognition* (ICPR 2004). 2004, 32–36.

[51] A.A. Liu, Y.T. Su, W.Z. Nie and M. Kankanhalli. (2017). Hierarchical clustering multi-task learning for joint human action grouping and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(1), pp.102-114.

[52] K. Reddy and M. Shah. (2012). Recognizing 50 Human Action Categories of Web Videos. *Machine Vision and Applications Journal*.

[53] K. Soomro, A.R. Zamir and M. Shah. (2012). UCF101: A dataset of 101 human actions classes from videos in the wild. In *CRCV-TR-12-01*.

[54] J. Kennedy and R. Eberhart. (1995). Particle Swarm Optimization. In *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, 1942–1948.

[55] Q. Chen, Y. Chen and W. Jiang. (2016). Genetic particle swarm optimization–based feature selection for very-high-resolution remotely sensed imagery object change detection. *Sensors*, 16 (8) 1204.

[56] S. Mirjalili, A. Lewis and A.S. Sadiq. (2014). Autonomous particles groups for particle swarm optimization. *Arabian Journal for Science and Engineering*. 39 (6) 4683–4697.

[57] Y. Zhang, L. Zhang, S.C. Neoh, K. Mistry and A. Hossain. (2015). Intelligent affect regression for bodily expressions using hybrid particle swarm optimization and adaptive ensembles. *Expert Systems with Applications*. 42 (22) 8678–8697.

[58] M. Nasir, S. Das, D. Maity, S. Sengupta, U. Halder and P.N. Suganthan. (2012). A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization. *Information Sciences*. 209 (2012) 16–36.

[59] A.R. Jordehi. (2015). Enhanced leader PSO (ELPSO): a new PSO variant for solving global optimisation problems. *Applied Soft Computing*. 26 (2015) 401–417.

[60] R.V. Babu, B. Rangarajan, S. Sundaram and M. Tom. (2015). Human action recognition inH.264/AVC compressed domain using meta-cognitive radial basis function network. *Applied Soft Computing*. 36 (2015) 218–227.

[61] N. Jaouedi, N. Boujnah and M.S. Bouhlel. (2020). A new hybrid deep learning model for human action recognition. *Journal of King Saud University-Computer and Information Sciences*, 32(4), pp.447-453.

[62] Y. Fu, T. Zhang and W. Wang. (2016). Sparse coding-based space-time video representation for action recognition. *Multimedia Tools and Applications.* 76(10):12645–12658. 10.1007/s11042-016-3630-9

[63] M. Latah. (2017). Human action recognition using support vector machines and 3D convolutional neural networks. *International Journal of Advances in Intelligent Informatics*, 3(1), pp.47-55.

[64] J. Zhang, L. Chen and J. Tian. (2017). 3D Convolutional Neural Network for Action Recognition. In *Proceedings of CCF Chinese Conference on Computer Vision*. 600–607.

[65] M. Rodriguez, C. Orrite, C. Medrano and D. Makris. (2017). One-shot learning of human activity with an MAP adapted GMM and simplex-HMM. *IEEE Transactions on Cybernetics*, 47(7), pp.1769-1780.

[66] N. Zhang, Z. Hu, S.H. Lee and E.J. Lee. (2017). Human action recognition based on global silhouette and local optical flow. In *Proceedings of the 2nd International Symposium on Mechanical Engineering and Material Science*.

[67] D. Naidoo, J. R. Tapamo and T. Walingo. (2018). Human Action Recognition using Spatial-Temporal Analysis and Bag of Visual Words. In *Proceedings of 14th International Conference on Signal-Image Technology & Internet-Based Systems* (SITIS). 697–702.

[68] R. Dasari and C.W. Chen. (2018). Mpeg cdvs feature trajectories for action recognition in videos. In *Proceedings of IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)* (pp. 301-304). IEEE.

[69] M. Tong, Y. Chen, L. Ma, H. Bai and X. Yue. (2018). NMF with local constraint and Deep NMF with temporal dependencies constraint for action recognition. *Neural Computing and Applications*. 1–25. 10.1007/s00521-018-3685-9

[70] F. Najar, S. Bourouis and N. Bouguila. (2019). Unsupervised learning of finite full covariance multivariate generalized Gaussian mixture models for human activity recognition. *Multimedia Tools and Applications*; 1–23.

[71] R. Leyva, V. Sanchez and C.T. Li. (2019). Compact and low-complexity binary feature descriptor and fisher vectors for video analytics. *IEEE Transactions on Image Processing*, 28(12), pp.6169-6184.

[72] H. Wang, A. Kläser, C. Schmid and C.L. Liu. (2013). Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision,* vol. 103, no. 1, pp. 60–79.

[73] V. Kantorov and I. Laptev. (2014). Efficient feature extraction, encoding and classification for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 2593–2600.

[74] S. Narayan and K.R. Ramakrishnan. (2014). A cause and effect analysis of motion trajectories for modeling actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* pp. 2633–2640.

[75] G. Luo, S. Yang, G. Tian, C. Yuan, W. Hu and S.J. Maybank. (2014). Learning human actions by combining global dynamics and local appearance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(12), pp.2466-2482.

[76] J. Uijlings, I.C. Duta, E. Sangineto and N. Sebe. (2015). Video classification with densely extracted hog/hof/mbh features: an evaluation of the accuracy/computational efficiency trade-off. *International Journal of Multimedia Information Retrieval,* vol. 4, no. 1, pp. 33–44.

[77] C. Liu, J. Liu, Z. He, Y. Zhai, Q. Hu and Y. Huang. (2016). Convolutional neural random fields for action recognition. *Pattern Recognition*, 59, pp.213-224.

[78] H. Wang, D. Oneata, J. Verbeek and C. Schmid. (2016). A robust and efficient video representation for action recognition. *International Journal of Computer Vision*, vol. 119, no. 3, pp. 219–238.

[79] S. Wilson and C.K. Mohan. (2017). Coherent and noncoherent dictionaries for action recognition. *IEEE Signal Processing Letters*, 24(5), pp.698-702.

[80] L. Zhang, Y. Feng, X. Xiang and X. Zhen. (2017). Realistic human action recognition: When cnns meet lds. In *Proceedings of 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1622-1626). IEEE.

[81] S. Zhang, C. Gao, J. Zhang, F. Chen and N. Sang. (2018). Discriminative part selection for human action recognition. *IEEE Transactions on Multimedia*, 20(4), pp.769-780.

[82] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar and L. Fei-Fei. (2014). Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. pp. 1725-1732.

[83] K. Simonyan and A. Zisserman. (2015). Very deep convolutional networks for large-scale image recognition. In *Proceedings of International Conference on Learning Representations*.

[84] J.Y.H. Ng, J., M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga and G. Toderici. (2015). Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*

[85] J. Donahue, L. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko and T. Darrell. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*

[86] S. Sharma, R. Kiros and R. Salakhutdinov. (2016). Action recognition using visual attention. In *Proceedings of International Conference on Learning Representations (ICLR) Workshop.*

[87] K. He, X. Zhang, S. Ren and J. Sun. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[88] C. Feichtenhofer, A. Pinz and A. Zisserman. (2016). Convolutional two-stream network fusion for video action recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

[89] G. Varol, I. Laptev and C. Schmid. (2018). Long-term temporal convolutions for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 40(6), pp.1510-1517.

[90] C. Zach, T. Pock and H. Bischof. (2007). A duality based approach for realtime TV-L1 optical flow. In *Proceedings of Joint Pattern Recognition Symposium.* pp. 214-223. Springer, Berlin, Heidelberg.

[91] B. Zhang, L. Wang, Z. Wang, Y. Qiao and H. Wang. (2016). Real-time action recognition with enhanced motion vector CNNs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2718-2726).

[92] Z. Shou, X. Lin, Y. Kalantidis, L. Sevilla-Lara, M. Rohrbach, S.F. Chang and Z. Yan. (2019). Dmc-net: Generating discriminative motion cues for fast compressed video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1268-1277).

[93] L. Zhang, W. Srisukkham, S.C. Neoh, C.P. Lim and D. Pandit. (2018). Classifier ensemble reduction using a modified firefly algorithm: An empirical evaluation. *Expert Systems with Applications*. 93 (2018) 395-422.

[94] D. Pandit, L. Zhang, S. Chattopadhyay, C.P. Lim and C. Liu. (2018). A Scattering and Repulsive Swarm Intelligence Algorithm for Solving Global Optimization Problems. *Knowledge-Based Systems*.

[95] X.S. Yang. (2010). Firefly Algorithm, Levy Flights and Global Optimization. *Research and Development in Intelligent Systems.* 26 (2010) 209–218.

[96] H. Wang, W.J. Wang, H. Sun and S. Rahnamayan. (2016). Firefly algorithm with random attraction. *International Journal of Bio-Inspired Computation*, 8 (1) 33–41.

[97] L. He and S. Huang. (2017). Modified firefly algorithm based multilevel thresholding for colour image segmentation. *Neurocomputing,* 240 (2017) 152-174.

[98] H. Wang, W. Wang, X. Zhou, H. Sun, J. Zhao, X. Yu and Z. Cui. (2017). Firefly algorithm with neighborhood attraction. *Information Sciences*. 382–383 (2017) 374–387.

[99] P. Kinghorn, L. Zhang and L. Shao. (2017). Deep learning based image description generation. In *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, 919-926, 2017.