Neural Computing and Applications
https://doi.org/10.1007/s00521-022-06947-6

ORIGINAL ARTICLE

An evolving ensemble model of multi-stream convolutional neural networks for human action recognition in still images

Sam Slade[1] · Li Zhang[2] · Yonghong Yu[3] · Chee Peng Lim[4]

Received: 6 September 2021 / Accepted: 10 January 2022

Abstract
Still image human action recognition (HAR) is a challenging problem owing to limited sources of information and large intra-class and small inter-class variations which requires highly discriminative features. Transfer learning offers the necessary capabilities in producing such features by preserving prior knowledge while learning new representations. However, optimally identifying dynamic numbers of re-trainable layers in the transfer learning process poses a challenge. In this study, we aim to automate the process of optimal configuration identification. Specifically, we propose a novel particle swarm optimisation (PSO) variant, denoted as EnvPSO, for optimal hyper-parameter selection in the transfer learning process with respect to HAR tasks with still images. It incorporates Gaussian fitness surface prediction and exponential search coefficients to overcome stagnation. It optimises the learning rate, batch size, and number of re-trained layers of a pre-trained convolutional neural network (CNN). To overcome bias of single optimised networks, an ensemble model with three optimised CNN streams is introduced. The first and second streams employ raw images and segmentation masks yielded by mask R-CNN as inputs, while the third stream fuses a pair of networks with raw image and saliency maps as inputs, respectively. The final prediction results are obtained by computing the average of class predictions from all three streams. By leveraging differences between learned representations within optimised streams, our ensemble model outperforms counterparts devised by PSO and other state-of-the-art methods for HAR. In addition, evaluated using diverse artificial landscape functions, EnvPSO performs better than other search methods with statistically significant difference in performance.

Keywords Convolutional neural network · Ensemble model · Human action recognition · Hyper-parameter optimisation · Object detection and classification

1 Introduction

Human action recognition (HAR) aims to identify human actions from visual data. A good HAR model is important in many applications, such as detecting falls, recognising violent behaviours, identifying theft and many other day-to-day activities in various sectors such as healthcare and security. Such potential benefits have led to significant interest in developing robust, accurate, and efficient HAR models. Recent HAR-based solutions cover three main data domains: (1) still images, (2) RGB video streams, and (3)

✉ Li Zhang
li.zhang@rhul.ac.uk

Sam Slade
samuel2.slade@northumbria.ac.uk

Yonghong Yu
yuyh@njupt.edu.cn

Chee Peng Lim
chee.lim@deakin.edu.au

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

[2] Department of Computer Science, Royal Holloway, University of London, Surrey TW20 0EX, UK

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

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

Published online: 30 January 2022

RGB-D video streams. In this respect, video action recognition has attracted significant attention, which takes both spatial and temporal information into account for action classification. However, the extraction of optical flow information requires substantial additional effort, with significant computational cost and complexity. Some of these issues can be overcome by using still images.

In comparison with video HAR, still image HAR has limited sources of information, i.e. only containing spatial information without any temporal cues. In addition, because of viewpoint variations, background clutter, rotations, occlusions, large intra-class and small inter-class variations, still image HAR is a challenging task. Owing to inefficiency in extracting low-level features directly from whole images caused by the aforementioned distracting factors (e.g. cluttered scenes and complex actions), diverse high-level cues, such as human body, body parts, poses, objects, and scene contexts, have been extracted for enhancing performance of still image HAR in existing studies [1]. Traditional non-deep learning based methods derive such high-level cues through multiple pre-processing steps, which lead to high computational costs. As an example, Zheng et al. [2] extracted a combination of human pose and context information for still image HAR, while pose primitive-based HAR was performed by Thurau and Hlavac [3]. Desai et al. [4] and Shapovalova et al. [5] extracted human body, objects, and human–object interaction, while Li and Fei-Fei [6] and Gupta et al. [7] derived human body, objects, and scene contexts for HAR. In addition, body parts, objects, and human–object interaction were used in Maji et al. [8], Desai and Ramanan [9], and Delaitre et al. [10], whereas Sener et al. [11], Yao and Fei-Fei [12], and Yao et al. [13] adopted human body, body parts, objects, and scene contexts.

In the literature, such high-level cues are then characterised by using various low-level features for HAR. As an example, Gupta et al. [7] employed histogram of oriented gradients (HOG), GIST, shape context, colour histogram, and edge distance features, while Li and Ma [14] adopted scale-invariant feature transform (SIFT), HOG, and GIST features. A number of existing studies used both HOG and SIFT features, e.g. Zheng et al. [2], Shapovalova et al. [5], Sener et al. [11], Yao and Fei-Fei [12], Le et al. [15], Yao et al. [16], Delaitre et al. [17], and Qazi et al. [18]. Other studies employed purely HOG features, e.g. Thurau and Hlavac [3], Desai et al. [4], Maji et al. [8], Desai and Ramanan [9], Delaitre et al. [10], and Yao et al. [13], while SIFT features were used purely in Li and Fei-Fei [6], Sharma et al. [19], and Dhulavvagol and Kundur [20].

However, such feature descriptors are subject to various drawbacks. As an example, although SIFT is invariant to scaling, rotation, and illumination changes, it is sensitive to threshold settings [21]. Owing to feature matching, it is computationally costly with large memory consumption [22, 23]. In comparison with SIFT, HOG is not scale and rotation invariant [24, 25]. Its performance degrades when dealing with regions cluttered with noisy edges [26]. Despite the generation of a basic low-dimensional spatial representation of a given image [27], GIST shows significant limitations in capturing fine image details [28][29]. In short, the low-level features extracted by traditional feature descriptors are susceptible to various drawbacks, limiting their discriminative capabilities in tackling still image HAR.

In comparison with traditional methods, deep convolutional neural networks (CNNs) conduct hierarchical layer-wise feature learning in an end-to-end fashion without the requirement of complex computing pipelines. Their feature detectors (i.e. the filters in CNNs) are trainable and highly adaptive. Since the filters learn to adapt to new tasks, CNNs are able to learn bespoke features from a given data set automatically. The machine learned features in earlier layers are similar to those (e.g. edges and corners) yielded by SIFT and HOG descriptors, while the final layers in CNNs are able to produce comparatively more abstract high-level representations (e.g. eyes and wheels). Their efficiency has been ascertained in various HAR tasks in recent years [30–36]. Besides that, CNNs yield superior performances over those of traditional methods in solving diverse other image classification tasks [37–40]. Therefore, we adopt CNNs in this research for still image HAR.

Notably, the configurations of CNN architectures affect model performance. As such, we focus on a well-established architecture, i.e. the VGG19 network [41], in view of its proven efficiency in tackling large-scale image classification tasks. In this research, to adapt such efficient deep networks to an alternative target domain, transfer learning is used to learn CNN feature maps from the new data set whilst keeping the prior learned features. It shows significant capabilities in overcoming data sparsity issues and achieves impressive performance by re-training a pre-trained network using a comparatively smaller data set.

However, to obtain a good balance between preserving the generalisability of the earlier layers and re-training the later layers on the new data set, the capability of identifying suitable transfer learning settings, such as the optimal number of re-trainable layers, poses a great challenge. Other learning hyper-parameters such as the learning rate and batch-size also influence the network performance. Optimising these hyper-parameter settings is challenging, which involves expert knowledge and iterative exploration. It presents a high knowledge barrier that needs focused attention and time. The manual fine-tuning process of hyper-parameters is thus undesirable, which we aim to overcome by using automated search methods.

While the use of a simple grid search can be exploited to identify aforementioned hyper-parameters, it is inefficient as many iterations are necessary. In comparison with such brute-force methods, swarm intelligence algorithms offer capabilities in solving diverse single and multi-objective optimisation problems. Such evolving search algorithms are motivated by observations of natural behaviours, such as ant colonies, beehives, and bird flocks. In this respect, one of the most prevalent algorithms is particle swarm optimisation (PSO). The PSO algorithm is robust for tackling diverse optimisation problems with fast convergence rates. However, owing to reliance on a global best leader, the PSO model is prone to being trapped in local optima. Many PSO variants have been proposed to adjust both exploratory and exploitative aspects of the search to help escaping from local optima in the process of finding the best solution.

In this research, we propose a new PSO variant for hyper-parameter fine-tuning in a transfer learning setting for undertaking HAR tasks with still images. Denoted as EnvPSO, this PSO variant incorporates Gaussian fitness surface prediction and adaptive coefficients to accelerate convergence. It is used to optimise the hyper-parameters of VGG19 deep networks, including the number of re-trained layers in the transfer learning process (denoted as layer strip-back), batch size, and learning rate. Moreover, motivated by the well-known two-stream CNN architecture proposed by Simonyan and Zisserman [31] where features extracted from multi-modal inputs are used for action classification, we design a three-stream based ensemble model with multiple optimised VGG19 networks using EnvPSO for tackling HAR problems. Specifically, in the first stream, we employ an optimised VGG19 network with raw images as inputs. In the second stream, mask R-CNN is first adopted to generate semantic segmentation masks for each input image. The yielded saliency maps are subsequently used as inputs for another optimised VGG19 network for action recognition. In the third stream, a fusion network is constructed by concatenating two VGG19 networks configured in the same manner as the first and second streams. Each of the three CNN streams, denoted as Streams 1, 2, and 3, is optimised independently by the EnvPSO algorithm to identify optimal settings for the learning rate, batch size, and layer strip-back hyper-parameters. These three streams are then combined in an ensemble manner. The final classification results are obtained by taking the average of probabilistic class predictions from the three CNN streams. In other words, the class predictions generated by the optimised CNN streams are summed and divided by the number of streams to produce an average prediction for each input image. A high-level depiction of the proposed EnvPSO-optimised CNN ensemble model is provided in Fig. 1.

Our proposed solution aims to maximise classification accuracy in HAR tasks on still images by taking advantage of diversity of different model architectures and feature inputs. Additionally, the need for expert knowledge and attention required to manually fine-tune a CNN model are overcome by employing a variant of standard PSO to optimise the batch-size, learning rate, and layer strip-back configurations. By incorporating the nonlinear adaptive coefficients and the environmental term embedding Gaussian fitness surface prediction, the proposed EnvPSO model is able to balance well between exploitation and exploration while accelerating convergence. Our research contributions are summarised as follows.

1. A new EnvPSO variant is proposed for automating the fine-tuning process of CNNs. Specifically, EnvPSO introduces three mechanisms to overcome stagnation, i.e. (1) a new optimisation parameter named layer strip-back, which determines the number of layers to be re-trained in the VGG19 networks during transfer learning; (2) nonlinear functions for search coefficient generation which enable the search process to achieve a better balance between diversification and intensification; (3) an additional environmental term embedding a Gaussian fitness surface prediction, which guides the search process towards optimal regions. These three mechanisms work cooperatively to overcome stagnation and automate hyper-parameter fine-tuning of CNNs.
2. An ensemble model with three CNN-based streams is proposed for tackling HAR with still images. Specifically, the first stream employs a VGG19 network with EnvPSO-optimised hyper-parameters, which uses the original images as its inputs. The second stream adopts another VGG19 network with EnvPSO-optimised hyper-parameters, which uses semantic segmentation masks yielded by mask R-CNN as inputs. Such extracted saliency maps from mask R-CNN provide another modality of inputs, which in particular offer better efficiency in representing various action classes (e.g. JugglingBalls, SoccerJuggling, and SkateBoarding) for recognition in human–object interaction. The third stream fuses both VGG19 networks trained with raw images and segmented masks, respectively, by using a flatten and concatenation layer before the fully connected layers. This fused CNN stream helps induce diversity in the learned feature sets extracted from raw images and segmented salient regions. The final classification result for each image is obtained by calculating the mean average of the results from the three streams. The EnvPSO-optimised VGG19 networks with a variety of learning configurations yield better diversity and complementary characteristics to

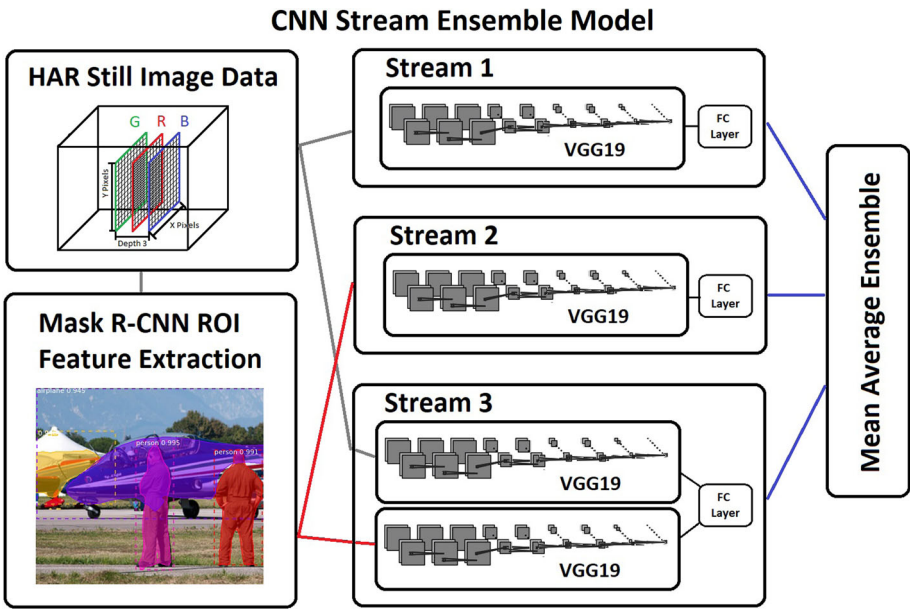

Fig. 1 A high-level representation of the proposed CNN stream ensemble model with the raw images and segmented masks yielded by mask R-CNN as inputs. Stream 1 employs an optimised VGG19 network with raw images as inputs. Stream 2 uses another optimised VGG19 network trained on the saliency maps yielded by mask R-CNN. Stream 3 fuses a pair of optimised VGG19 networks with raw images and segmented masks as inputs, respectively. Each stream is individually optimised using EnvPSO to identify its optimal settings

enhance ensemble model performance, as demonstrated in a series of empirical studies.

The organisation of the remaining part of this paper is as follows. Section 2 presents swarm intelligence-based algorithms such as PSO and its variants, as well as state-of-the-art methods for handling still image-based HAR tasks. In Sect. 3, the proposed EnvPSO algorithm and the ensemble model integrating three EnvPSO-optimised CNN streams are explained. In Sect. 4, the performance of the proposed ensemble model is compared with those from baseline and state-of-the-art methods, along with detailed analysis and discussion of their implications. In Sect. 5, a further evaluation using unimodal and multi-modal benchmark test functions is presented, in order to further evaluate the effectiveness of the proposed EnvPSO algorithm. Conclusions and suggestions for future work are given in Sect. 6.

2 Related work

In this section, we introduce the original PSO algorithm and diverse state-of-the-art PSO variants. Recent studies on HAR are also discussed.

2.1 Particle swarm optimisation

PSO is a useful swarm intelligence algorithm for solving optimisation tasks, such as optimal hyper-parameter selection in CNNs [39, 42–45]. The algorithm works on the assumption that multiple agents can usually find a solution close to the global optima by emulating swarming behaviours found in nature. Its search process is as follows. Firstly, a swarm population in a given search space is initiated. Each particle moves around in the search space by following local and global optimal signals (see Equation 1). A fitness function is used to evaluate the current position of each particle. Specifically, a new velocity is calculated using the inertia weight component, as well as the social- and cognitive-inspired terms. In particular, the social-inspired term establishes a tendency of agents to cluster together to exploit some promising regions of the search space. The cognitive-inspired term promotes a tendency of agents to investigate other optimal areas identified by each particle on its own. To achieve swarming behaviours, each particle records its position with the best fitness score as p_{best_i}, while the best solution found by the overall swarm is recorded as g_{best}. Subsequently, the cognitive-based term is formed as $r_1c_1(p^t_{best_i} - x^t_i)$, which specifically influences the extent an agent conducts search near its own personal best solution. The social-based term is defined as $r_2c_2(g^t_{best} - x^t_i)$, which dictates the extent an agent is compelled to search near the current global best solution. These terms are formalised in Equation 1:

$$v^{t+1}_i = wv^t_i + r_1c_1(p^t_{best_i} - x^t_i) + r_2c_2(g^t_{best} - x^t_i) \quad (1)$$

where v^{t+1}_i is the velocity of the ith particle at the $(t+1)$th iteration and w is the inertia weight defining the contribution of the particle's previous velocity v^t_i towards a new one generated in the next iteration. The personal best solution of particle i at the tth iteration is denoted as $p^t_{best_i}$, while the global best solution of the overall swarm at the tth iteration is represented as g^t_{best}. Parameters r_1 and r_2 are random factors sampled from uniform distribution $U(0, 1)$, while c_1 and c_2 are acceleration coefficients that determine the

contribution of cognitive- and social-based terms, respectively. The next particle position x_i^{t+1} is then obtained using Equation 2 by summing the current particle position x_i^t and new velocity v_i^{t+1}. The pseudo-code of the original PSO model is illustrated in Algorithm 1.

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{2}$$

Algorithm 1 The original PSO algorithm

```
Initialise the swarm size n_p
Initialise a swarm of particles
Initialise the search parameters w, c_1, and c_2
while t < t_max do
    for each particle i = 1, ...., n_p do
        if f(x_i^t) > f(p_best_i) then
            p_best_i = x_i^t
        end if
        if f(x_i^t) > f(g_best) then
            g_best = x_i^t
        end if
    end for
    for each particle i = 1, ...., n_p do
        Update each particle velocity using Equation 1
        Update each particle position using Equation 2
    end for
end while
return g_best
```

2.2 Variants of particle swarm optimisation

The original PSO algorithm shows efficient search capabilities in tackling diverse optimisations problems. Nonetheless, owing to the guidance of single global best leader, the swarm tends to converge prematurely, leading to local optima solutions [46–48]. As a result, many PSO variants have been proposed to tackle the challenges. As an example, Fielding and Zhang [49] proposed a Swarm Optimised DenseBlock Architecture Ensemble (SODBAE) integrated with a PSO variant for image classification. The model was capable of devising CNN architectures with residual connections and dense connectivity to increase network diversity. Specifically, it employed adaptive acceleration coefficients generated using cosine annealing mechanisms to overcome stagnation. Two weight inheritance learning mechanisms were introduced to enable the devised CNN layers to inherit weights from previously optimised ones based on their positions and parameter matrix sizes, with the attempt to reduce computational costs. The model outperformed other state-of-the-art methods as well as manually designed deep networks in a case study with the CIFAR-10 data set.

Nobile et al. [50] proposed a fuzzy self-tuning PSO (FST-PSO) algorithm. It provided fully automated parameter configurations to each particle by integrating fuzzy logic into the PSO algorithm. Two linguistic variables were used to establish fuzzy membership functions, i.e. one for determining the distance between the current particle and global best position as 'close', 'medium', or 'far', while another for measuring fitness improvement of a particle between two successive iterations as 'worse', 'same', or 'better'. These linguistic variables were used in conjunction with a list of rules associated with the inertia weight, social and cognitive search coefficients, and lower/upper clamping values for velocity. Through dynamically adjusting these fuzzy variables, each particle was capable of exploring more promising search regions autonomously. Evaluated on 12 benchmark functions, FST-PSO illustrated fast convergence speed, while maintaining competitive performance, as compared with classical search methods, such as Differential Evolution (DE) and Artificial Bee Colony (ABC).

Tan et al. [40] proposed a PSO variant to optimise hyper-parameters of CNNs as well as cluster centroids of fuzzy C-means (FCM) clustering for skin lesion segmentation. PSO was combined with helix and DE search mechanisms to increase search diversification. A spiral function was used to assign search coefficients to these search operations, while Simulated Annealing (SA) and Levy flight were employed to increase intensification. The model then assigned these local and global search operations in a cascading manner. It started with SA-based local exploitation, and then switched to other search strategies such as PSO, helix or DE actions when the search process became stagnant. In this way, the swarm performed multiple search actions simultaneously in each iteration, in order to diversify the search process. The model was used to not only optimise hyper-parameters of pixelwise CNNs, but also fine-tune the cluster centroids of FCM. The optimised CNN and FCM components formed two separate ensemble models for lesion segmentation. Evaluated using three skin lesion data sets, i.e. Dermofit Image Library, PH2, and ISIC 2017, the devised PSO-based ensemble model illustrated significant superiority over other clustering and deep networks in lesion segmentation.

Singh et al. [51] proposed a multi-level PSO (MPSO) model for optimisation of architectures and hyper-parameters of CNNs. The proposed model exploited the concept of multiple populations. Specifically, the initial swarm at level one was used for CNN architecture generation (i.e. identification of the most optimal settings of convolutional, pooling, and fully connected layers), while multiple populations at level two were subsequently used to optimise hyper-parameters (e.g. number of filters, filter size, and number of neurons) of each CNN from level one. An adaptive inertia weight implemented by a sigmoid function was leveraged to balance diversification and intensification. Evaluated using five well-known data sets, including

MNIST, CIFAR-10, and CIFAR-100, the devised model with optimal hyper-parameters produced an impressive performance.

Bai et al. [52] proposed a dynamic weight PSO-based sine map (SDWPSO) algorithm for optimising weights and biases of a backpropagation neural network (BPNN) for reliability prediction in engineering problems. A new position updating operation was proposed, where dynamic weights were used to adjust the proportions of contributions of the current position, the new velocity and the global best solution for position updating. The sine map with an adaptive control factor was used to adjust the inertia weight. Evaluated using 14 benchmark functions and reliability prediction of turbocharger and industrial robot systems, the model outperformed Support Vector Machine (SVM) and Artificial Neural Network (ANN) methods significantly.

Lan et al. [53] developed a hierarchical sorting swarm optimiser (HSSO) to solve large-scale optimisation problems. HSSO incorporated a new learning strategy to sort the particles into a hierarchical structure based on fitness scores. Specifically, the particles were recursively sorted into groups containing solutions with promising or poor fitness values. Promising particles were used in each subsequent recursion. This hierarchical structure employed elite solutions with promising fitness scores to update the velocities and positions of worst-performing particles. In addition, the personal best solution in the cognitive term was also replaced with those promising solutions in the hierarchical structure. The mean position of the overall swarm was adopted in the social term as opposed to a global best position. Using 39 generic benchmark test functions, HSSO showed improved exploration and exploitation capabilities against those of social learning PSO (SL-PSO), a Competitive Swarm Optimiser (CSO), Efficient Population Utilisation Strategy PSO (EPUS-PSO), Dynamic Multi-Swarm PSO (DMS-PSO), and Multi-level Cooperative Coevolution (MLCC).

Han et al. [54] developed an adaptive gradient multi-objective PSO (AGMOPSO) algorithm to address slow convergence and sub-optimal performance inherent in multi-objective optimisation problems. The main goal of multi-objective optimisation is to achieve a weighting of contribution across all the evaluation functions (objectives) by optimising some target variables. This ideal weighting is known as the Pareto-optimal set. A stock ticker multi-objective gradient (stocktickerMOG) method was devised to approximate the optimal Pareto set of solutions. A unique self-adaptive flight mechanism which affected both social and cognitive terms was introduced. To achieve this, a fixed sized archive was updated with the global best position, provided that it was not dominated by any current entries in the archive. During each PSO iteration, Multi-Objective Gradient (MOG) was used to obtain gradient information so that the archive entries can be incremented towards the Pareto-optimal set. A unique self-adaptive flight parameter was calculated based on the distance between the closest and furthest particles corresponding to the swarm leader as well as the distance between the current particle and global best solution. This flight parameter was applied to each particle differently depending on its dominance state with respect to the current entries in the archive. This allowed each particle to dynamically adapt the amount of contribution from the social and cognitive terms. Evaluated on a series of established multi-objective benchmark functions (ZDT [55] and DTLZ [56]) using the Inverted Generational Distance (IGD) and spacing metrics, AGMOPSO achieved better diversity and accuracy as compared with seven multi-objective PSO algorithms as well as non-dominated sorting genetic algorithm II (NSGA-II) and strength Pareto evolutionary algorithm 2 (SPEA2).

Cai et al. [57] combined PSO with density peaks clustering (PDPC) to address the limitations in manual selection of initial cluster centroids and the influence of a distance cut-off parameter required by density peaks clustering (DPC). The distance cut-off parameter was determined by calculating the Gaussian distances between all data points and taking the mean value of the maximum and minimum Gaussian distances. Initial cluster centroids of DPC were selected using PSO, where the inverse product of density and distance was used as the fitness function. Evaluated using nine UCI benchmark data sets, PDPC showed great superiority in solving cluster centroid selection, yielding promising accuracy, precision, and recall scores in contrast to several methods, including K-means clustering, Improved K-means clustering, original DPC and density peak K-medoids.

The aforementioned PSO variants are useful for tackling issues of premature convergence of original PSO, where stagnation is often attributed to non-optimal exploration and exploitation of the search processes. Many studies change the flight characteristics of the cognitive and social terms. These changes are often applied to the velocity updating operation, as defined in Equation 1, which plays a significant role in determining a particle's search behaviour. The velocity updating operation often incorporates certain new factors to affect the social and/or cognitive terms. In some cases, the inertia weight is adjusted as well, in order to obtain a delicate control of the velocity scale applied to each particle in each iteration. In comparison with the existing studies, EnvPSO has the following contributions, i.e. (1) a new environmental term is introduced, which estimates the fitness surface of unexplored search regions by using a Gaussian filter and information obtained from previously explored search space. It provides each

particle with a sense of environmental awareness to complement the effects of both social and cognitive terms. (2) An exponential function is embedded to adjust the search effects of the social and cognitive terms adaptively in each iteration. (3) A new optimisation parameter (i.e. layer strip-back) is proposed to determine the number of re-trainable layers of each CNN model in the transfer learning process to increase network variations. By adopting adaptive scheduling of the social and cognitive terms as well as providing additional environmental awareness, our model achieves an enhanced trade-off of intensification and diversification. The empirical results indicate its capabilities in identifying hyper-parameters that yield distinctive competent stream ensemble CNN models for undertaking HAR problems.

2.3 Human action recognition

HAR has gained increasing research attention, owing to its broad range of real-life deployments such as healthcare, security, and surveillance [30]. As an example, Sharma et al. [58] presented an expanded parts model (EPM) to tackle HAR problems. The model selected discriminative part templates with an associated scale space location and scored them using a novel SVM-like classifier. A unique scoring function was proposed, which promoted learning diverse spatial and descriptive image patches to best represent the action. The EPM model, when visualised, showed an interesting collage of class relevant image patches spatially overlaid atop the original image with non-relevant parts of the images remaining black. This gave an idea of how the classifier matched parts with relevant aspects in an image to optimise accuracy. Evaluated using the Stanford40 and Human Attributes (HAT) data sets, the EPM model in combination with VGG16-based feature extraction achieved superior mean average precision (MAP) scores as compared with nine other methods, including spatial pyramid matching.

Zhang et al. [59] presented a part-based method called minimum annotation effort (MAE) to handle still image-based HAR tasks. The model included two main components, i.e. delineation of the 'action mask' and a unique feature representation for action classification. Delineating the action mask required two steps, i.e. object parts generation and action mask discovery. To address the first issue, bounding-box based object proposals were obtained using unsupervised selective search and passed through a VGG16 network. A multi-max pooling technique was applied to the outputs from the last convolutional layer of the VGG16 network to yield object parts. To retrieve the action mask, an energy minimisation problem on a Markov random field was formulated. The solution produced a shared global parts model, a part model for each class and action-masks for each image. In addition, feature representation was conducted by applying product quantisation to the initial object proposals that had sufficient overlaps with the action mask. These formed the inputs to a one-vs-all linear SVM classifier for action classification. Evaluated using benchmark still image data sets (such as PASCAL VOC 2012, Stanford40, and Willow7), the model outperformed existing methods such as regularised max pooling (RMP), object bank, locality-constrained linear coding (LLC), and EPM.

Wang and Wang [60] proposed a Joint learning hierarchical spatial sum product network (JHS-SPN) for HAR tasks. A novel feature representation scheme was introduced. Image patches were sequentially extracted from the images. Action features were established by extracting CNN features from these sampled image patches. The feature vectors were clustered and used to fine-tune a CNN model. Multiple SVMs were trained on these feature clusters to produce part activation vectors. JHS-SPN altered the original sum product network (SPN) model by introducing hierarchical partitioning. It learned optimal channels by dividing an image and capturing deformable spatial relationships between object parts. Part activation vectors and spatial relationships were extracted from each image subdivision, in order to reduce the computation complexity. Based on the Willow7 action data set, JHS-SPN produced superior MAP scores as compared with those from EPM, Discriminative spatial Saliency (Dsal), and the interaction pairs method. Evaluated on the Stanford40 data set, JHS-SPN outperformed EPM, LLC, object bank, and spatial pyramid matching methods.

Li et al. [61] proposed attention-based transfer learning for image-video adaptation for both HAR and human interaction recognition. A new human interaction image (HII) data set was introduced. Specifically, the method employed class-discriminative spatial attention maps and a Siamese EnergyNet structure for video classification. Class-discriminative spatial attention maps were generated for each video frame using a pre-trained CNN integrated with gradient-weighted class activation mapping (Grad-CAM). These attention maps were subsequently combined with word embedding vectors derived from the class description. The combined feature vectors were used as inputs to the Siamese EnergyNet. This network comprised four parallel dense CNN layers, which was optimised using both energy loss and triplet loss functions. To boost training efficiency, these four parallel dense CNN layers adopted four different types of inputs, i.e. a ground truth label, a false example, a positive example from a different video clip and an incorrect example with minor differences from the ground truth. The model produced competitive MAP scores on the UCF101 data set against 11 other state-

of-the-art methods. Its superior performance on the HII data set was also demonstrated.

Safaei [62] proposed an ensemble method combining spatio-temporal CNN (STCNN) and zero-shot tensor decomposition (ZTD) to solve still image HAR problems. A novel strategy for generating spatio-temporal features along with STCNN and ZDT models was formulated. A new large-scale image data set, namely UCF-Star, was also introduced. The spatio-temporal feature extraction process was unique as the generated temporal information from still images did not inherently exist. To achieve this, the optical flow vectors across several frames were clustered into quantised groups. Taking an image and its corresponding motion clusters as labels, a CNN was optimised using a spatial loss function to classify the regions as probability distribution over the motion vectors. In effect, this produced vertical and horizontal predicted optical flow information. A 3-channel tensor was produced for each image by combining these optical flow predictions with a saliency map derived from a bottom-up ranking method. These spatio-temporal features were used to fine-tune a VGG16 network pre-trained on ImageNet forming the first part of the ensemble model, i.e. STCNN. The second part of the ensemble model was based on ZTD. It conducted HAR by forming action prototypes, applying Tucker decomposition and then performing classification by calculating the set of joint probability distributions between class labels and each test image. The STCNN and ZTD models were combined using multiple linear regression (MLR). Evaluated using the UCFSI-101 (i.e. extracted frames from UCF101), Willow7, Stanford40, WIDER, and UCF-Star data sets, the MLR ensemble method integrating STCNN and ZTD outperformed object bank, LLC, and multi-region CNN methods, significantly.

Yu et al. [63] proposed a non-sequential CNN (NCNN) to solve still image HAR tasks. The NCNN model added multiple parallel branches of convolutional layers to a pre-trained CNN, in order to separately learn spatial and channel-wise features. An end-to-end trainable ensemble of CNN models incorporating NCNN blocks was formed. This ensemble model was compared against traditional ensemble methods (e.g. majority voting, averaging, and weighted averaging) using three different voting strategies (e.g. tuning weight, hard, and soft voting schemes). An ensemble of VGG16, VGG19, ResNet50, VGG16_NCNN, VGG19_NCNN, and ResNet50_NCNN using the tuning weight voting scheme achieved the best performance on the Willow7 data set.

Liu et al. [64] proposed loss guided activation for still image HAR tasks. A novel human mask loss was introduced for optimising a unique human localisation stream. This stream along with another action classification stream was appended to the final convolutional layer of an Inception-ResNet-v2 network. Such strategies enabled joint predictions on both human action classes and a human localisation heatmap, forcing the learned feature representations to focus on the most action-relevant human subjects in the image. The method showed great superiority over 7 other state-of-the-art methods on the MPII and Stanford40 data sets.

Yan et al. [65] proposed multi-branch attention networks for still image HAR problems. The method leveraged the idea of human attention as applied to viewing images. To achieve this, a soft attention mechanism was devised by adding two branches to a VGG16 model, one branch to capture scene level attention while another to handle region-level attention. A two-step alternating optimisation technique was introduced. The classification and region-level attention parameters were first trained before training those associated with scene-level attention. The method showed great performance on the PASCAL VOC 2012 and Stanford40 data sets.

3 EnvPSO-optimised ensemble CNN model for human action recognition

The proposed ensemble model comprises two main components, i.e. EnvPSO and EnvPSO-optimised CNN stream ensemble model. The CNN stream ensemble model is used to generate class predictions for HAR with still images as inputs. EnvPSO is used to optimise the hyper-parameters of each CNN stream, i.e. the learning rate, batch size, and layer strip-back. Once the CNN streams are optimised, they are trained and used to generate class predictions which are subsequently summed and divided by the number of streams to produce an average prediction for each input image. We describe the key components in the following subsections, leading with the proposed EnvPSO variant. Then, the details of the EnvPSO-optimised CNN stream ensemble model are explained.

3.1 The proposed PSO variant

As previously mentioned, PSO establishes two key elements by stimulating its swarm behaviours, i.e. social and cognitive terms. The social term replicates a collaborative behaviour by influencing the search directions of particles towards the global best solution. The cognitive term guides each particle to move towards its personal best experience. Instead of using fixed coefficients for both terms in a standard PSO algorithm, we aim to fine-tune them, and enhance exploration and exploitation of particles. On the other hand, a standard PSO algorithm does not take environmental factors, such as fitness prediction, into account,

which can be beneficial to complement both social and cognitive terms in accelerating convergence.

Therefore, in this research, a new PSO variant, i.e. EnvPSO, is proposed. It incorporates a new environmental element embedding Gaussian fitness surface prediction, and linear and exponential adaptive coefficients to balance between diversification and intensification. Specifically, linear and exponential functions are used to generate adaptive search parameters that allow the swarm to focus on global exploration in the beginning and local exploitation towards the end during the search process. In other words, adaptive functions are proposed to adjust both social and cognitive terms to gradually move from exploration to exploitation. To complement the social and cognitive terms, a third environmental term is proposed, which estimates the fitness surface of the search space for an input function using a Gaussian distribution. It simulates particles to move towards more promising search regions during the search process, in an attempt to accelerate convergence. Details of EnvPSO are shown in Algorithm 2.

3.1.1 Adaptive coefficients

As indicated in Equation 1, the standard PSO algorithm assigns constant values to the acceleration coefficients, i.e. c_1 and c_2, which guide the search process. In this research, we investigate the effects of adjusting these parameters during the search process. Specifically, we propose linear and exponential functions for search coefficient generation. Equations 3-4 and Equations 5-6 define both linear and exponential formulae, respectively. Moreover, static coefficients are employed in EnvPSO by setting $c_1 = 2.5$ and $c_2 = 2.0$, for performance comparison purpose.

$$c_1 = c_{max} - \frac{c_{max} - c_{min}}{i_{max}} i \tag{3}$$

$$c_2 = c_{min} + \frac{c_{max} - c_{min}}{i_{max}} i \tag{4}$$

$$c_1 = \frac{c_{max} - c_{min}}{1 + e^{\frac{5}{i_{max}}(i - \frac{i_{max}}{2})}} + c_{min} \tag{5}$$

$$c_2 = \frac{c_{min} - c_{max}}{1 + e^{\frac{5}{i_{max}}(i - \frac{i_{max}}{2})}} + c_{max} \tag{6}$$

Algorithm 2 The proposed EnvPSO algorithm

Initialise the swarm size n_p
Initialise a swarm of particles
Initialise the fitness array $\mathbf{A}$
Initialise the fitness hyper-surface $\mathbf{S}$
Initialise the search parameters w, c_1, and c_2
while $t < t_{max}$ **do**
  for each particle $i = 1, ..., n_p$ **do**
    if $f(x_i^t) > f(p_{best_i})$ **then**
      $p_{best_i} = x_i^t$
    end if
    if $f(x_i^t) > f(g_{best})$ **then**
      $g_{best} = x_i^t$
    end if
    Update $\mathbf{A}$ using Equation 7
  end for
  Update $\mathbf{S}$ using Equation 11
  for each particle $i = 1, ..., n_p$ **do**
    Update search coefficients c_1 and c_2 using Equations 3-4 or Equations 5-6, respectively
    Update each particle velocity using Equation 13
    Update each particle position using Equation 2
  end for
end while
return g_{best}

where $c_{max} = 2.5$ and $c_{min} = 0.5$, while i denotes the current iteration and i_{max} represents the maximum number of iterations. Figure 2 illustrates the adaptive search coefficients generated using Equations 3-4 and 5-6, respectively.

Such adaptive linear and exponential coefficients enable the swarm to focus on global exploration at the beginning of the search process and local exploitation towards the end.

Besides adaptive social- and cognitive-based terms, we propose an environmental term pertaining to fitness surface estimation using Gaussian distribution, as explained in the following subsection.

3.1.2 Gaussian fitness surface prediction

To further enhance the exploitation and exploration capabilities of PSO, we introduce a third environmental term to complement both social and cognitive-based terms in the velocity-updating formula. In essence, this new strategy adds an environmental awareness to particles by providing information on the function being evaluated. Since it is not possible to obtain the fitness scores of unevaluated positions in the search space, we can instead estimate the fitness scores associated with vicinity of previously evaluated positions. Using these estimations, we can create a rough landscape of the fitness surface for the input function. As the algorithm progresses, estimation of the complete fitness surface becomes more accurate. Based on the estimated surface, we can extract gradient information to influence the velocity of a particle by pushing it along the direction towards fitter solutions. The extracted gradient information lays the foundation for the proposed third environmental term in accelerating convergence.

A pictorial example of this fitness surface is displayed in Fig. 3. It shows how the landscape of estimated fitness surface changes over time when EnvPSO is used to solve a classic minimisation problem, i.e. the Ackley benchmark function. Initially, the landscape of estimated fitness surface (magenta) appears flat (when $i = 1$ in Fig. 3). When particles explore and evaluate positions of the input function (i.e. Ackley function), the associated gradient information in each dimension of the estimated fitness surface is extracted and exploited to influence their velocity. Notice that the estimated surface does not form a one-to-one representation pertaining to the input function. Instead, the estimated surface is convolved with a dimensionally appropriate Gaussian kernel, in order to smooth the fitness landscape and provide a better approximation of the shape of the input function. This leads to appropriate gradient information to be utilised for influencing velocities of particles in the search process.

Specifically, we generate the gradient information by collecting all the currently evaluated positions in the search space and mapping them to a zero index n-dimensional integer array, where n represents the number of targeted hyper-parameters. Mapping parameters with a continuous domain in this way requires an array of infinite size. To solve this problem, we choose several equidistant points between the maximum and minimum values of the continuous domain to serve as indexes of a particular dimension. Once defined, each value in fitness array $\mathbf{A}$ is initialised to zero. When a particle is evaluated, its fitness value $f(x_i^t)$ is stored in $\mathbf{A}$ at an index corresponding to its current particle position x, as defined in Equation 7.

$$\mathbf{A}(x_i^t) = f(x_i^t) \tag{7}$$

where x_i^t is the position of the ith particle at iteration t. After evaluating all particles in the current iteration, an n-dimensional fitness hyper-surface $\mathbf{S}$ is created by convolving a Gaussian filter over $\mathbf{A}$ using Equations 8, 9, 10, and 11. Firstly, Equation 8 is used to calculate the standard deviation of the Gaussian operation.

$$\sigma_d = \theta \times (max(V_d) - min(V_d)) \tag{8}$$

where σ_d is the standard deviation of dimension d with θ as a predefined smoothing factor. Then, σ_d is used in Equation 9 to generate the Gaussian kernel for convolution operations.

$$G_d(r) = \frac{1}{\sqrt{2\pi}\sigma_d} e^{-\frac{r^2}{2\sigma_d^2}} \tag{9}$$

where G_d is the Gaussian kernel for dimension d and its domain R is defined in Equation 10.

$$R = \{r \mid r \text{ is an integer, and } -4\sigma_d + 0.5 \le r \le 4\sigma_d + 1.5\} \tag{10}$$

The Gaussian kernel in the dth dimension is convolved sequentially along the dth axis of A as indicated in Equation 11.

$$S_d(\tau) = A(\tau) * G_d(\tau) \tag{11}$$

Before updating each particle's position and velocity, its current position x_i^t is used to index a point on the fitness hyper-surface $S_d(\tau)$ generated using Equation 11, from which the gradient information of the surface in each dimension is extracted. The gradient information is calculated using second-order finite central differences, as in Equation 12.

$$\Delta x_{id} = \frac{\mathbf{S}(x_{id} + h) - \mathbf{S}(x_{id} - h)}{2h} \tag{12}$$

where Δx_{id} is the gradient associated with dimension d of the ith particle at an indexed position x. Note that $x_{id} + h$ represents the proceeding neighbouring point of x_{id} at a predetermined distance h, while $x_{id} - h$ indicates an

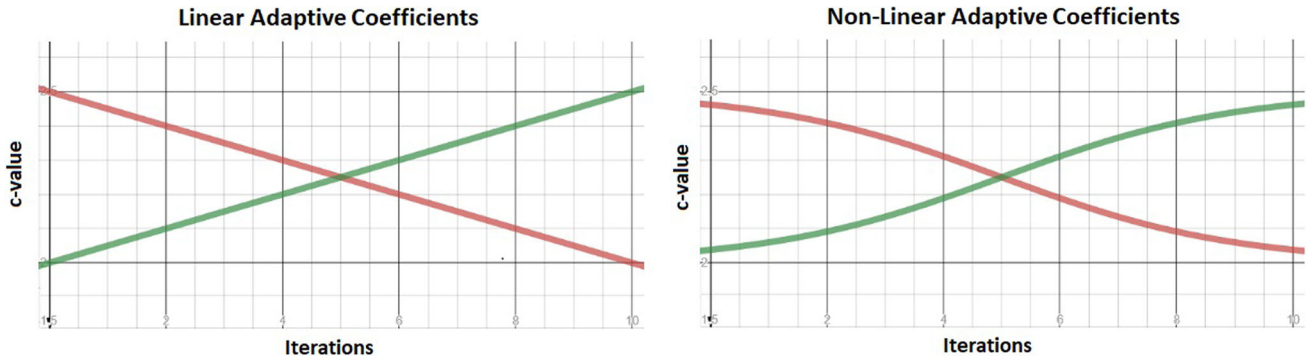

Fig. 2 *Left:* Equation 3 (red line) generates the linear cognitive coefficient c_1 and Equation 4 (green line) generates linear social coefficient c_2.*Right:* Equation 5 (red line) generates exponential cognitive coefficient c_1 and Equation 6 (green line) generates exponential social coefficient c_2 (Color figure online)

indexed position in the opposite direction. Since **S** is indexed with integers incrementing by 1, $h = 1$ is applied to obtain the adjacent position. Figure 4 shows the underlying procedure.

With the gradient information extracted from Equation 12, the environmental term Δx_i^t for the ith particle can be constructed, resulting in a vector of fitness gradient information with length d for velocity updating. Equation 13 is used to update each particle's velocity.

$$v_i^{t+1} = wv_i^t + r_1c_1(p_{best}^t - x_i^t) + r_2c_2(g_{best}^t - x_i^t) + \Delta x_i^t \tag{13}$$

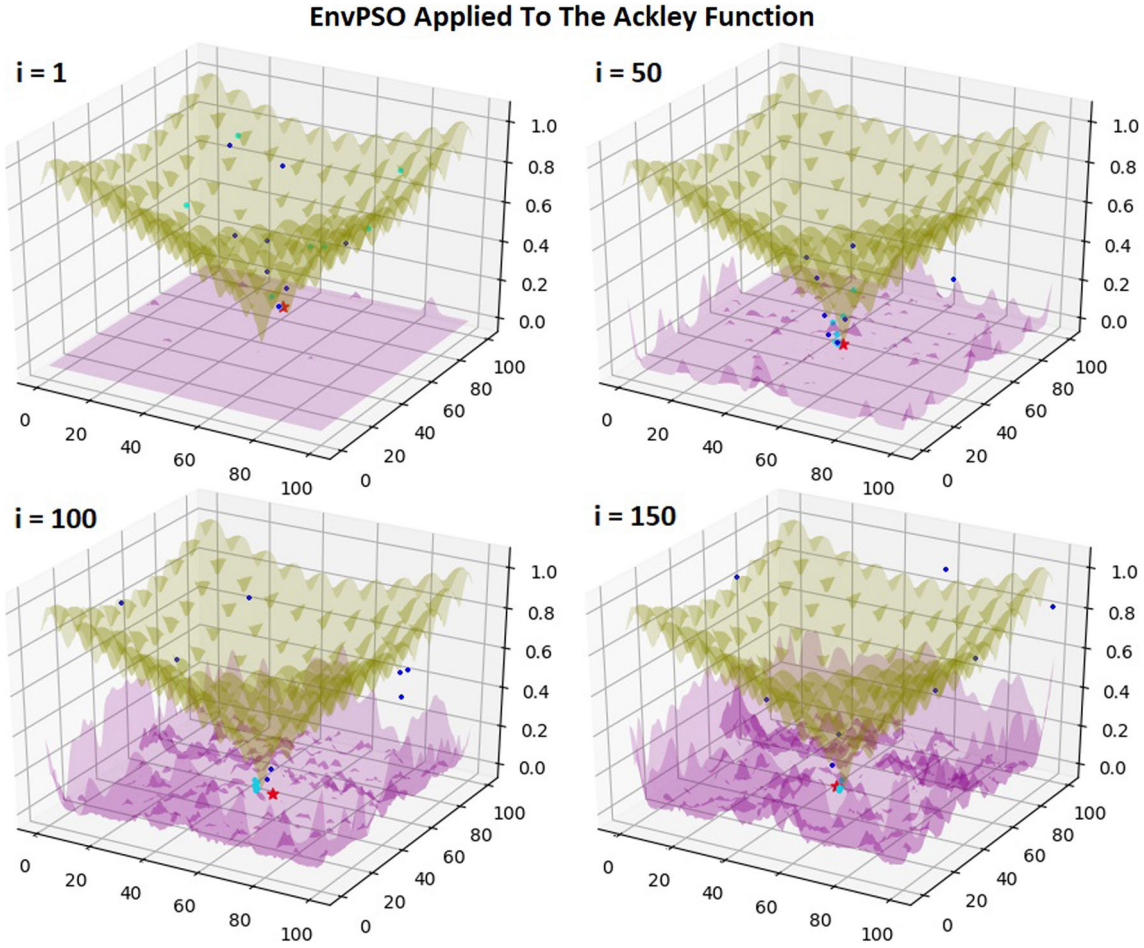

Fig. 3 Variations of the Ackley function (yellow surface) and estimated Gaussian fitness surface (magenta surface) yielded by EnvPSO at iteration i=1, 50, 100, and 150. Blue points indicate current positions of particles, cyan dots show their historical personal best positions, while red star indicates the current global best position (Color figure online)

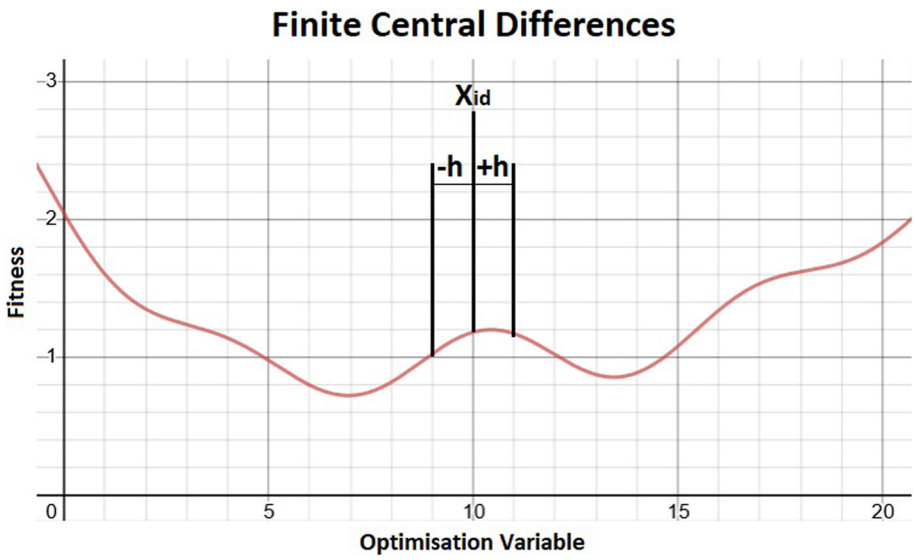

Fig. 4 The use of finite central differences to an arbitrary function, where x_i refers to the ith particle and the red line represents the estimated fitness surface of the function, where $d = 0$ indicates a one-dimensional input. Here h is the step-size in Equation 12, which is set to 1 so that it lines up with the integer indexing scheme of A

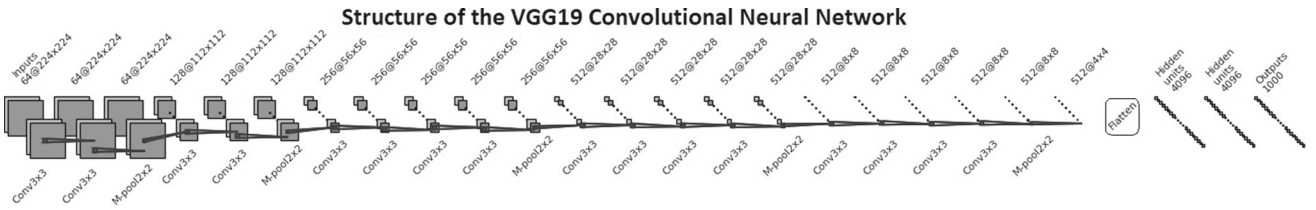

Fig. 5 Layer configurations of the VGG19 network. The ImageNet pre-trained VGG19 models used in the proposed three streams are provided in the Python package tensorflow.keras.application, which require an input shape of (224, 224, 3). Each network is adjusted by replacing the final three dense layers with three new counterparts, where the first and second dense layers have 1000 and 100 neurons, respectively, while the final output layer has neurons equivalent to the target classes in the training set

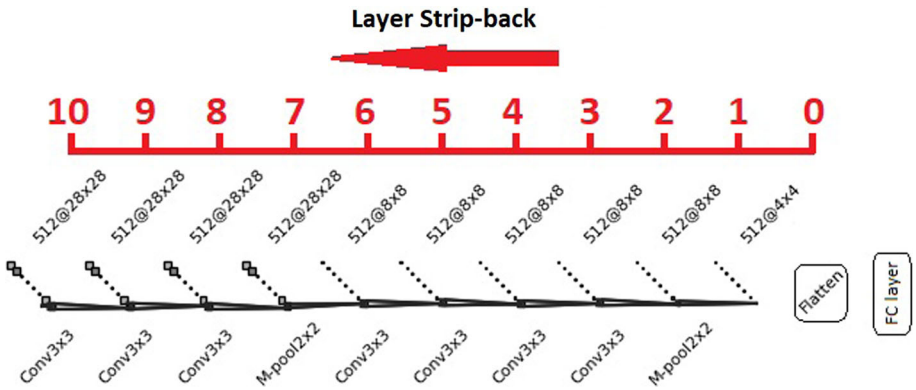

Fig. 6 The layer strip-back parameter is applied to the VGG19 network. Note that zero indicates that no convolutional layer prior to the flatten layer needs to be re-trained

Finally, the new particle velocity is used for updating its position using Equation 2. The proposed Gaussian fitness estimation surface equips the swarm with higher chances in exploring promising search regions, while reducing the risk of being trapped in local optima, in order to accelerate convergence.

The final PSO addition, namely layer strip-back, defines the number of CNN layers to be re-trained in the transfer learning process when EnvPSO is used to optimise network hyper-parameters. An analysis is provided below.

3.1.3 Layer strip-back

Three CNN streams are used to form the ensemble model. The first stream is based on a VGG19 [41] backbone pre-trained on the ImageNet data set. Its structure is displayed in Fig. 5. To optimise matrix calculations and GPU memory allocation when training a pre-trained CNN for a new task, we can manually select a number of layers to be re-trained. By reducing the number of trained layers, we reduce the number of required matrix calculations, leading

to economical use of computation cycles and GPU memory. Rather than manually determining the number of re-trained layers for transfer learning, we automate the layer selection process by creating a variable called layer strip-back, which is presented in Fig. 6.

This variable is assigned an integer value in a range of [0, 10], which determines the number of layers back from the final layer of the backbone network used for re-training. For instance, if the layer strip-back value is 2, then only the last two layers in the network need to be re-trained. This variable is automatically determined, like any other hyper-parameters (i.e. the learning rate and batch size), during the optimisation process. After optimisation with EnvPSO, the proposed CNN ensemble model is used in a multi-stream form for HAR tasks.

3.2 The multi-stream ensemble model

Motivated by the well-known two-stream CNN architecture proposed by [31], where spatial and temporal information was extracted by separate streams for action classification, we propose an ensemble model consisting of three EnvPSO-optimised CNN streams, as shown in Fig. 1, to diversify action recognition. The first stream employs a VGG19 network with raw images as inputs. The second stream adopts another VGG19 network with the segmented masks yielded by mask R-CNN as inputs. The third stream fuses two VGG19 networks with raw images and segmented masks as inputs, respectively. The network in each CNN stream is individually optimised. Specifically, optimal transfer learning settings, which include the learning rate, batch size, and layer strip-back hyper-parameters, are devised using EnvPSO for each stream. The three optimised streams are combined in an ensemble manner using the average of their probabilistic class predictions.

Moreover, the search ranges of the optimised hyper-parameters, i.e. the learning rate, batch size, and layer strip-back, are shown in Table 1. The three optimised hyper-parameters affect network performance. As an example, the learning rate affects model learning behaviours. A very small learning rate is more inclined to be stuck in local optima, which requires substantial training effort to reach optimal solutions. A moderate setting is more likely to result in steady delicate training steps while obtaining promising performances. In addition, the batch size defines the number of samples processed before updating the network parameters. According to Masters and Luschi [66], a suitable batch size ranges between 8 and 32. Since it is highly likely that there are multiple configurations that can produce promising performances, the capability of identifying optimal settings is important. Furthermore, the layer strip-back hyper-parameter determines the number of re-trainable layers in the transfer learning process. A moderate setting can solicit sufficient knowledge from the new domain while taking advantage of prior knowledge learned from the pre-trained domains. A comparatively small setting may not be effective enough to learn sufficient new feature representations (especially when the new domain is very different from the pre-trained domain), which can limit network performance. Therefore, we optimise the learning rate, batch size, and layer strip-back hyper-parameters for each CNN stream using the proposed EnvPSO algorithm. Further details of each stream are explained in the following subsections.

Table 1 Hyper-parameter search ranges

Hyper-parameter	Range
Batch size	[8, 64]
Learning rate	[0.001, 0.01]
Layer strip-back	[0, 10]

3.2.1 Stream 1—VGG19 with Raw Images

The first stream is a VGG19 network [41] pre-trained on the ImageNet data set. Its structure is displayed in Fig. 5. It is adjusted by replacing the original final three dense layers with three new fully connected dense layers, where the first dense layer has 1000 neurons, the second dense layer with 100 neurons, and the final output layer has neurons equivalent to the target classes in the training data set. The input images are resized to (224, 224, 3), in order to match the input shape of the first convolutional layer of the VGG19 network. An overview of this first stream is provided in Fig. 7. In addition, as mentioned above, EnvPSO is used to identify the optimal transfer learning configurations of this CNN stream, i.e. the learning rate, batch size, and layer strip-back hyper-parameters, to better adapt it to the new tasks.

3.2.2 Stream 2—VGG19 with mask R-CNN features

The second stream is composed in a manner similarly to that of the first CNN stream, but differs by the input it receives. Instead of using the resized raw images as inputs, a pre-processing step is applied to the raw images to extract saliency maps via a mask R-CNN [67] pre-trained on the MSCOCO data set. Mask R-CNN uses a Region Proposal Network (RPN) to propose candidate object bounding boxes. Classification and bounding box regression are then performed, while concurrently producing a binary segmentation mask for each class. This allows retrieval of the class probability, the bounding box offset and a binary segmentation mask for each detected object in a given

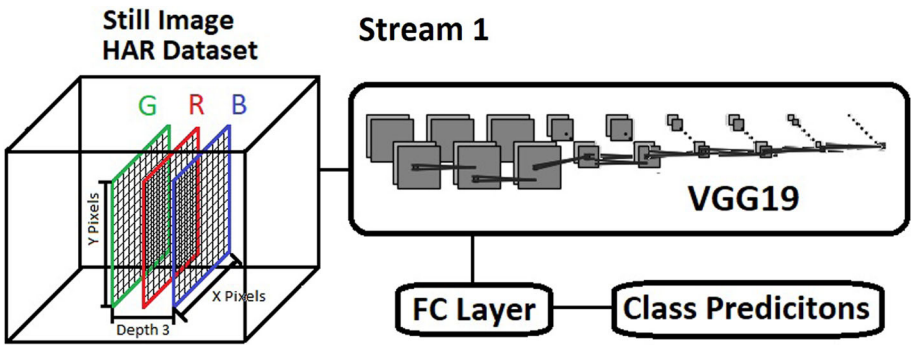

Fig. 7 An overview of the first stream

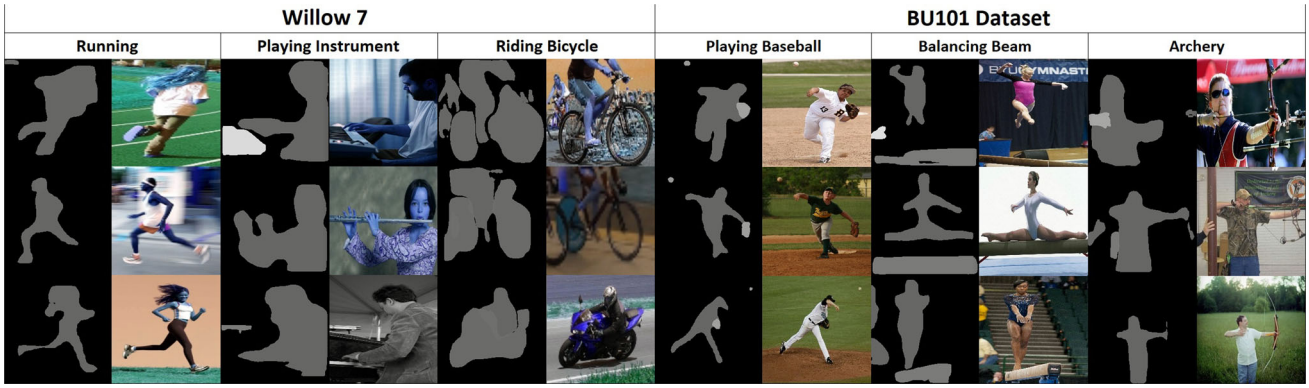

Fig. 8 Examples from three of the 7 classes from the Willow7 data set as well as three of the 101 classes from the BU101 data set. Each column displays three examples of the grey scale images generated using mask R-CNN and their corresponding raw images. Each grey shade represents a different class prediction for the region of pixels it covers in the raw image

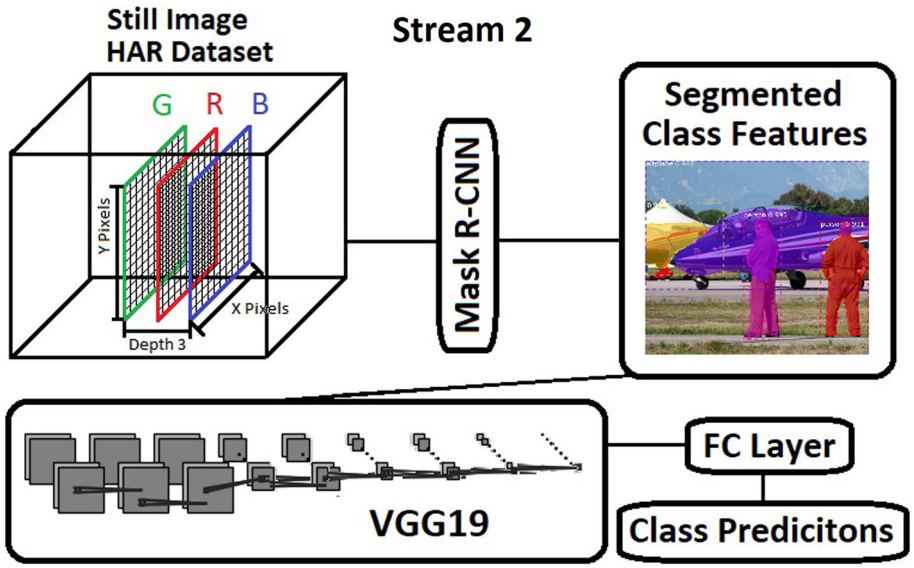

Fig. 9 An overview of the second stream

input image. In addition, each detected class is represented by a particular shade. This pre-processing procedure using mask R-CNN yields a resized grey-scale unsigned 8-bit integer image with class-encoded segmentation masks for all detected objects (see Fig. 8). This output grey-scale image is used as the input to the VGG19 network in Stream 2. In this way, we represent class categories as different shades, allowing previously identified class information to inform subsequent classification. Figure 9 illustrates the overview of this stream.

Using mask R-CNN, we transform raw image inputs into saliency maps containing object and location data, in

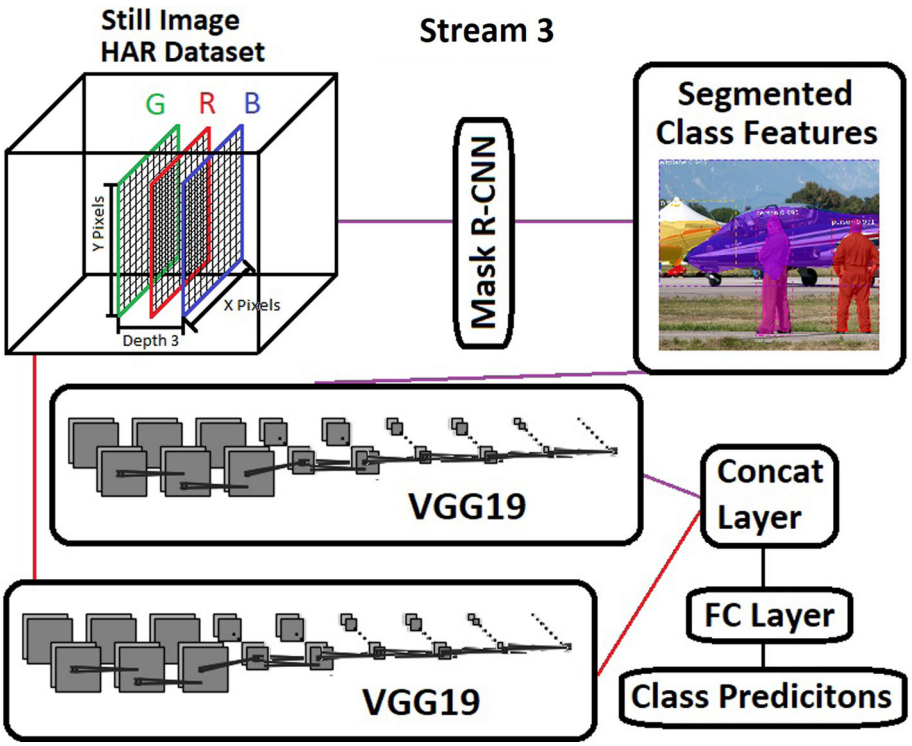

Fig. 10 An overview of the third stream

order to create a new input modality. In particular, applying these inputs to a separate VGG19 network allows this second stream to better represent various actions (e.g. JumpRope, JugglingBalls, PizzaTossing, and SkateBoarding) for recognition in human–object interaction. In addition, EnvPSO is used to identify optimal settings of the learning rate, batch size, and layer strip-back hyper-parameters, with respect to the transfer learning process for this stream.

3.2.3 Stream 3—A fusion of streams 1 and 2

As indicated in Fig. 10, the last stream fuses two VGG19 networks using raw images and segmented masks extracted by mask R-CNN as inputs, respectively. It adds a flattening layer after the final convolutional layer of each network, and concatenates them to form an end-to-end trainable CNN. Its inputs are both raw images as used in Stream 1 and pre-processed saliency maps as adopted in Stream 2. These two types of input images are simultaneously used for training.

The EnvPSO algorithm is used to optimise the learning rate, batch size, and layer strip-back hyper-parameters of this third stream in the transfer learning process. Based on optimised Streams 1, 2, and 3, we construct an ensemble model to overcome bias and variance of single stream to further enhance performance.

3.2.4 Stream ensemble model

As discussed earlier, each constituent stream is optimised independently using EnvPSO to identify the optimal learning rate, batch size, and layer strip-back settings. Specifically, during the training stage, the target stream is trained for three epochs at each EnvPSO iteration. Then, it is evaluated based on a validation set to yield the class predictions. The MAP indicator is used as the fitness score pertaining to the particle's position in the search space. Once the optimisation process is completed, the optimal hyper-parameters are used to train the corresponding CNN stream for 100 epochs. After training, the CNN models are evaluated using the test set, giving the final class predictions. Once all the streams are evaluated, their outputs are combined by taking the average of predictions. Specifically, the class predictions generated by the optimised CNN streams are summed and divided by the number of streams to produce an average prediction for each input image. We repeat this procedure for 10 trials and take the average results, in order to avoid randomness in CNN training. The mean MAP result over 10 runs is used for performance comparison, as indicated in Fig. 11. This multi-stream EnvPSO-optimised ensemble model is illustrated in Algorithm 3. Such an ensemble strategy is not only able to embed distinctive transfer learning strategies in different streams to increase diversity, but also to strengthen weak base learners and overcome bias and

variance of optimised base networks for performance enhancement.

Moreover, for the aforementioned ensemble model, there is only one pre-processing step required, i.e. semantic segmentation mask generation using Mask R-CNN. As indicated in Section 3.2.2, Mask R-CNN is used to extract semantic segmentation masks from raw images. The extracted saliency maps are used as the inputs to CNNs in Streams 2 and 3, as indicated in Figs. 9, 10. These segmented masks provide a new type of inputs in comparison with raw image inputs used in other CNNs, in order to increase model diversity. In particular, they are used to better represent actions with respect to human–object interaction.

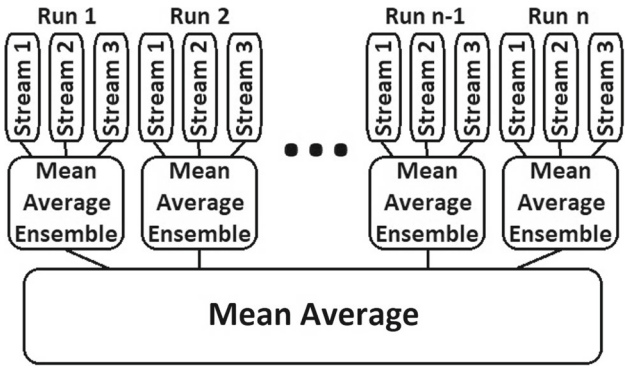

Fig. 11 Construction of the ensemble model, where the class predictions of each stream are combined using the mean average

stream as well as the ensemble model of every possible permutation of the streams. Streams with default hyper-parameter settings instead of optimised ones are also provided to highlight the performance of the optimised streams and ensemble models. In addition, we compare the MAP results with those from other state-of-the-art existing methods.

The following settings are followed, in order to ensure consistency in experiments. Every CNN stream is trained with a stochastic gradient descent optimiser using a categorical cross-entropy loss function, as well as a Nesterov momentum of 0.01 and a decreasing learning rate that reduces by 1/5 when the validation loss does not improve over three consecutive epochs. The settings of static, linear,

Algorithm 3 The Proposed Multi-stream CNN Ensemble Model with EnvPSO-Optimised Hyper-parameters

runs = 10
run = 0
while run < runs **do**
  for stream = 1, 2, 3 **do**
    Conduct hyper-parameter optimisation for each stream using EnvPSO in Algorithm 2
  end for
end while
run = 0
while run < runs **do**
  for stream = 1, 2, 3 **do**
    Train each optimised CNN model in each stream using a larger number (i.e., 100) of epochs
    Evaluate the trained network using the test set for each stream
  end for
  Calculate the mean result of the three streams in each ensemble model to yield final class prediction
end while
Take mean average of the MAP results

4 Evaluation of the ensemble model with HAR data sets

In this section, we evaluate the proposed three-stream ensemble model with EnvPSO-optimised hyper-parameters using two HAR data sets, i.e. the Willow7 [17] and BU101 [68] data sets. To better understand the impact of additional contributions to original PSO, we evaluate each proposed strategy separately. Specifically, we compare the MAP scores of CNN models trained with hyper-parameters optimised by PSO and EnvPSO using static, linear and nonlinear search coefficients in each individual CNN

and exponential search coefficients used in PSO and EnvPSO are shown in Table 2. The identified optimal configurations by PSO and EnvPSO are used to train corresponding CNN ensemble models on each data set. The trained ensemble models are subsequently evaluated on the unseen test set. We adopt the following settings throughout the experiments, i.e. population=10, maximum number of iterations=30, and dimension=3. In addition, a total of 10 runs are performed to construct 10 optimised stream ensemble models. The mean results of the 10 stream ensemble models are used for performance comparison. In addition, the default networks without any optimisation process purely re-train the last three layers using the new data set, instead of using the dynamic number of layers recommended by the layer strip-back parameter. Such default networks employ a default learning rate of 0.001 and a default batch size of 32. The mean result of the default ensemble model over 10 runs is computed for performance comparison based on each data set.

As mentioned earlier, for multi-stream ensemble models with both optimised and default parameter settings, there is only one pre-processing step required, i.e. semantic segmentation mask generation using mask R-CNN. Specifically, mask R-CNN is used to extract semantic segmentation masks from raw images. These extracted saliency maps are then used as the inputs for CNNs in Streams 2 and 3, as indicated in Figs. 9, 10. Except for the aforementioned segmentation mask generation, there is no other pre-processing step required. These segmented masks create a new input type in comparison with raw image inputs used in other CNNs to increase model diversity.

4.1 Data sets

We use the following two key data sets that have been used in several related studies.

4.1.1 Willow7

The Willow7 data set [17] consists of 7 classes containing 968 images extracted from Flickr. The classes are, 'Interacting with Computer', 'Photographing', 'Playing Instrument', 'Riding Bike', 'Riding Horse', 'Running', and 'Walking'. We employ the official train, validation, and test data splits for each class category in our experiments.

4.1.2 BU101

The BU101 data set [68] comprises 23.8K manually filtered web images pertaining to actions from 101 classes. These action classes are divided into five categories, i.e. human–object interaction, body-motion only, human–human interaction, playing musical instruments, and sports. The action classes in BU101 have a 1-1 correspondence with those of the UCF101 video action data set. Some example classes are, 'MoppingFloor', 'PullUps', 'Knitting', 'SkateBoarding', and 'Typing'. In addition, a total of 2769 images are taken from Stanford40, which share the same class categories (e.g. 'PlayingViolin' and 'Rowing') as those in UCF101. Each class in the BU101 data set contains 100-300 images extracted from the above sources. This data set does not have an official train/test data split. We use a train/validation/test split of 70/10/20, as adopted in other existing studies [61]. Specifically, we apply the above split to each class so that we obtain the same ratio of class samples to form train/validation/test sets.

4.2 Results

The MAP metric is computed to determine the effectiveness of the EnvPSO-optimised CNN ensemble model. The mean results of 10 separate runs using the Willow7 and BU101 data sets are shown in Tables 3 and 4, respectively. The numbers in the first row of these tables refer to which streams are being ensembled to obtain the final predictions. Static, linear, and nonlinear refer to constant, linear, and nonlinear (exponential) search coefficients, respectively.

In Tables 3 and 4, Streams 1, 2 and 3 represent optimised VGG19 with raw images as inputs, optimised VGG19 with extracted mask R-CNN salient features as inputs, and fusion of both Streams 1 and 2, respectively. As illustrated in Tables 3 and 4, ensemble models with default and optimised settings combining Stream 1/Stream 2 with Stream 3 achieve enhanced performances, indicating that additional diversity introduced by Stream 3 offers significant advantage over those individual streams. In addition, ensemble models of Streams 1 and 3 typically achieve the best performance with both data sets for nearly all search methods. The most effective configuration for both data

Table 2 EnvPSO and PSO settings

Method	Value
Static	Cognitive acceleration coefficient $c1 = 2.5$ Social acceleration coefficient $c2 = 2.5$ Inertia weight $w = 0.1$
Linear	Linear cognitive and social search coefficients generated using Equations 3 and 4 with $c_{max} = 2.5$ and $c_{min} = 0.5$, Inertia weight $w = 0.1$
Nonlinear	Nonlinear exponential cognitive and social search coefficients generated using Equations 5 and 6 with $c_{max} = 2.5$ and $c_{min} = 0.5$, Inertia weight $w = 0.1$

Table 3 The mean MAP results over 10 runs for the CNN stream ensemble models with optimised and default hyper-parameter settings using the Willow7 data set. (The '+' symbol indicates the streams that have been ensembled.)

Stream	Optimised							Non-optimised
	PSO			EnvPSO				Default(%)
	Static (%)	Linear (%)	Nonlinear (%)	Static (%)	Linear (%)	Nonlinear (%)	Stream Avg. (%)	
1	64.6	**66.0**	68.4	69.2	72.8	76.2	69.53	61.5
2	49.3	58.1	59.0	60.7	63.0	72.4	60.42	39.7
3	62.2	61.0	67.3	**75.5**	75.7	76.5	69.70	62.1
1+2	62.4	65.9	66.5	66.0	74.7	71.4	67.82	60.4
1+3	**64.9**	**66.0**	**69.4**	73.6	**76.2**	**76.8**	71.15	63.4
2+3	59.2	64.0	64.3	70.1	71.4	75.5	67.42	**64.0**
1+2+3	63.5	65.1	67.5	70.3	76.0	73.3	69.28	63.4
MAP Avg.	60.87	63.73	66.06	69.34	72.83	74.59	67.90	59.21
Total Avg	63.55			72.25				

Bold indicates the best results

Table 4 The mean MAP results over 10 runs for the CNN stream ensemble models with optimised and default hyper-parameter settings using the BU101 data set. (The '+' symbol indicates the streams that have been ensembled.)

Stream	Optimised							Non-optimised
	PSO			EnvPSO				Default(%)
	Static (%)	Linear (%)	Nonlinear (%)	Static (%)	Linear (%)	Nonlinear (%)	Stream Avg. (%)	
1	83.6	84.6	85.6	88.8	89.1	88.9	85.80	72.5
2	61.8	66.0	66.1	72.0	62.2	70.6	65.47	29.9
3	85.7	86.6	86.8	88.8	88.5	89.6	87.33	74.1
1+2	82.3	84.1	82.7	88.0	87.1	88.2	84.52	71.0
1+3	**86.6**	**87.1**	**87.5**	**89.6**	**89.5**	**89.7**	87.82	**76.4**
2+3	82.4	84.6	85.7	88.2	86.2	88.6	85.53	71.4
1+2+3	85.5	86.3	86.4	89.6	88.8	89.5	87.15	76.0
MAP Avg.	81.21	82.76	82.97	82.46	84.49	86.44	83.37	67.33
Total Avg	82.31			84.46				

Bold indicates the best results

sets is the ensemble model of Streams 1 and 3 optimised by EnvPSO with nonlinear adaptive coefficients, where the proposed strategies such as Gaussian fitness surface prediction and adaptive exponential coefficients work cooperatively to enhance local and global search capabilities, as compared with the original PSO algorithm.

Notably, the networks of Stream 2 with optimised and default settings show poor performance in comparison with those of Streams 1 and 3. This could be owing to a reduction of available information in the segmented mask image features, since many aspects of original images are removed including colour and local pixel information within the segmented areas and backgrounds. Despite this missing information, the networks still manage to classify over 50% of the class instances correctly using this method alone in most test cases. This suggests that processing raw images with mask R-CNN is able to produce salient features that benefit the classification tasks. Stream 3, however, does not suffer from this problem as both inputs (i.e. mask R-CNN extracted salient features and raw images) are combined through the two fused VGG19 networks, allowing the resulting networks to access more information. This is reflected in the results for optimised networks of Stream 3 revealing the second highest stream average results of 69.70% and 87.33% for the Willow7 and BU101 data sets, respectively. The ensemble models constructed by optimised Streams 1 and 3 produce scores similar to or better than those of Stream 3, as indicated by the stream average results of 71.15% and 87.82% for Willow7 and BU101, respectively; the highest of all the stream average

results. In other words, by ensembling optimised Streams 1 and 3, a consistent enhancement in performance with respect to both data sets is achieved. A similar observation has also been obtained for the ensemble models with default settings incorporating Streams 1 and 3.

Analysing the average results of static, linear, and nonlinear coefficients for both original and proposed PSO algorithms reveals that the proposed nonlinear exponential formulae for search coefficient generation contribute towards a more optimal configuration in exploration and exploitation pertaining to hyper-parameter search. In other words, the results of both PSO and EnvPSO using adaptive exponential search coefficients show consistent enhancement in most test cases.

The average results for all EnvPSO-optimised streams are 72.25% and 84.46% for Willow7 and BU101, respectively. In contrast, the corresponding mean results of all the PSO-optimised streams are inferior, i.e. 63.55% and 82.31%, for Willow7 and BU101, respectively. The differences between these EnvPSO and PSO results are therefore 8.7% for Willow7 and 2.15% for BU101. These differences highlight the overall superiority of the EnvPSO optimised streams over those optimised by the baseline PSO method. Owing to adoption of the Gaussian surface prediction function, the search process of EnvPSO is better guided and is capable of exploring and exploiting optimal regions more thoroughly with better chances of attaining global optimality. In addition, Gaussian surface prediction in conjunction with adaptive exponential search coefficients further diversifies the search process with more balanced local and global search operations for hyper-parameter search, while accelerating convergence. Our resulting hyper-parameters show greater efficiency in re-training VGG19 networks for undertaking HAR problems.

Moreover, a marked improvement in MAP scores is observed by comparing the EnvPSO or PSO optimised streams with those from default settings without any optimisation process for both Willow7 and BU101 data sets. Specifically, as indicated in Tables 3 and 4, the average results of all EnvPSO and PSO optimised streams are 67.9% and 83.37% for Willow7 and BU101 data sets, respectively. The corresponding mean results of the default streams are 59.21% and 67.33%, respectively. As such, the differences between the optimised and default results are 8.69% for Willow7 and 16.04% for BU101. This indicates that the optimisation process improves the network efficiency, producing better generalised solutions. This is owing to the fact that in the default networks, the transfer learning process purely focuses on re-training the last three layers. In contrast, a dynamic number of layers is recommended by the optimisation process to enhance feature learning capabilities and better adapt the yielded networks to a new domain. Moreover, in comparison with EnvPSO and PSO devised ensemble networks with diverse base model configurations, the default ensemble networks employ fixed base model settings, i.e. a fixed number (3) of re-trained layers in combination with a fixed learning rate (0.001) and a fixed batch size (32), which constrain ensemble diversity, therefore limiting their performance.

4.2.1 Hyper-parameter selection

We analyse the identified mean optimal hyper-parameters for Stream 1 CNN models as an example case study to indicate efficiency of the proposed EnvPSO model. Tables 5 and 6 show the selected mean hyper-parameters for Stream 1 CNN models over 10 runs for each search method on the Willow7 and BU101 data sets, respectively.

Referring to Table 5 for the Willow7 results, comparing EnvPSO and PSO in static, linear, and nonlinear coefficient settings reveals that the average layer strip-back configurations identified by EnvPSO are consistently higher. Such higher layer strip-back settings from EnvPSO offer better capabilities for re-training the network on the new data sets without interfering with the useful filter configurations in earlier layers. In comparison with larger and smaller learning rates yielded by PSO with constant and adaptive coefficients, EnvPSO produces moderate learning rates, leading to a better trade-off between performance and convergence speed. These optimal settings, i.e. larger layer strip-back configurations and moderate learning rates, account for the better MAP results from Stream 1 CNN models from EnvPSO, as illustrated in Table 5.

The best configuration is EnvPSO with nonlinear adaptive coefficients, producing a moderate mean learning rate and the highest layer strip-back setting amongst all

Table 5 Average hyper-parameters identified by each search method for Stream 1 CNN models over 10 runs on the Willow7 data set

Variant	Batch size	Learning rate	Layer strip-back	MAP (%)
EnvPSO stat	34.50	0.0051	4.7	69.2
EnvPSO lin	35.40	0.0056	5.9	72.8
EnvPSO nonlin	35.20	0.0053	6.1	76.2
PSO stat	39.40	0.0059	4.6	64.6
PSO lin	39.90	0.0036	4.6	66.0
PSO nonlin	36.90	0.0043	4.4	68.4

Table 6 Average hyper-parameters identified by each search method for Stream 1 CNN models over 10 runs on the BU101 data set

Variant	Batch size	Learning rate	Layer strip-back	MAP (%)
EnvPSO stat	14.5000	0.0084	4.2	88.8
EnvPSO lin	15.1000	0.0082	5.6	89.1
EnvPSO nonlin	13.3000	0.0070	4.5	88.9
PSO stat	8.7000	0.0069	3.2	83.6
PSO lin	9.8000	0.0076	3.5	84.6
PSO nonlin	11.9000	0.0064	3.6	85.6

methods. In contrast, the worst configuration is PSO with static coefficients, which yields a smaller mean layer strip-back setting with the largest average learning rate. Such settings result in a fast convergence to sub-optimal solutions as well as poor acquisition of new domain knowledge and discriminative characteristics, as indicated by the lower MAP results in Table 5.

Next we analyse the identified average hyper-parameters of each search method for the Stream 1 CNNs with respect to BU101 in Table 6. Again, the EnvPSO models with both static and adaptive coefficients produce larger layer strip-back settings than those from PSO. This further indicates that EnvPSO consistently identifies a stronger correlation between enhanced results and comparatively more re-training of network layers in the transfer learning process. The best configuration is EnvPSO with linear coefficients, which extracts the highest mean layer strip-back and batch-size settings, as well as a moderate average learning rate. Such optimal settings enable better re-training of network using the new data set as well as better efficiency in extracting spatial patterns in each batch of this comparatively larger and more complex data set. On the contrary, PSO with static coefficients yields the smallest layer strip-back and batch-size settings, therefore the lowest performance amongst all methods. Since the training set of BU101 is larger than that of Willow7, there are larger numbers of batches in the BU101 training set than those in the Willow7 training set. Therefore, comparatively smaller

Table 7 HAR methods on Willow7

Studies	Methodology	MAP
Zhang et al. [59]	MAE	75.31%
Yu et al. [63]	Deep ensemble learning voting strategy (DELVS3) using tuning weight voting on 6 deep learning models	73.69%
Yu et al. [63]	DELVS2 using tuning weight voting on 3 deep learning models, i.e. VGG16_NCNN, VGG19_NCNN, and ResNet50_NCNN	71.89%
Delaitre et al. [10]	A locally order-less spatial pyramid bag-of-features model using action-specific body parts and object interaction representations	71.70%
Safaei and Foroosh [69]	Ranked saliency map and predicted optical flow + STCNN	71.60%
Safaei and Foroosh [69]	STCNN + intermediate feature space tensor Q	66%
Sharma et al.[58]	EPM with additional context (EPM + context)	67.60%
Sharma et al.[58]	EPM without context of 1.5x extension of bounding boxes	66.00%
Sharma et al.[70]	Discriminative spatial saliency with max margin classifier	65.90%
Wang and Wang [60]	Sum-product network (SPN) with classification by the most probable explanation (MPE) method.	48.70%
Wang and Wang [60]	Flat spatial SPN (FS-SPN).	65.30%
Wang and Wang [60]	Individual learning hierarchical spatial SPN (IHS-SPN).	71.30%
Wang and Wang [60]	Joint learning hierarchical spatial SPN (JHS-SPN). This method is the same as IHS-SPN except that it learns the weights of the shared edges and images between SPNs from two different classes.	71.70%
Wang and Wang [60]	Spatial pyramid matching as proposed by Lazebnik et al. [71]	63.70%
Ours	Multi-stream ensemble with EnvPSO-based hyper-parameter optimisation	**76.80%**

Table 8 HAR methods on BU101

Studies	Methodology	MAP
Li et al. [61]	ResNet101 pre-trained on ImageNet	88.30%
Safaei and Foroosh [69]	STCNN	70.06%
Safaei et al. [72]	a two-stream spatio-temporal network (TSSTN)	72.8%
Alraimi [73]	VGG11 + visual word embedding.	81.70%
Alraimi [73]	VGG13 + visual word embedding. This configuration is the same as the VGG11 + visual word embedding with a different backbone network (VGG13).	77.80%
Alraimi [73]	VGG11 pre-trained on ImageNet	73.40%
Alraimi [73]	VGG16 + visual word embedding. This configuration is the same as the VGG11 + visual word embedding with a different backbone network (VGG16).	58.10%
Alraimi [73]	VGG16 pre-trained on ImageNet	56.60%
Safaei [62]	STCNN with prior knowledge	72.30%
Safaei [62]	ZTD with prior knowledge	71.16%
Safaei [62]	VGG13 pre-trained on ImageNet	70.02%
Safaei [62]	ZTD without prior knowledge	68.24%
Ours	Multi-stream ensemble with EnvPSO-based hyper-parameter optimisation	**89.70%**

batch sizes are identified by both EnvPSO and PSO for BU101 than those of Willow7.

In short, under both static and adaptive coefficient settings, EnvPSO selects higher layer strip-back configurations on average as compared with those yielded by PSO in both data sets for Stream 1 CNN models. These findings indicate that EnvPSO is capable of optimising the layer strip-back parameters to fine-tune more CNN layers during re-training. Combined with moderate and higher average learning rates, EnvPSO is able to conduct better re-training of CNN streams and extract better new domain knowledge from the data samples, while providing better generalisation in dealing with unseen test samples without succumbing to over-fitting or under-fitting issues. Similar characteristics of identified hyper-parameters are obtained for optimisation of VGG19 networks in Streams 2 and 3, where EnvPSO yields larger layer strip-back and moderate learning rate configurations.

In comparison with the optimal settings identified by EnvPSO and PSO, the networks with default settings adopt a comparatively smaller number (i.e. 3) of re-trained layers in combination with a smaller learning rate (i.e. 0.001), which extract limited domain knowledge and discriminative characteristics, therefore compromising the model performance.

We now compare the devised CNN stream ensemble model using EnvPSO with adaptive exponential coefficients against state-of-the-art methods on both Willow7 and BU101 data sets, as shown in Tables 7 and 8, respectively.

Table 7 illustrates the comparison for the Willow7 data set. Each existing study shown in Table 7 employs the overall data set for evaluation. As illustrated in Table 7, our devised CNN stream ensemble model achieves an MAP score of 76.8%, outperforming all existing methods on the Willow7 data set. Our optimised three CNN streams illustrate significant diversity, as evidenced by the identified different layer strip-back and learning configurations. Such distinctive model settings enable the extraction of different internal feature representations, providing complementary properties to enhance ensemble model performance. In addition, the best baseline method is the MAE model [59], with an MAP result of 75.31%. This MAE model uses various techniques (such as Markov random field) to extract a contextual segmentation mask that links a person and the object being interacted with, in order to enhance classification performance. In our approach, we use a similar saliency extraction method based on mask R-CNN, where the segmented regional images provide context for the person and related objects. Besides the above, other strategies such as adoption of multiple types of inputs, hyper-parameter fine-tuning of stream CNNs and ensembling mechanisms are able to enhance performance. Therefore, our approach leads to better robustness than those of [59].

The second-best baseline method is DELVS [63], where six base methods are embedded to yield 73.69% of mean MAP. The model proposes a tuning weight voting ensemble method to integrate the results of the following six base methods, i.e. VGG16, VGG19, ResNet50, VGG16_NCNN, VGG19_NCNN, and ResNet50_NCNN.

The ensemble method achieves promising performance by taking advantage of diverse deep networks and their potential to produce different internal representations with respect to training data. In comparison, our ensemble model achieves better diversification using both backbone networks and input data. EnvPSO is first used to devise optimal network and learning settings for each stream CNN model. Besides using original input images, saliency maps yielded by mask R-CNN are exploited as inputs in our CNN streams. In this way, our ensemble model incorporates distinctive base networks with different learning behaviours as well as diverse input channels for tackling HAR tasks.

We subsequently compare our optimised CNN stream ensemble model with existing studies in Table 8 for BU101. Since there is no official test/train split for the BU101 data set, Table 8 shows an estimated indication of model performance. EnvPSO-optimised CNN stream ensemble model achieves a mean MAP score of 89.7% indicating superior performance against those from existing methods. Owing to the optimised transfer learning process using EnvPSO supported by the layer strip-back parameter, our approach is able to fine-tune different numbers of re-trainable layers to better extract discriminative features and distinguish subtle variations of different action classes. Furthermore, we adopt a stream ensemble model incorporating diverse optimised base networks with both raw images and segmented salient regional proposals as inputs to diversify the ensemble operation. Our yielded CNN stream ensemble models therefore possess better robustness and diversity, as compared with those from the existing methods. In addition, Li et al. [61] and Alraimi [73] employed ResNet101 and VGG11/13 models with embedding strategies and obtained promising performances. However, these models (and most of existing methods) employ a standard transfer learning process without applying any adaptive re-training mechanism to dynamically adjust the number of re-trainable layers. In addition, the use of automatic hyper-parameter fine-tuning and/or salient regional features as additional input is not available in [61] and [73]. These models also do not perform ensemble of distinctive optimised networks equipped with diverse learning options and different input contexts, therefore limiting the performance.

We present a theoretical analysis between EnvPSO and PSO, as follows. EnvPSO incorporates a new environmental term embedding a Gaussian fitness estimation surface as well as exponential adaptive coefficients to balance the search process and accelerate convergence. Specifically, the environmental term yielded from the gradient information of Gaussian fitness estimation surface adjusts the velocity of particles towards more promising search regions, leading to optimal discovery of hyper-parameter configurations. As such, it produces streams with better generalisation capabilities. By implementing exponential adaptive coefficients, EnvPSO illustrates a greater ability to tailor its exploration and exploitation to overcome local optima traps, leading to efficient CNN streams with effective network and learning settings. Furthermore, the introduction of layer strip-back parameter provides a unique way to optimise the number of layers to be fine-tuned. These proposed mechanisms work cooperatively to mitigate premature convergence and account for superior performance of our proposed ensemble model. In contrast, standard PSO employs a single leader-based search process. Without the fitness estimation surface as additional guidance, it is more likely to become stagnant, leading to sub-optimal hyper-parameters. Such settings of comparatively less efficient layer strip-back configurations fail to train a sufficient number of CNN layers to form a better generalised representation of training data. As a result, it extracts limited domain knowledge, which in turn affects the performance of the resulting stream ensemble model.

On the other hand, using mask R-CNN to generate class segmented images as a pre-processing step yields salient information for training VGG19 networks. Combining these pre-processed regional images and raw images as a 'multi-modal' input for CNN streams enriches spatial feature representations and better represents subtle variations between different action classes. Furthermore, incorporating multiple unique streams into an ensemble model enhances the overall performance by leveraging differences between the underlying learned representations present within different streams.

Table 9 Benchmark Functions

Name	Range
Ackley	[−15, 30]
Dixon-Price	[−10, 10]
Griewank	[−600, 600]
Rastrigin	[−5.12, 5.12]
Rothyp	[−65, 65]
Rosenbrock	[−5, 10]
Sphere	[5.12, 5.12]
Sumpow	[−1, 1]
Zakharov	[−5, 10]
Sumsqu	[−5.12, 5.12]
Powell	[−4, 5]

Table 10 Experimental settings of the additional baseline methods

Name	Parameter settings
MPSO [78]	Time-varying acceleration coefficients and an adaptive inertia weight factor.
ELPSO [79]	$c_1 = c_2 = 2$, Standard deviation of Gaussian mutation=1, scale parameter of Cauchy mutation=2, scale factor of DE-based mutation=1.2, and an adaptive inertia weight factor.
DNLPSO [80]	$c_1 = c_2 = 1.49445$, Refreshing gap=3, regrouping period=5, and an adaptive inertia weight factor.
GPSO [81]	Maximum velocity=0.6, inertia weight=0.9, acceleration constants $c_1 = 2.6, c_2 = 1.5$, Crossover probability = 0.7, mutation probability = 0.3.
DA [82]	Alignment factor=0.1, separation factor=0.1, enemy factor=1, cohesion factor=0.7, food factor=1, and an adaptive inertia weight factor.
ALO [83]	Using adaptive parameter settings.

5 Evaluation using benchmark test functions

To further examine the performance of EnvPSO, we present another evaluation using eleven benchmark functions [74–77], as shown in Table 9. Each benchmark function produces a unique shape that presents a challenging task to attain the global minima. In particular, we use seven unimodal functions of Sum Squares (Sumsqu), Zakharov, Sum of Different Powers (Sumpow), Sphere, Rosenbrock, Rotated Hyper-Elipsoid (Rothyp), and Dixon-Price, as well as four multi-modal functions of Powell, Rastrigin, Griewank, and Ackley.

From Table 3 and Table 4, the superior results of the proposed EnvPSO model in tackling HAR tasks indicate the benefits of adding a Gaussian Fitness Surface and nonlinear adaptive coefficients. To re-confirm the observation, we compare this version of EnvPSO with a number of classical search methods and PSO variants using the aforementioned benchmark functions. In addition to original PSO, the following methods are used for comparison, i.e. a modified PSO (MPSO) [78], Enhanced Leader PSO (ELPSO) [79], Dynamic Neighbourhood Learning PSO (DNLPSO) [80], Genetic PSO (GPSO) [81], Dragonfly Algorithm (DA) [82] and Ant Lion Optimisation (ALO) [83]. The settings of these methods are extracted from their original publications shown in Table 10.

Each search method terminates according to the total number of function evaluations, as defined by $Eval_{max} = population \times iter_{max}$ with $population = 50$ and $iter_{max} = 500$, while $dimension = 30$ is adopted in the experiment. To reduce the effect of random errors and other biases, we repeat each experimental run 30 times.

Table 11 illustrates the mean, minimum, maximum, and standard deviation results over a set of 30 runs for all the test functions. As shown in Table 11, EnvPSO outperforms all the methods and achieves the best global minima in all the benchmark functions. The Wilcoxon rank sum test is conducted to evaluate the performance outcome statistically. As shown in Table 12, all the p-values except for two are lower than 0.05, ascertaining the statistically better performance of EnvPSO as compared with those of compared methods. The exceptions are for both Ackley and Rosenbrock landscapes, where the results of EnvPSO are statistically similar to those of DNLPSO and PSO, respectively.

6 Conclusion

In this research, we have proposed a multi-stream CNN ensemble model for undertaking human action recognition. A new PSO variant, denoted as EnvPSO, has been designed to perform automatic optimal hyper-parameter selection. It incorporates a Gaussian fitness surface estimation method and exponential adaptive coefficients to search for global optimality. Specifically, the time-varying exponential coefficients optimally calibrate the contribution of both social and cognitive components during each iteration, while gradient information yielded by the Gaussian fitness estimation surface is used to guide the search agents towards promising search regions. A new layer strip-back optimisation parameter is also proposed for determining the number of re-trainable layers of a stream CNN model at the fine-tuning stage.

A multi-stream ensemble model integrating three optimised CNN streams using EnvPSO is subsequently constructed for action classification. The ensemble diversity is not only enhanced by diverse learned representations of differing CNN networks with optimised distinctive transfer learning configurations, but also enriched by various input channels using raw images and mask R-CNN segmented salient features. The empirical results indicate that EnvPSO yields better efficiency in hyper-parameter selection for optimising each CNN stream in the ensemble model. Evaluated with two still image human action data sets, i.e.

Table 11 Evaluation results for the benchmark functions with dimension=30

		EnvPSO	PSO	DA	ALO	MPSO	DNLPSO	ELPSO	GPSO
Ackley	mean	**2.10E+00**	6.18E+00	7.55E+00	1.90E+01	1.72E+01	2.58E+00	1.49E+01	1.72E+01
	min	1.16E+00	3.04E+00	4.27E+00	1.90E+01	1.51E+01	**8.83E-03**	1.11E+01	1.58E+01
	max	**3.09E+00**	9.70E+00	1.19E+01	1.90E+01	1.87E+01	1.14E+01	1.60E+01	1.81E+01
	std	4.94E-01	1.85E+00	1.79E+00	**9.12E-03**	8.10E-01	2.38E+00	9.82E-01	5.57E-01
Dixon-Price	mean	**8.92E-01**	9.80E+00	1.31E+03	1.75E+06	7.92E+05	3.78E+02	1.43E+05	5.02E+05
	min	**6.67E-01**	6.78E-01	1.98E+01	1.07E+06	3.59E+04	6.68E-01	3.57E+04	1.56E+05
	max	**4.35E+00**	9.58E+01	1.14E+04	2.37E+06	1.61E+06	8.65E+03	2.44E+05	7.79E+05
	std	**7.74E-01**	2.65E+01	2.25E+03	2.77E+05	4.61E+05	1.58E+03	5.61E+04	1.46E+05
Griewank	mean	**2.16E-02**	3.59E-01	8.60E+00	5.85E+02	2.90E+02	4.05E+00	1.38E+02	2.87E+02
	min	1.31E-05	2.35E-02	1.90E+00	4.01E+02	1.52E+02	**1.28E-07**	6.44E+01	2.11E+02
	max	**1.00E-01**	1.43E+00	2.44E+01	6.89E+02	4.49E+02	7.61E+01	1.96E+02	3.73E+02
	std	**2.94E-02**	4.42E-01	5.79E+00	6.83E+01	7.19E+01	1.40E+01	2.71E+01	4.25E+01
Rastrigin	mean	**4.20E+01**	6.52E+01	1.18E+02	4.29E+02	3.37E+02	9.78E+01	2.71E+02	3.48E+02
	min	**1.89E+01**	3.09E+01	2.90E+01	3.81E+02	2.50E+02	2.99E+01	2.27E+02	3.10E+02
	max	**8.56E+01**	1.01E+02	2.51E+02	4.78E+02	4.26E+02	1.96E+02	3.20E+02	3.86E+02
	std	**1.46E+01**	1.71E+01	4.68E+01	2.16E+01	4.64E+01	4.49E+01	2.08E+01	1.86E+01
Rothyp	mean	**6.27E-05**	3.05E+00	5.74E+03	3.84E+05	1.89E+05	2.59E+03	9.60E+04	1.75E+05
	min	1.09E-05	8.67E-03	4.16E+02	3.00E+05	8.30E+04	**3.92E-07**	7.81E+04	1.32E+05
	max	**1.77E-04**	8.50E+01	2.08E+04	4.87E+05	3.76E+05	4.37E+04	1.21E+05	2.11E+05
	std	**3.73E-05**	1.55E+01	4.86E+03	4.41E+04	7.30E+04	8.74E+03	1.05E+04	1.98E+04
Rosenbrock	mean	**4.71E+01**	9.75E+01	3.52E+03	1.49E+06	3.37E+05	1.34E+02	1.07E+05	3.45E+05
	min	**6.25E-01**	1.59E+01	1.84E+02	5.76E+05	1.37E+05	2.71E+01	1.98E+04	1.26E+05
	max	**9.23E+01**	1.07E+03	1.87E+04	2.07E+06	6.38E+05	8.44E+02	2.26E+05	4.94E+05
	std	**3.32E+01**	1.88E+02	4.73E+03	3.74E+05	1.33E+05	1.51E+02	4.76E+04	9.16E+04
Sphere	mean	**1.28E-05**	2.66E-02	1.72E+00	1.70E+02	7.87E+01	1.36E+00	4.04E+01	8.97E+01
	min	2.18E-06	5.64E-03	1.97E-01	1.31E+02	2.37E+01	**4.08E-07**	2.11E+01	5.67E+01
	max	**4.84E-05**	8.12E-02	3.78E+00	1.96E+02	1.69E+02	1.92E+01	5.77E+01	1.27E+02
	std	**1.17E-05**	1.80E-02	1.16E+00	1.88E+01	3.36E+01	4.66E+00	8.27E+00	1.71E+01
Sumpow	mean	**3.17E-12**	1.05E-05	2.80E-05	6.65E-01	5.92E-01	1.18E-07	1.34E-02	1.38E-01
	min	3.71E-18	5.66E-07	**1.05E-51**	2.47E-01	3.80E-04	1.71E-21	3.22E-03	1.70E-02
	max	**4.44E-11**	3.49E-05	2.23E-04	1.10E+00	2.00E+00	2.68E-06	3.98E-02	5.32E-01
	std	**8.49E-12**	8.53E-06	5.44E-05	2.08E-01	5.79E-01	4.92E-07	9.32E-03	1.14E-01
Zakharov	mean	**5.30E+01**	1.35E+02	1.67E+02	6.98E+02	3.80E+02	1.16E+02	3.50E+02	4.44E+02
	min	**3.09E+01**	8.46E+01	5.77E+01	5.82E+02	2.61E+02	5.97E+01	2.94E+02	3.79E+02
	max	**7.36E+01**	1.98E+02	2.76E+02	7.50E+02	4.85E+02	2.71E+02	3.85E+02	4.79E+02
	std	**1.16E+01**	3.14E+01	5.62E+01	4.74E+01	4.42E+01	5.08E+01	2.02E+01	2.36E+01
Sumsqu	mean	**3.15E-05**	7.76E-02	3.69E+01	2.40E+03	1.19E+03	7.95E+00	5.55E+02	1.19E+03
	min	6.09E-07	6.33E-03	2.18E+00	1.49E+03	3.43E+02	**3.10E-08**	2.72E+02	7.62E+02
	max	**1.92E-04**	3.81E-01	1.10E+02	2.82E+03	2.20E+03	1.16E+02	8.34E+02	1.54E+03
	std	**4.24E-05**	9.17E-02	2.57E+01	2.94E+02	5.27E+02	2.49E+01	1.15E+02	2.09E+02
Powell	mean	**1.05E-03**	1.13E-01	1.12E+02	1.19E+04	6.10E+03	1.60E+01	1.46E+03	3.72E+03
	min	**3.08E-04**	7.34E-03	4.55E+00	6.51E+03	6.50E+02	2.27E-03	7.97E+02	2.26E+03
	max	**2.32E-03**	4.83E-01	4.35E+02	1.53E+04	1.52E+04	3.02E+02	2.43E+03	5.89E+03
	std	**5.29E-04**	1.24E-01	1.14E+02	2.91E+03	4.32E+03	5.47E+01	4.34E+02	9.82E+02

Bold indicates the best results

BU-101 and Willow7, the proposed multi-stream CNN ensemble model with EnvPSO hyper-parameter optimisation outperforms the counterparts with default and optimised settings identified by PSO and other state-of-the-

Table 12 The Wilcoxon rank sum test results for the benchmark functions over 30 runs

	PSO	DA	ALO	MPSO	DNLPSO	ELPSO	GPSO
Ackley	3.34E-11	3.02E-11	1.72E-12	3.02E-11	7.73E-01	3.02E-11	3.02E-11
Dixon-Price	2.20E-07	3.02E-11	3.02E-11	3.02E-11	4.20E-10	3.02E-11	3.02E-11
Griewank	1.43E-08	3.02E-11	3.02E-11	3.02E-11	2.39E-08	3.02E-11	3.02E-11
Rastrigin	2.00E-06	3.02E-11	3.02E-11	3.02E-11	2.83E-08	3.02E-11	2.02E-08
Rothyp	3.02E-11	3.02E-11	3.02E-11	3.02E-11	2.03E-07	3.02E-11	3.02E-11
Rosenbrock	5.01E-02	3.02E-11	3.02E-11	3.02E-11	1.75E-05	3.02E-11	3.02E-11
Sphere	3.02E-11	3.02E-11	3.02E-11	3.02E-11	1.61E-06	3.02E-11	3.02E-11
Sumpow	3.02E-11	3.02E-11	3.02E-11	3.02E-11	9.76E-10	3.02E-11	1.01E-08
Zakharov	3.02E-11	3.02E-11	2.92E-11	3.02E-11	9.76E-10	3.02E-11	1.33E-10
Sumsqu	3.02E-11	3.02E-11	3.02E-11	3.02E-11	2.19E-08	3.02E-11	3.02E-11
Powell	3.02E-11	3.02E-11	2.92E-11	3.02E-11	3.34E-11	3.02E-11	3.02E-11

art methods. Therefore, it is evident that the proposed search strategies, which include Gaussian fitness surface estimation and exponential coefficients, account for better efficiency of our devised ensemble model with better generalised internal representations of diverse action classes. Our model also outperforms a number of classical and advanced search methods statistically significantly for solving diverse unimodal and multi-modal benchmark functions, as confirmed by statistical test results.

For future work, we will focus on optimising elements (such as the CNN blocks) of the stream architectures and further fine-tuning their configurations. We will also investigate an entire new deep architecture generated using EnvPSO for each stream in the ensemble model, in order to increase feature extraction diversity. Other surface estimation techniques (such as *n*-dimensional interpolation methods) will be studied to further improve fitness surface estimation with respect to environmental components. We also aim to evaluate the proposed model for hyper-parameter fine-tuning in complex and dynamic computer vision tasks such as video action recognition, object detection, and visual question generation.

Acknowledgements This work was supported in part by European Regional Development Fund (ERDF) and in part by RPPTV Ltd. for jointly funding a Ph.D. studentship via the Intensive Industrial Innovation Programme North East (IIIPNE, Grant No. 25R17P01847).

Author contributions Sam Slade contributed to the conceptualisation, data curation, formal analysis, investigation, methodology, resources, software, validation, visualisation, roles/writing—original draft, writing—review and editing. Li Zhang: was involved in the conceptualisation, formal analysis, investigation, methodology, resources, supervision, validation, roles/writing—original draft, writing—review and editing. Yonghong Yu contributed to the supervision, validation, writing—review and editing. Chee Peng Lim contributed to the supervision, validation, writing—review and editing.

Availability of data and materials This research employs publicly available human action data sets for experimental studies.

Code availability The authors will publish the code for the proposed work in a dedicated website after the acceptance of the paper.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Ethics approval The proposed work has gained organisational ethical approval.

Consent to participate The consent to participate has been obtained from all the co-authors for the proposed studies.

Consent for publication The consent for publication has been obtained from all the co-authors for the proposed studies.

References

1. Guo G, Lai A (2014) A survey on still image based human action recognition. Pattern Recognit 47(10):3343–3361
2. Zheng Y, Zhang Y-J, Li X, Liu B-D (2012) Action recognition in still images using a combination of human pose and context information. In: 2012 19th IEEE International Conference on Image Processing, pp. 785–788 . IEEE
3. Thurau C, Hlavác, V (2008) Pose primitive based human action recognition in videos or still images. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE
4. Desai C, Ramanan D, Fowlkes C (2010) Discriminative models for static human-object interactions. In: 2010 IEEE Computer

Society Conference on Computer Vision and Pattern Recognition-Workshops, pp. 9–16. IEEE
5. Shapovalova N, Gong W, Pedersoli M, Roca FX, Gonzalez J (2011) On importance of interactions and context in human action recognition. In: Iberian Conference on Pattern Recognition and Image Analysis, pp. 58–66. Springer
6. Li, L.-J., Fei-Fei L (2007) What, where and who? classifying events by scene and object recognition. In: 2007 IEEE 11th International Conference on Computer Vision, pp. 1–8. IEEE
7. Gupta A, Kembhavi A, Davis LS (2009) Observing human-object interactions: using spatial and functional compatibility for recognition. IEEE Trans Pattern Anal Mach Intell 31(10):1775–1789
8. Maji S, Bourdev L, Malik J (2011) Action recognition from a distributed representation of pose and appearance. In: CVPR 2011, pp. 3177–3184. IEEE
9. Desai C, Ramanan D (2012) Detecting actions, poses, and objects with relational phraselets. In: European Conference on Computer Vision, pp. 158–172. Springer
10. Delaitre V, Sivic J, Laptev I (2011) Learning person-object interactions for action recognition in still images. Adv Neural Inform Process Syst 24:1503–1511
11. Sener F, Bas C, Ikizler-Cinbis N (2012) On recognizing actions in still images via multiple features. In: European Conference on Computer Vision, pp. 263–272. Springer
12. Yao B, Fei-Fei L (2012) Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses. IEEE Trans Pattern Anal Mach Intell 34(9):1691–1703
13. Yao B, Khosla A, Fei-Fei L (2011) Classifying actions and measuring action similarity by modeling the mutual context of objects and human poses. a) A 1(D2), p. 3
14. Li P, Ma J (2011) What is happening in a still picture? In: The First Asian Conference on Pattern Recognition, pp. 32–36. IEEE
15. Le DT, Bernardi R, Uijlings J (2013) Exploiting language models to recognize unseen actions. In: Proceedings of the 3rd ACM Conference on International Conference on Multimedia Retrieval, pp. 231–238
16. Yao B, Jiang X, Khosla A, Lin AL, Guibas L, Fei-Fei L (2011) Human action recognition by learning bases of action attributes and parts. In: 2011 International Conference on Computer Vision, pp. 1331–1338. IEEE
17. Delaitre V, Laptev I, Sivic J (2010) Recognizing human actions in still images: a study of bag-of-features and part-based representations. In: BMVC 2010-21st British Machine Vision Conference, pp. 1–11
18. Qazi HA, Jahangir U, Yousuf BM, Noor A (2017) Human action recognition using SIFT and HOG method. In: 2017 International Conference on Information and Communication Technologies (ICICT), pp. 6–10. IEEE
19. Sharma G, Jurie F, Schmid C (2013) Expanded parts model for human attribute and action recognition in still images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 652–659
20. Dhulavvagol PM, Kundur NC (2017) Human action detection and recognition using SIFT and SVM. In: International Conference on Cognitive Computing and Information Processing, pp. 475–491. Springer
21. Li B, Xiao R, Li Z, Cai R, Lu B-L, Zhang L (2011) Rank-SIFT: Learning to rank repeatable local interest points. In: CVPR 2011, pp. 1737–1744. IEEE
22. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
23. Kreowsky P, Stabernack B (2021) A Full-featured FPGA-Based pipelined architecture for SIFT extraction. IEEE Access 9:128564–128573
24. Aslan MF, Durdu A, Sabanci K, Mutluer MA (2020) CNN and HOG based comparison study for complete occlusion handling in human tracking. Measurement 158:107704
25. Wang X, Han TX, Yan S (2009) An HOG-LBP human detector with partial occlusion handling. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 32–39. IEEE
26. Yang H, Shao L, Zheng F, Wang L, Song Z (2011) Recent advances and trends in visual tracking: a review. Neurocomputing 74(18):3823–3831
27. Oliva A, Torralba A (2006) Building the gist of a scene: the role of global image features in recognition. Prog Brain Res 155:23–36
28. Siagian C, Itti L (2007) Rapid biologically-inspired scene classification using features shared with visual attention. IEEE Trans Pattern Anal Mach Intell 29(2):300–312
29. Xie B, Qin J, Xiang X, Li H, Pan L (2018) An image retrieval algorithm based on gist and sift features. Int J Netw Secur 20(4):609–616
30. Zhang L, Lim CP, Yu Y (2021) Intelligent human action recognition using an ensemble model of evolving deep networks with swarm-based optimization. Knowledge-Based Sys 220:106918
31. Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos. arXiv preprint arXiv:1406.2199
32. Al-Faris M, Chiverton J, Ndzi D, Ahmed AI (2020) A review on computer vision-based methods for human action recognition. J Imaging 6(6):46
33. Degardin B, Proença H (2021) Human behavior analysis: a survey on action recognition. Appl Sci 11(18):8324
34. Zhang H-B, Zhang Y-X, Zhong B, Lei Q, Yang L, Du J-X, Chen D-S (2019) A comprehensive survey of vision-based human action recognition methods. Sensors 19(5):1005
35. Yao G, Lei T, Zhong J (2019) A review of convolutional-neural-network-based action recognition. Pattern Recognit Lett 118:14–22
36. Kong Y, Fu Y (2018) Human action recognition and prediction: a survey. arXiv preprint arXiv:1806.11230
37. Zhang L, Mistry K, Neoh SC, Lim CP (2016) Intelligent facial emotion recognition using moth-firefly optimization. Knowl-Based Syst 111:248–267
38. Lawrence T, Zhang L, Rogage K, Lim CP (2021) Evolving deep architecture generation with residual connections for image classification using particle swarm optimization. Sensors 21(23):7936
39. Tan TY, Zhang L, Lim CP (2020) Adaptive melanoma diagnosis using evolving clustering, ensemble and deep neural networks. Knowl-Based Syst 187:104807
40. Tan TY, Zhang L, Lim CP, Fielding B, Yu Y, Anderson E (2019) Evolving ensemble models for image segmentation using enhanced particle swarm optimization. IEEE Access 7:34004–34019
41. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
42. Wang Y, Zhang H, Zhang G (2019) cPSO-CNN: An efficient PSO-based algorithm for fine-tuning hyper-parameters of convolutional neural networks. Swarm and Evol Comput 49:114–123
43. da Silva GLF, Valente TLA, Silva AC, de Paiva AC, Gattass M (2018) Convolutional neural network-based PSO for lung nodule false positive reduction on CT images. Comput Methods Prog Biomed 162:109–118
44. Soon FC, Khaw HY, Chuah JH, Kanesan J (2018) Hyper-parameters optimisation of deep CNN architecture for vehicle logo recognition. IET Intell Trans Syst 12(8):939–946

45. Tan TY, Zhang L, Lim CP (2019) Intelligent skin cancer diagnosis using improved particle swarm optimization and deep learning models. Appl Soft Comput 84:105725
46. Fielding B, Zhang L (2018) Evolving image classification architectures with enhanced particle swarm optimisation. IEEE Access 6:68560–68575
47. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82
48. Mistry K, Zhang L, Neoh SC, Lim CP, Fielding B (2016) A micro-GA embedded PSO feature selection approach to intelligent facial emotion recognition. IEEE Trans Cybernet 47(6):1496–1509
49. Fielding B, Zhang L (2020) Evolving deep denseBlock architecture ensembles for image classification. Electronics 9(11):1880
50. Nobile MS, Cazzaniga P, Besozzi D, Colombo R, Mauri G, Pasi G (2018) Fuzzy Self-tuning PSO: a settings-free algorithm for global optimization. Swarm Evol Comput 39:70–85
51. Singh P, Chaudhury S, Panigrahi BK (2021) Hybrid MPSO-CNN: Multi-level particle swarm optimized hyperparameters of convolutional neural network. Swarm and Evol Comput 63:100863
52. Bai B, Zhang J, Wu X, wei Zhu G, Li X (2021) Reliability prediction-based improved dynamic weight particle swarm optimization and back propagation neural network in engineering systems. Exp Syst Appl 177:114952
53. Lan R, Zhang L, Tang Z, Liu Z, Luo X (2019) A hierarchical sorting swarm optimizer for large-scale optimization. IEEE Access 7:40625–40635
54. Han H, Lu W, Zhang L, Qiao J (2017) Adaptive gradient multiobjective particle swarm optimization. IEEE Trans Cybernet 48(11):3067–3079
55. Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: empirical results. Evol Comput 8(2):173–195
56. Deb K, Thiele L, Laumanns M, Zitzler E (2005) Scalable test problems for evolutionary multiobjective optimization. In: Evolutionary Multiobjective Optimization, pp. 105–145. Springer, London
57. Cai J, Wei H, Yang H, Zhao X (2020) A novel clustering algorithm based on DPC and PSO. IEEE Access 8:88200–88214
58. Sharma G, Jurie F, Schmid C (2016) Expanded parts model for semantic description of humans in still images. IEEE Trans Pattern Anal Mach Intell 39(1):87–101
59. Zhang Y, Cheng L, Wu J, Cai J, Do MN, Lu J (2016) Action recognition in still images with minimum annotation efforts. IEEE Trans Image Process 25(11):5479–5490
60. Wang J, Wang G (2016) Hierarchical spatial sum-product networks for action recognition in still images. IEEE Trans Circuits Syst Video Technol 28(1):90–100
61. Li J, Wong Y, Zhao Q, Kankanhalli MS (2017) Attention transfer from web images for video recognition. In: Proceedings of the 25th ACM International Conference on Multimedia, pp. 1–9
62. Safaei M (2020) Action recognition in still images: confluence of multilinear methods and deep learning methods and deep learning. PhD thesis, University of Central Florida
63. Yu X, Zhang Z, Wu L, Pang W, Chen H, Yu Z, Li B (2020) Deep ensemble learning for human action recognition in still images. Complexity **2020**, 1–23. Article ID 9428612
64. Liu L, Tan RT, You S (2018) Loss guided activation for action recognition in still images. In: Asian Conference on Computer Vision, pp. 152–167. Springer
65. Yan S, Smith JS, Lu W, Zhang B (2017) Multibranch attention networks for action recognition in still images. IEEE Trans Cognit Develop Syst 10(4):1116–1125
66. Masters D, Luschi C (2018) Revisiting small batch training for deep neural networks. arXiv preprint arXiv:1804.07612
67. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969
68. Ma S, Bargal SA, Zhang J, Sigal L, Sclaroff S (2017) Do less and achieve more: training CNNs for action recognition utilizing action images from the web. Pattern Recognit 68:334–345
69. Safaei M, Foroosh H (2019) Still image action recognition by predicting spatial-temporal pixel evolution. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 111–120. IEEE
70. Sharma G, Jurie F, Schmid C (2012) Discriminative spatial saliency for image classification. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3506–3513. IEEE
71. Lazebnik S, Schmid C, Ponce J (2006) Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), vol. 2, pp. 2169–2178. IEEE
72. Safaei M, Balouchian P, Foroosh H (2002) UCF-STAR: A large scale still image dataset for understanding human actions. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 2677–2684
73. Alraimi ASA (2019) Development of new models for vision-based human activity recognition. PhD thesis, Universitat Rovira i Virgili
74. Pandit D, Zhang L, Chattopadhyay S, Lim CP, Liu C (2018) A scattering and repulsive swarm intelligence algorithm for solving global optimization problems. Knowl-Based Syst 156:12–42
75. Zhang L, Lim CP, Yu Y, Jiang M (2021) Sound classification using evolving ensemble models and Particle Swarm Optimization. Appl Soft Comput, p. 108322
76. Zhang L, Lim CP (2020) Intelligent optic disc segmentation using improved particle swarm optimization and evolving ensemble models. Appl Soft Comput 92:106328
77. Zhang L, Srisukkham W, Neoh SC, Lim CP, Pandit D (2018) Classifier ensemble reduction using a modified firefly algorithm: An empirical evaluation. Exp Syst Appl 93:395–422
78. Nayak DR, Dash R, Majhi B (2018) Discrete ripplet-II transform and modified PSO based improved evolutionary extreme learning machine for pathological brain detection. Neurocomputing 282:232–247
79. Jordehi AR (2015) Enhanced leader PSO (ELPSO): a new PSO variant for solving global optimisation problems. Appl Soft Comput 26:401–417
80. Nasir M, Das S, Maity D, Sengupta S, Halder U, Suganthan PN (2012) A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization. Inform Sci 209:16–36
81. Chen Q, Chen Y, Jiang W (2016) Genetic particle swarm optimization-based feature selection for very-high-resolution remotely sensed imagery object change detection. Sensors 16(8):1204
82. Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput Appl 27(4):1053–1073
83. Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 83:80–98