Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.035214

ARTICLE

Improved Shark Smell Optimization Algorithm for Human Action Recognition

Inzamam Mashood Nasir[1,*], Mudassar Raza[1], Jamal Hussain Shah[1], Muhammad Attique Khan[2], Yun-Cheol Nam[3] and Yunyoung Nam[4,*]

[1]Department of Computer Science, COMSATS University Islamabad, Wah Campus, Wah Cantt, 47040, Pakistan

[2]Department of Computer Science, HITEC University, Taxila, Pakistan

[3]Department of Architecture, Joongbu University, Goyang, 10279, South Korea

[4]Department of ICT Convergence, Soonchunhyang University, Asan, 31538, Korea

*Corresponding Authors: Inzamam Mashood Nasir. Email: inzamam.mashood@hitecuni.edu.pk;
Yunyoung Nam. Email: ynam@sch.ac.kr

Received: 11 August 2022 Accepted: 15 November 2022 Published: 08 October 2023

ABSTRACT

Human Action Recognition (HAR) in uncontrolled environments targets to recognition of different actions from a video. An effective HAR model can be employed for an application like human-computer interaction, health care, person tracking, and video surveillance. Machine Learning (ML) approaches, specifically, Convolutional Neural Network (CNN) models had been widely used and achieved impressive results through feature fusion. The accuracy and effectiveness of these models continue to be the biggest challenge in this field. In this article, a novel feature optimization algorithm, called improved Shark Smell Optimization (iSSO) is proposed to reduce the redundancy of extracted features. This proposed technique is inspired by the behavior of white sharks, and how they find the best prey in the whole search space. The proposed iSSO algorithm divides the Feature Vector (FV) into subparts, where a search is conducted to find optimal local features from each subpart of FV. Once local optimal features are selected, a global search is conducted to further optimize these features. The proposed iSSO algorithm is employed on nine (9) selected CNN models. These CNN models are selected based on their top-1 and top-5 accuracy in ImageNet competition. To evaluate the model, two publicly available datasets UCF-Sports and Hollywood2 are selected.

KEYWORDS

Action recognition; improved shark smell optimization; convolutional neural networks; machine learning

1 Introduction

Human Action Recognition (HAR) includes the action recognition of a person through imaging data which has various applications. Recognition approaches can be divided into three categories: multi-model, overlapping categories, and video sequences [1]. This data used for recognition is the major difference between images and video categories. Data in form of images and videos are acquired through cameras in controlled and uncontrolled environments. With the advancement of technology in past decades, various smart devices have been developed which to collect images and video data for

HAR, health monitoring, and disease prevention [2]. Different research has been carried out on HAR through images or videos over the last three decades [3,4]. Human visual systems get visual information about an object such as its movement, shape, and its variations. This information is used to investigate the biophysical processes of HAR. Computer vision systems have achieved very good accuracy while catering to different challenges such as occlusion, background clutter, scale and rotation invariance, and environmental changes [5].

HAR depending upon the action complexity can be divided into primitive, single-person, interaction, and group action recognition [6]. The basic movement of a single human body part considers primitive action, a set of primitive actions of one person includes including single-person action, a collection of humans and objects involves in interaction while collective actions performed by a group of people are group actions. Computer vision-based HAR systems are divided into hand-crafted feature-based methods and deep learning-based methods. The combined framework of hand-crafted and deep features is also employed by many researchers [7].

The data plays an important role in efficient HAR systems. The HAR data is categorized into color channels, depth, and skeleton information. Texture information can be extracted from color channels, i.e., RGB which is close to the visual appearance, but illumination variations can affect the visual data [8]. Depth map information is invariant to the lighting changes which is helpful in foreground object extractions. 3D information can also be captured through a depth map, but noise factors should be considered while capturing the depth map. Skeletons information can be gathered through color channels and depth maps, but it can be exploited from environmental factors [9]. HAR systems use different levels of features such as whole data as the input of HAR used in [10]. Apart from features, motion is an important factor that can be incorporated into the feature computation step. It includes optical flow for capturing low-level feature information in multiple video frames. Some researchers included motion information in the classification step with Conditional Random Fields, Hidden Markov Models, Long-Short Term Memory (LSTM), Recurrent Neural Networks (RNN), and 3D Convolutional Neural Networks (CNN) [11–15]. These HAR systems have good recognition accuracy using the most appropriate feature set.

A CNN-based convolutional 3D (C3D) network was proposed in [16]. The major difference between the 3D CNN and the proposed one was that it utilized the whole video as an input instead of a few frames or segmented frames, which makes it robust for large databases. The architecture of the C3D network comprises several layer groups like convolutional layer = 8, maximum pooling layers = 5, fully connected layers = 2, and the last softmax loss layer. UCF 101 dataset was utilized to evaluate the best combination of the proposed network architecture. The best performance achieved by the proposed network was using a $3 \times 3 \times 3$ convolutional filter without updating the other parameter. The researcher came up with RNNs [17] to overcome the limitation action of CNN models of information derivation from long timelapse. RNN has proved robust while extracting time dimension features and has one drawback of gradient disappearance. The mentioned problem is addressed by presenting Long Short-Term Memory Network (LSTM) [18], which utilizes processors to gauge the information integrity and relevance. Normally, input gates, output gates, and forget gates are utilized in the processor. The information flow is controlled by gates in the processor and unnecessary information which requires large memory chunks is stored for long-term tasks.

A ConvNet architecture for the spatiotemporal fusion of video fragments has evaluated its performance on dataset UCF-101 by achieving an accuracy of 93.5% and HMDB-51 by achieving an accuracy of 69.2% [19]. An architecture is proposed to handle 3D signals effectively and efficiently and introduced Factorized Spatio-Temporal Convolutional Network (FSTCN). It was tested on two publicly available datasets UCF-101 and achieved 88.1% accuracy, while achieved 59.0% accuracy on HMDB-51 [20]. In another method, LSTM models are trained to utilize the differential gating scheme, which focuses on the varying gain due to the slow movements between the successive frames, change based on Derivate of States (DoS) and this combined called differential RNN (dRNN). The method is implemented on KTH and MSRAction3D datasets. The accuracy achieved on their datasets is 93.96% and 92.03%, respectively [21].

This article presents an improved form of the Shark Smell Algorithm (SSO), which reduces redundant features. The proposed algorithm utilizes both, SSO and White Shark Optimization (WSO) properties to solve the redundancy issues. The proposed iSSO divides the population into sub-spaces to find local and global optimal features. In the end, these extracted local features are used to optimize global features. Features are extracted using 9 pre-trained CNN models, which are selected based on their top-1 and top-5 accuracies in ImageNet competition. This model is tested on two publicly available datasets UCF-Sports (D1) and Hollywood2 (D2) and it has obtained better results than state-of-the-art (SOTA) methods.

2 Proposed Methodology

In an uncontrolled environment, various viewports, illuminations, and changing backgrounds, traditional hand-crafted features have been proved insufficient [22]. In the age of big data and the evolution of ML methods, Deep Learning (DL) has achieved remarkable results [23–25]. These results have motivated researchers around the globe to apply these DL methods to domains involving video data. The challenge of ImageNet classification drastically changed the dimensions of DL methods, when CNNs made a huge breakthrough. The main difference between CNN methods and local feature-based methods is that CNN iteratively and automatically extracts deep features through its interconnected layers.

2.1 Transfer Learning of Pre-Trained CNN Models

Artificial Intelligence (AI) and Machine Learning (ML) have a sub-domain, called Transfer Learning (TL), which transforms the learned knowledge of one problem (base problem) into another problem (target problem). TL improves the learning of a model through the data provided for the target problem. A model trained to classify Wikipedia text can be utilized to classify the texts of simple documents after TL. A model trained to classify cards can also classify birds. The nature of this problem is the same, which is to classify objects. TL provides scalability to a trained model, which enables it to recognize different types of objects. Since 2015, after the first CNN model, AlexNet [22] was proposed, a lot of CNN architectures were proposed. The base for all these models was a competition, where a dataset, ImageNet [26], having 1000 classes was presented. The efficiency of all proposed CNN models to date is still measured on how the proposed model performs on the ImageNet dataset. In this research, nine of the most used CNN models are selected, where, through TL, features of input images from selected datasets will be extracted. Table 1 lists all selected CNN models along with their depth, size, input size, number of parameters, and their top-1 and top-5 accuracies on ImageNet datasets.

Table 1: Different characteristics of selected pre-trained CNN models

Model	Depth	Size (MB)	Input size	Parameters (Millions)	Accuracy (%)	
					Top-1	Top-5
VGG19 (Vg) [23]	19	535	$224 \times 224 \times 3$	143.6	71.3	90.0
MobileNetV2 (Mo) [24]	53	13	$299 \times 299 \times 3$	3.5	71.3	90.1
Resnet50 (Re) [25]	50	96	$224 \times 224 \times 3$	25.6	74.9	92.1
EfficientNet-B0 (Ef) [27]	82	20	$224 \times 224 \times 3$	5.3	77.1	93.3
DarkNet53 (Da) [28]	53	155	$256 \times 256 \times 3$	41.6	77.2	93.8
DenseNet201 (De) [29]	201	77	$224 \times 224 \times 3$	20.0	77.3	93.6
Xception (Xe) [30]	71	85	$299 \times 299 \times 3$	22.9	79.0	94.5
InceptionResNetV2 (In) [31]	164	209	$299 \times 299 \times 3$	55.9	80.3	95.3
NASNetLarge (Na) [32]	-	332	$331 \times 331 \times 3$	88.9	82.5	96.0

The structure of all these selected pre-trained models is different because of the nature and arrangement of layers. The selected feature extraction layer and extracted features per image vary from model to model. For Vg, the fc7 layer is selected to extract 4096 features for a single image. 1280 and 4032 features are extracted from the global_average_pooling2d_1 and global_average_pooling2d_2 layers of Mo and Na models, respectively. avg_pool is selected as a feature extraction layer for Re, De, Xe, and In models, which extracted 2048, 1920, 2048, and 1536 features, respectively. avg1 is selected as the feature extraction layer for Da, and it extracted 1024 features against a single image. When the Ef model is used as a feature extractor, it extracts 1280 features from the GlobAvgPool layer. All these extracted features are forwarded to iSSO for optimization.

2.2 Improved Shark Smell Optimization (iSSO)

The meta-heuristic model used in this article is an improved form of Shark Smell Optimization (SSO) [33]. The SSO was proposed after inspiration was taken from the species of sharks. Sharks are considered as most hazardous and strongest predacious in the universe [34]. Sharks are creatures with a keen ability to smell and highly contrasted vision due to their sturdy eyesight and powerful muscles. They have more than 300 sharp, pointing, and triangular teeth in their gigantic jaws. Sharks usually strike with a large and abrupt bite of prey, which proves so sudden that the prey cannot avoid it. These sharks hunt the prey by using their extreme sense of smelling and hearing the traits of prey. The iSSO algorithm initially divides the whole search space into ş subparts. The algorithm then performs the local and global search to find the optimum prey in both, local and global search spaces of ş. Once an optimum prey is located, the search then continues to find all the optimal prey in the remaining subparts. The process mentioned below is for a single subpart. The whole process will be repeated for all ş. Another factor is the quantity of selected optimal features. For this, ş denotes the total selected features.

2.2.1 Prey Tracking

Sharks wander in the ocean freely just like any other organism of the sea and search for prey. In that search, sharks update their positions by the traits of prey. They apply all their tricks to locate, stalk and track down the prey. All senses of sharks along with their average distance range are illustrated in Fig. 1. All these illustrated features help them to exploit and search the whole space for hunting prey.

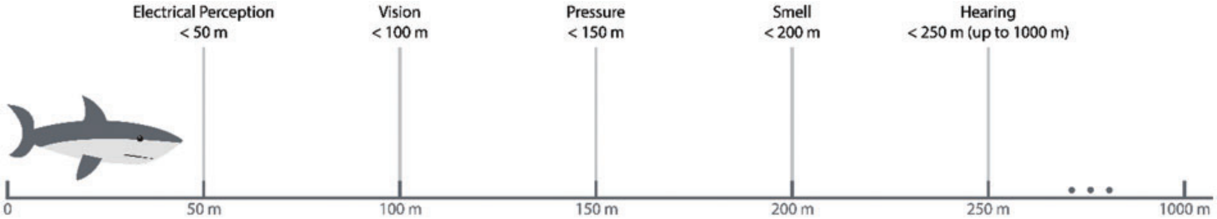


Figure 1: Senses of shark along with its average distance range

2.2.2 *Prey Searching (Exploration and Exploitation)*

The sharks have a very unfamiliar sense of hearing, that is, they can hear any wavelength from the full length of their body. Their whole body can detect any change in water pressure and reveal the nearby movements of the targeted prey. The attention of sharks is usually attained by moving prey, which leaves a disturbance in water pressure. Sharks even have body organs, which can detect the tiny electromagnetic fields, produced through the swimming of prey. Turbulence due to the prey's motion helps sharks to sense the frequency of waves and accurately predict the size and location of prey. The velocity of waves detected by sharks is described as:

$$\upsilon = \omega \times \omega_f \tag{1}$$

where υ denotes the velocity of wavy motion, ω denotes the wavelength that defines the distance between shark and prey and ω_f denotes the frequency of waves during the wavy motion. This frequency is determined by the total number of cycles, completed by the shark in a second. The sharks utilize their extraordinary sense to exploit the whole space and to detect prey. Once, a prey is in the nearby area, the senses of the shark grow exponentially, and it travels towards the pined point position of the prey. The following equation is assumed to be used to update the position of a shark with constant acceleration:

$$\rho = \rho_i + (\upsilon_i \times \Delta T) + \left(0.5 \times A_{cc} \times \Delta T^2\right) \tag{2}$$

here, a new position of the shark is denoted by ρ, the primitive position is denoted by ρ_i and the initial velocity is denoted by υ_i. The interval taken to travel between current and initial positions is represented by ΔT and A_{cc} denotes the constant acceleration factor. Many preys disburse their scent when they leave their position. When a shark reaches that position, it finds no prey and thus starts to search for the prey randomly and explore the nearby areas by using its sense of smell, hearing, and sight. The first step of this algorithm is to generate a search space of all possible solutions. Search space of m sharks in n dimensions, with a position of all sharks, is presented as:

$$\mathrm{P} = \begin{bmatrix} \mathrm{P}_1^1 & \dots & \mathrm{P}_n^1 \\ \vdots & \vdots & \vdots \\ \mathrm{P}_1^m & \dots & \mathrm{P}_n^m \end{bmatrix} \tag{3}$$

here, P is a 2D matrix, containing the positions of all sharks in search space, n denotes the total number of decision variables and P_n^i represents x^{th} shark in n^{th} dimension. This population is generated by randomly initialized upper and lower bounds as:

$$P^x_y = LB_y + \left(rand \times \left(LB_y - UB_y\right)\right) \tag{4}$$

here, P^x_y denotes the initial matrix of x^{th} shark in the y^{th} dimension, while LB_y and UB_y denotes the lower and upper bounds of search space in the y^{th} dimension. *rand* is a randomly generated number between the intervals [0, 1]. A fitness function assesses each possible solution for every new position of a shark. If a better position is found, then the current position is renovated with the better one, else the shark remains at its current position.

Now is the time for the shark to move toward prey. When a shark detects the waves of moving prey, it locks its target and starts moving towards that prey, which is defined as:

$$\rho\upsilon^a_{s+1} = \Lambda\left[\upsilon^a_s + \varsigma_1\left(P_{gb_s} - P^a_s\right) \times rand_1 + \varsigma_2\left(P^{\upsilon^a s}_b - P^a_s\right) \times rand_2\right] \tag{5}$$

In the above equation, $a = 1, \ldots x$ denotes the selected shark in search space of size x, the velocity of a^{th} shark in $(s+1)$ steps denoted by υ^a_{s+1}, υ^a_s is the velocity of a^{th} shark in s^{th} step, P^a_s is the current position of a^{th} shark in s^{th} step, P_{gb_s} is the current global best position of the a^{th} shark in s^{th} iteration, $P^{\upsilon^a s}_b$ is the last known best position of the shark, while υ^a represents an index vector of the best-positioned shark at i^{th} index. $rand_1$ and $rand_2$ are two randomly generated numbers, while ς_1 and ς_2 denotes the force element on sharks, which controls the impact of P_{gb_s} and $P^{\upsilon^a s}_b$, respectively. Λ denotes the factor for constriction. υ and Λ are calculated as per the following equations, respectively:

$$\upsilon = \lfloor a \times rand\,(1, a) \rfloor + 1 \tag{6}$$

$$\Lambda = \frac{2}{\left|2 - \mathbb{C} - \sqrt{\mathbb{C}^2 - 4\mathbb{C}}\right|} \tag{7}$$

here, $\mathbb{C}$ represents the coefficient of acceleration. The value of $\mathbb{C}$ for this work is equal to 2.145 after extensive experiments. ς_1 and ς_2 are calculated as:

$$\varsigma_1 = \varsigma_{max} + (\varsigma_{max} - \varsigma_{min}) \times e^{-\left(\frac{4s}{S}\right)^2} \tag{8}$$

$$\varsigma_2 = \varsigma_{min} + (\varsigma_{max} - \varsigma_{min}) \times e^{-\left(\frac{4s}{S}\right)^2} \tag{9}$$

here, maximum, and current iterations are denoted by S and s. Active motion of sharks can be achieved by using subordinate and initial velocities denoted by ς_{max} and ς_{min}. For this work, these velocities for ς_{max} and ς_{min} is set at 0.14 and 1.35, respectively.

The sharks spend most of their time searching for optimal prey and to achieve it, they constantly change their positions. Their position changes when either they smell the scent of prey or they feel the movement in waves, caused by prey. Sometimes, a potential prey leaves its position and leaves some scent, either they feel a shark coming towards them or in search of food. In this case, the shark starts to stray randomly in search of other prey. The position of the shark, in that case, is updated as per the following equation:

$$P^a_{s+1} = \begin{cases} P^a_s \cdot \sim \oplus P_0 + \mathrm{UB} \cdot g + LB \cdot h & ; rand < mF \\ P^a_s + \dfrac{\upsilon^a_s}{\omega_f} & ; rand \geq mF \end{cases} \tag{10}$$

here, P^a_{s+1} denotes the newly taken position of a^{th} shark in the $(s+1)^{th}$ iteration, $\sim$ represents the complement operator, g and h represent two 1D binary vectors, P_0 is a logical vector, ω_f is the frequency, while movement force denoted by mF is a factor to denote the senses of shark, i.e., smell and hear, which increases with the increasing number of iterations. $\oplus$ denotes a bitwise xor operation. g, h, P_0, ω_f and mF is defined as:

$$g = scd\left(\mathrm{P}_s^a - UB\right) > 0 \tag{11}$$

$$h = scd\left(\mathrm{P}_s^a - LB\right) < 0 \tag{12}$$

$$\mathrm{P}_0 = g \oplus h \tag{13}$$

$$\omega_f = \omega_{f_{min}} + \frac{\omega_{f_{max}} - \omega_{f_{min}}}{\omega_{f_{max}} + \omega_{f_{min}}} \tag{14}$$

$$mF = \frac{1}{\left(p + e^{\frac{(0.5\times S)-s}{q}}\right)} \tag{15}$$

here, *scd* is a factor, which changes the direction of the moving shark, $\omega_{f_{max}}$ and $\omega_{f_{mix}}$ denote the maximum and minimum frequencies during its motion, p and q denote any positive constants to maintain the exploitation and exploration behavior of the shark. For this work, the values of $\omega_{f_{max}}$ and $\omega_{f_{min}}$ are kept at 0.31 and 0.03 after in-depth analysis. Sharks have a behavior, which tends to maintain their position closer to the prey:

$$\mathrm{P}_{s+1}^{\grave{a}} = \mathrm{P}_{gb_s} + \left(rand_1 \times \overrightarrow{dist}\right)\left(scd\left(rand_2 - 0.3\right)\right) \tag{16}$$

here, $\mathrm{P}_{s+1}^{\grave{a}}$ denotes the new position of a^{th} shark with-respect-to the nearest prey, $scd\left(rand_2 - 0.3\right)$ will provide a value of -1 or 1, which will update the direction of the search. The direction of search will be updated only if $rand_3 < Sense$. $\overrightarrow{dist}$ is the distance between the a^{th} shark and prey and it is defined as:

$$\overrightarrow{dist} = \left|rand \times \left(\mathrm{P}_{gb_s} - \mathrm{P}_s^a\right)\right| \tag{17}$$

The *Sense* is a parameter, which denotes the key senses of a shark while moving towards the prey and it is defined as:

$$Sense = \left|1 - e^{\frac{-r\times s}{S}}\right| \tag{18}$$

here, r is a positive constant, which is used to manage the behavior of exploitation and exploration of sharks. During the evaluation of this study, the value of r is kept at 0.002.

The behavior of sharks is simulated mathematically by preserving the initial two optimal solutions and updated white shark position w.r.t these optimum solutions. The following equation is used to preserve the stated behavior:

$$\mathrm{P}_{s+1}^a = \frac{\mathrm{P}_s^a + \mathrm{P}_{s+1}^a}{2 \times rand} \tag{19}$$

This relation shows that the position of the shark is always updated w.r.t. the optimal position of prey. The final location of the shark will be somewhere in the search space, near the optimum prey. The final algorithm of iSSO is presented in Algorithm 1.

Algorithm 1: Iterative optimization procedure of iSSO

```
 Input: Feature vector
 Output: Optimized features
Parameter initialization
Initial positions are randomly generated
The population is initialized by assigning initial velocities
```

(Continued)

Algorithm 1 (continued)

```
The population is evaluated on the initial positions
while ($s < S$) do
    Update $\upsilon,\ \Lambda,\ \varsigma_1,\ \varsigma_2,\ g,\ h,\ \mathrm{P}_0,\ \omega_f, mF, Sense$ using the related equations
    for $a = 1$ to a do
        Eq. (5)
    end for
    for $a = 1$ to a do
        Eq. (10)
    end for
    for $a = 1$ to a do
        if $rand \le Sense$
            Eq. (17)
        end if
    end for
    If a shark reaches beyond the search space, adjust its position
    Update the new position by $s = s + 1$
end while
Return optimum solution
```

After extensive experiments, the value of ş and ₣ is set at 14 and 0.65. The impact of these values is also presented in the result section.

3 Experimental Results

The proposed iSSO algorithm is evaluated by performing multiple experiments under different parameters, which efficiently verifies the performance of this algorithm. This section provides an in-depth view of performed experiments along with ablation analysis and comparison with existing techniques.

3.1 Experimental Setup and Datasets

The proposed iSSO algorithm is evaluated on two (2) benchmark datasets including UCF-Sports Dataset (D1) [35] and Hollywood2 Dataset (D2) [36]. D1 contains a total of 150 videos from 10 classes included in this dataset, which represents human actions from different viewpoints and a range of scenes. D2 contains a total of 1,707 videos across 12 classes. These videos are extracted from 69 Hollywood movies.

The proposed iSSO model is trained, tested, and validated using an HP Z440 workstation having an NVIDIA Quadro K2000 with a GPU memory of 2 GB DDR5. This card has 382 CUDA cores along with a 128-bit memory interface and 17 GB/s memory bandwidth. MATLAB2021a was used for training, testing, and validation. All selected pre-trained models are transfer learned with an initial learning rate of 0.0001 with an average decrease of 5% after 7 epochs. The whole process has 160 epochs and overall momentum of 0.45. Selected datasets are split using the standard 70-15-15 ratio for training, testing, and validation. During the testing of the proposed model, eight (8) classifiers were trained, which include Bagged Tree (BTree), Linear Discriminant Analysis (LDA), three kernels of k-Nearest Neighbor (kNN), i.e., Ensemble Subspace kNN (ES-kNN), Weighted kNN (W-kNN) and Fine kNN (F-kNN), and three kernels of Support Vector Machine (SVM), i.e., Cubic SVM (C-SVM), Quadratic SMV (Q-SVM) and Multi-class SVM (M-SVM). The performance of the proposed iSSO algorithm is evaluated using six metrics, such as Sensitivity (Sen), Correct Recognition Rate (CRR), Precision (Pre), Accuracy (Acc), Prediction Time (PT), and Training Time (TT). All experimental results presented in the next section are achieved after performing each experiment at least five times, using the same environment and factors.

3.2 Recognition Results

The efficiency of the proposed model is evaluated by performing multiple experiments. Initially, the impact of all selected pre-trained models is noted by feeding the dataset and extracting features from the selected output layer. In the next experiment, the proposed iSSO algorithm is employed on extracted deep features. And finally, the iSSO-enabled CNN model with the highest accuracy is further forwarded to the other classifiers. It is noteworthy that all the selected classifiers were used during this experiment, but F-kNN achieved the highest accuracy, thus Table 2 contains the results of F-kNN. While using D1, the Na model achieved the highest average Acc of 97.44 was achieved. This average accuracy has a factor, of ±1.36%, which it alters during the five experiments. Similarly, Na obtained 96.97% CRR. The F-kNN took 206 min on average to train and 0.53 s to predict an input image. The lowest average Acc of 73.02% was obtained by the Vg model, whereas Ef took the highest TT of 347 min.

Table 2: Performance of iSSO on selected CNN models on D1

CNN Model	iSSO		Acc (%)	CRR (%)	TT (m)	PT (s)
	No	Yes				
Vg	✓		73.02 ± 1.56	73.76 ± 2.06	240 ± 18	0.76 ± 0.29
		✓	75.53 ± 2.96	75.95 ± 1.27	209 ± 21	0.59 ± 0.15
Mo	✓		79.18 ± 4.59	78.78 ± 1.74	312 ± 36	0.72 ± 0.32
		✓	82.95 ± 1.59	83.28 ± 2.06	277 ± 22	0.57 ± 0.12
Re	✓		83.28 ± 2.73	83.58 ± 2.02	267 ± 16	0.78 ± 0.23
		✓	86.71 ± 1.72	86.76 ± 2.11	266 ± 26	0.59 ± 0.29
Ef	✓		74.51 ± 2.07	74.81 ± 2.56	347 ± 28	0.82 ± 0.19
		✓	77.68 ± 1.74	76.16 ± 1.33	300 ± 19	0.55 ± 0.25
Da	✓		88.86 ± 2.82	89.25 ± 1.77	319 ± 21	0.79 ± 0.22
		✓	91.84 ± 4.61	90.35 ± 1.28	265 ± 12	0.55 ± 0.39
De	✓		92.56 ± 1.64	92.98 ± 1.75	267 ± 22	1.15 ± 0.22
		✓	94.74 ± 1.32	95.04 ± 1.87	234 ± 33	0.72 ± 0.28
Xe	✓		82.22 ± 2.18	82.68 ± 1.68	258 ± 26	0.87 ± 0.33
		✓	84.65 ± 2.20	84.96 ± 2.48	230 ± 40	0.66 ± 0.24
In	✓		80.83 ± 1.84	81.27 ± 1.82	225 ± 30	1.04 ± 0.38
		✓	82.26 ± 2.09	83.89 ± 1.45	**194 ± 10**	0.76 ± 0.33
Na	✓		95.42 ± 1.59	94.94 ± 2.42	254 ± 28	0.71 ± 0.31
		✓	**97.44 ± 1.36**	**96.97 ± 1.82**	206 ± 30	**0.53 ± 0.24**

Once a model with the best performance is selected in the first experiment, this model is used to train all selected classifiers. As mentioned earlier, F-kNN performed better on D1 when Na was selected as the base CNN model. This classifier achieved average Sen of 97.37%, an average CRR of 96.97%, and a Pre of 97.28%. The second-best average Acc of 91.75% was achieved by Es-kNN. The worst-performing classifier was BTree, which could only achieve an 80.83% average Acc. The lowest average TT was of 193 s and the lowest average PT of 0.39 s was taken by LDA, but it could only achieve 84.16% Acc.

The proposed model is also evaluated on D2, where the Da network achieved a maximum average Acc of 80.66%. The change factor of this model is 1.04%, after performing the same experiment 5 times. The average CRR of this model is noted at 79.68%. The best classifier for this model is M-SVM, which took 139 min on average to train and 0.48 s on average to predict an input image. The second-best average Acc of 78.27% is achieved by De, which also achieves 78.66% CRR. For this model, M-SVM took 221 min to train and 0.54 s to predict. The lowest average accuracy of 60.02% on D2 is again achieved by Vg, where the selected classifier took 297 min to train and 1.45 s to predict an input image. The performances of all selected CNN models with and without the iSSO algorithm are compared in Table 3.

Table 3: Performance of iSSO on selected CNN models on D2

CNN Model	iSSO		Acc (%)	CRR (%)	TT (m)	PT (s)
	No	Yes				
Vg	✓		60.02 ± 1.64	60.35 ± 2.63	297 ± 25	1.45 ± 0.17
		✓	63.25 ± 1.07	62.96 ± 2.08	224 ± 17	0.82 ± 0.14
Mo	✓		74.14 ± 1.65	73.85 ± 2.13	292 ± 46	0.88 ± 0.31
		✓	76.83 ± 2.17	76.23 ± 2.38	236 ± 28	0.55 ± 0.12
Re	✓		71.61 ± 2.35	70.19 ± 2.31	245 ± 28	1.19 ± 0.15
		✓	73.82 ± 2.19	74.86 ± 1.96	176 ± 32	0.88 ± 0.26
Ef	✓		68.45 ± 2.16	67.54 ± 2.33	230 ± 12	0.99 ± 0.28
		✓	72.12 ± 1.54	70.69 ± 2.82	192 ± 21	0.63 ± 0.34
Da	✓		77.22 ± 1.36	78.69 ± 2.09	161 ± 42	0.65 ± 0.24
		✓	**80.66 ± 1.04**	**79.68 ± 1.21**	**139 ± 32**	**0.48 ± 0.22**
De	✓		75.23 ± 2.62	75.22 ± 2.76	254 ± 44	0.79 ± 0.39
		✓	78.27 ± 1.93	78.66 ± 1.15	221 ± 27	0.54 ± 0.37
Xe	✓		63.04 ± 1.95	63.33 ± 2.17	212 ± 24	0.73 ± 0.21
		✓	66.86 ± 2.27	66.38 ± 2.68	180 ± 27	0.53 ± 0.14
In	✓		66.14 ± 2.12	67.14 ± 2.39	236 ± 14	0.74 ± 0.16
		✓	69.24 ± 2.04	68.23 ± 2.29	204 ± 24	0.56 ± 0.24
Na	✓		71.28 ± 2.89	71.31 ± 2.67	364 ± 26	0.61 ± 0.23
		✓	75.62 ± 2.38	74.76 ± 1.86	292 ± 48	0.46 ± 0.15

After the selection of the best-performing CNN model, all selected classifiers are trained on the extracted features of that CNN model. During this experiment, selected evaluation matrices are used to note the performance of each classifier. M-SVM has achieved the best average Sen of 79.22%, best average CRR of 79.68%, best Pre of 79.84%, and best average Acc of 80.66%. This classifier requires 280 min for training and 0.48 s for predicting an input image. The second-best average Acc of 75.88% is obtained by W-kNN, which took 280 min to train and 0.36 s to predict. The lowest TT is noted at 115 min for BTree, but the achieved average Acc is 50.95%.

3.3 Ablation Analysis of iSSO

This section discusses the importance of selecting values of parameters used in the iSSO algorithm. It should be noted that all readings of this section are performed using the network, which obtained the highest accuracy for each dataset, i.e., Na for D1 and Da for D2. Secondly, the classifier used for this analysis is also retrieved from the best experiment for each dataset, i.e., f-kNN for D1 and M-SVM for D2. All experiments in this analysis are performed thrice and an average reading of three experiments is mentioned against each parameter.

The first and most important factor of the iSSO algorithm is the number of subparts ş, into which the whole search space, the feature vector, is divided. Table 4 represents the impact of different values for this parameter on accuracy and training time. It is noteworthy that the less value of ş decreases TT but reduces the performance of the algorithm.

Table 4: Impact of different values of ş

Value of ş	D1		D2	
	Acc (%)	TT (m)	Acc (%)	TT (m)
12	96.3	227	78.6	124
13	98.1	234	80.4	140
14	**99.8**	**248**	**81.7**	**157**
15	95.6	267	79.1	181
16	94.5	290	74.8	219

Another important parameter is ₣, which selects the total number of features after the completion of an algorithm. The impact of ₣ on TT and Acc is shown in Table 5. It is visible that with the increase of selected features, the Acc and TT increase for both datasets until the value of ₣ reaches 0.65.

Table 5: Impact of different values of ₣

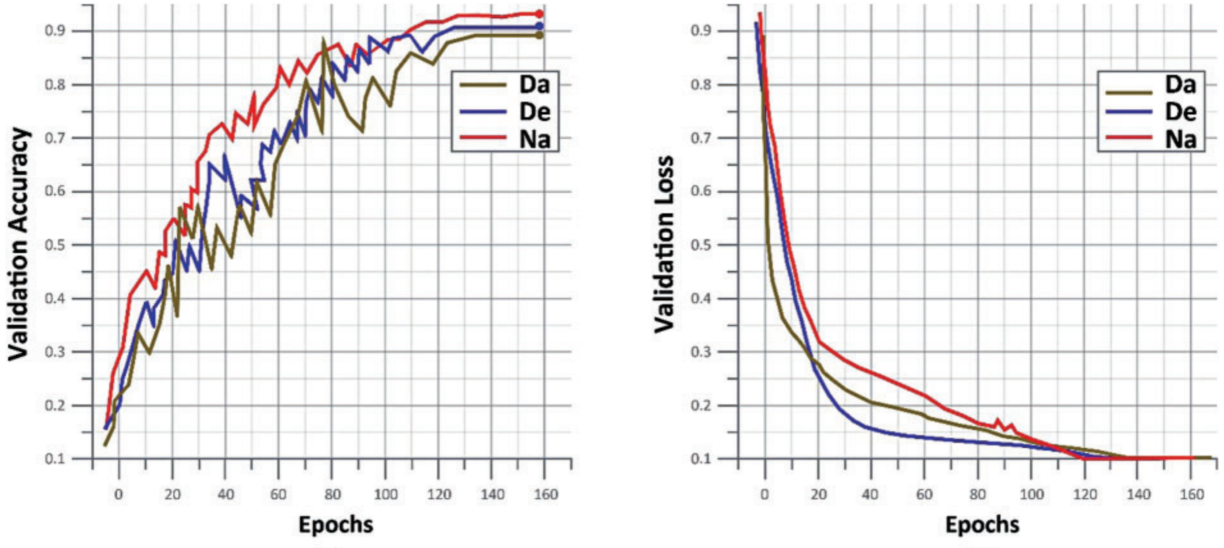
Value of ₣	Selected features of Na from 4032	D1		Selected features of Da from 1024	D2	
		Acc (%)	TT (m)		Acc (%)	TT (m)
0.55	2218	97.1	235	564	79.7	138
0.60	2420	98.6	241	615	80.7	149
0.65	**2621**	**99.8**	**248**	**666**	**81.7**	**157**
0.70	2833	98.9	255	717	80.2	167
0.75	3025	98.1	263	768	78.9	178

The coefficient of acceleration ℂ determines how quickly the shark will move from its current position. The quicker the movement is, the less exploration it will make. The acceleration must neither be too fast nor too slow, as the faster shark will skip important and potential prey and slower sharks will take too much time in exploration. Another factor is the behavior of sharks *r* during the exploitation and exploration process. The value of *r* determines the intervals, by which each prey should be searched for. Lesser value of *r* will increase the searching time and ultimately increases the TT. Table 6 represents the comparison of different values of ℂ and *r*.

Table 6: Impact of different values of ş

Value of ℂ	D1		D2		Value of *r*	D1		D2	
	Acc (%)	TT (m)	Acc (%)	TT (m)		Acc (%)	TT (m)	Acc (%)	TT (m)
2.135	92.0	163	71.5	75	0.2	82.5	117	61.9	47
2.140	96.5	208	78.7	113	0.02	93.0	182	75.4	95
2.145	**99.8**	**248**	**81.7**	**157**	**0.002**	**99.8**	**248**	**81.7**	**157**
2.150	95.6	378	78.6	208	0.0002	96.1	392	78.0	267
2.155	94.5	549	70.6	294	0.00002	95.0	674	75.3	397

The values of ς_{max}, ς_{min}, $\omega_{f_{max}}$ and $\omega_{f_{min}}$ do not majorly impact the overall performance of iSSO, specifically in terms of Acc and TT. At the selected values of these parameters, the iSSO has obtained the highest possible performance. Tweaking these parameters marginally changes the results, which can be ignored. The validation accuracy and validation loss of the proposed model on both datasets are shown in Fig. 2, where Figs. 2a and 2b are the validation accuracy and validation loss on D1, respectively, while Figs. 2c and 2d are the validation accuracy and validation loss on D2, respectively. It can be seen that 50% accuracy on both datasets is achieved on the initial 40 epochs, the validation loss is also reduced to less than 50% in the same number of epochs, which shows the high convergence of the proposed model.



(a) (b)

Figure 2: (Continued)

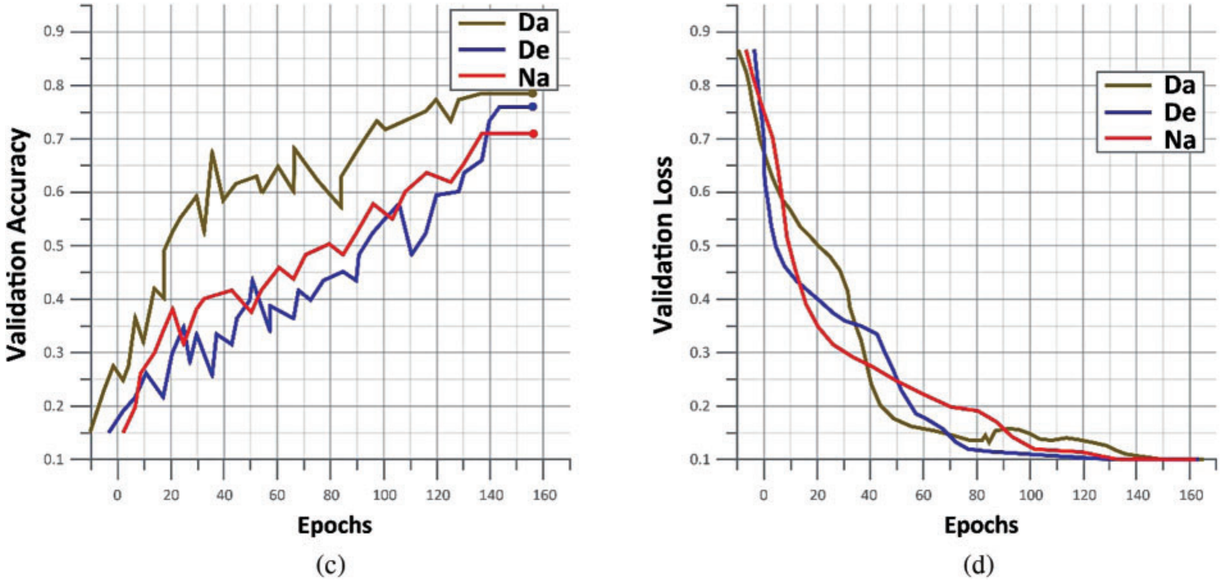


Figure 2: Validation accuracy and validation loss on D1 and D2

3.4 *Comparison with Existing Techniques*

A hybrid model was proposed in [37] by combining Speeded Up Robust Features (SURF) and Histogram of Oriented Gradients (HOG) for HAR. This model was cable of extracting global and local features as it obtained motion regions by adopting background subtraction. Motion edge features, effectively described by the directional controllable filters were utilized in HOG to extract information on local edges. The bag of Word (BoW) model was also obtained by performing k-means clustering. In the end, Support Vector Machines (SVM) were used to recognize the motion features. This model was tested on SBU Kinect Interaction, UCF Sports, and KTH datasets and achieved accuracies of 98.5%, 97.6%, and 98.2%, respectively. QWSA-HDLAR model was proposed in [38] for the recognition of human actions. This model utilized TL-enabled CNN architecture, called NASNet for feature extraction. The NASNet model also employs a tuning process for hyper-parameters to optimally increase performance. In the end, a hybrid model containing CNN and RNN, called CNN-BiRNN, was used to classify different human actions. This model was tested on D1 and KTH, and it achieved an average recognition rate of 99.0% and 99.6% on both datasets, respectively.

An attention mechanism based on bi-directional LSTM (BiLSTM) and dilated CNN (dCNN) was proposed in [39], which extracted effective features of the HAR frame. Salient features were extracted using the dCNN and these features were fed to the BiLSTM model for the learning process. The learning process helped the model for long-term dependencies, which boosted the evaluation performance and extracted HAR-related cues and patterns. This model was evaluated on J-HMDB, D1, and UCF11 and achieved 80.2%, 99.1%, and 98.3% accuracies, respectively. A DCNN-based model was proposed in [40], which took the input of globally contrasted frames. The resnet-50 model was transferred and learned and it extracted features from a fully connected and global average pooling layer. Both features were fused using Canonical Correlation Analysis (CCA) and then fine-tuned using the Shanon Entropy-based technique. The proposed model was tested on KTH, UT-Interaction, YouTube, D1, and IXMAS datasets and achieved accuracies of 96.6%, 96.7%, 100%, 99.7%, and 89.6%, respectively. The authors in [41] proposed the HAR model using feature fusion and optimization techniques. Before feature engineering, the color transformation was applied to enhance the video frames. Optical flow extracted the moving region after the frames fusion, and these regions were forwarded to extract texture and shape features. Finally, weighted entropy was utilized to select related features and M-SVM was used to classify the actions. This model experimented on UCF YouTube, D1, KTH, and Weizmann datasets and it achieved 94.5%, 99.3%, 100%, and 94.5%, respectively. Table 7 compares the proposed model with existing techniques.

Table 7: Comparison with existing techniques on D1

Method	Year	CRR (%)
Hybrid model using SURF and HOG [37]	2022	97.6
QWSA-HDLAR [38]	2022	99.0
HAR using LSTM and dCNN [39]	2021	99.1
Resnet-50 with CCA and shanon entropy [40]	2021	99.7
Length control features using weighted-entropy [41]	2020	99.3
STA3D [42]	2021	90.0
3 Module model [43]	2021	88.9
Proposed	**-**	**99.8**

HAR was carried out using three models in [44] including where extraction of compact features, re-sampling of shot framerate, and detection of the shot boundary. The main objective of this research was to emphasize the extraction of relevant features. This model was tested on Weizmann, UCF, KTH, and D2 datasets using the second model, it achieved 97.8%, 95.6%, 97.0%, and 73.6% accuracies, respectively. A lightweight deep learning model was proposed in [45], which recognizes human actions using surveillance streams of CNN models. An ultra-fast object recognizer named Minimum-Output-Sum-of-Squared-Error (MOSSE) locates the subject in a video, while the LiteFlowNet CNN model was used to extract pyramid convolutional features of successive frames. In the end, Gated Recurrent Unit (GRU) was trained to perform HAR. Experiments were conducted on YouTube, Hollywood2, UCF-50, UCF-101 and HMDB51 datasets and overall average accuracy of 97.1%, 71.3%, 95.2%, 95.5% and 72.3%, respectively.

Double-constrained BOW (DC-BOW) was presented in [46], which utilized spatial information of features on three different scales including hidden scale, presentation scale, and descriptor scale. Length and Angle Constrained Linear Coding (LACLC) methods were obtained by constructing a loss function between local features and visual words. To optimize the features, spatial differentiation between extracted features of every cluster was considered. LACLC and a hierarchical weighted approach were applied to extract the related features. The proposed model was tested on UCF101, D2, UCF11, Olympic Sports, and KTH datasets and it achieved accuracies of 88.9%, 67.13%, 96%, 92.3%, and 98.83%, respectively. A Spatiotemporally Attentive 3D Network (STA3D) was proposed in [42] for the propagation of important temporal descriptors and refining of spatial descriptors in 3D Fully Convolutional Networks (3D-FCN). To refine spatial descriptors and propagate temporal descriptors, an adaptive up-sampling module was also proposed. This technique was evaluated on D1 and D2, where it achieved 90% and 71.3% accuracies, respectively. A DCNN-based model is proposed in [43], which has three modules, reasoning and memory, attention, and high-level representation modules. The first modules concentrated on temporal and spatial reasoning so that temporal and spatial patterns could be efficiently discriminated. The second and third modules were mainly utilized for learning through captured spatial saliencies. This model was evaluated on D1 and D2, where it achieved 88.9% and 78.9% accuracies. Table 8 compares the performance of the proposed model with existing techniques.

Table 8: Comparison with existing techniques on D2

Method	Year	Acc (%)
Shot framerate and shot boundary [44]	2021	73.6
DS-GRU [45]	2021	71.3
CS-BOW [46]	2021	67.1
STA3D [42]	2021	71.3
3 Module Model [43]	2021	78.9
Proposed	**-**	**81.7**

4 Conclusion

In this article, an analysis of pre-trained CNN models is presented, where 9 models are selected based on their total parameters, size, and Top-1 and Top-5 accuracies. These selected pre-trained CNN models are trained on the selected dataset using the TL. The output layer of these pre-trained models is mentioned, and no experiments are performed based on a selection of the output layer. The extracted features of these CNN models are forwarded to the proposed iSSO, which is an improved algorithm from the traditional SSO. The iSSO algorithm divides the feature vector into subsets, where each subset is then used to find the local and global best features. The selection of local and global best features is inspired by the searching capabilities of the white shark, which uses its senses to find the optimal prey. Once the features are selected, the results are taken using selected publicly available datasets. The limitation of this work is the training time, which is too high, i.e., the lowest training time for D1 is 194 min and for D2, it is 139 min. The one reason for taking this much TT is the dataset, which includes videos. But the main reason is the architecture of these models, which have too many repeated blocks of layers, which can be reduced. In the future, the architecture of the best-performing CNN models of this article will be analyzed to detect and reduce the repeated blocks of layers. The impact of these repeated blocks can also be analyzed.

Acknowledgement: Not applicable.

Funding Statement: This work was supported by the Collabo R&D between Industry, Academy, and Research Institute (S3250534) funded by the Ministry of SMEs and Startups (MSS, Korea), the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2023-00218176), and the Soonchunhyang University Research Fund.

Author Contributions: The authors confirm their contribution to the paper as follows: study conception and design: I.M.N, M.A.K, and M.R; data collection: I.M.N, M.A.K, and M.R; draft manuscript preparation: I.M.N, M.A.K, M.R, and J.H.S; funding: J-C.N and Y.N; validation: JH.S, Y-C.N, and Y.N; software: I.M.N, M.A.K, Y.N, and Y-C.N; visualization: JH.S, Y-C.N, and Y.N; supervision: M.A.K, M.R and Y.N. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The dataset used in this work is publically available for research purpose.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] V. Sharma, M. Gupta, A. K. Pandey, D. Mishra and A. Kumar, "A review of deep learning-based human activity recognition on benchmark video datasets," *Applied Artificial Intelligence*, vol. 36, no. 1, pp. 1–17, 2022.

[2] S. K. Yadav, K. Tiwari, H. M. Pandey and S. A. Akbar, "A review of multimodal human activity recognition with special emphasis on classification, applications, challenges and future directions," *Knowledge-Based Systems*, vol. 223, no. 11, pp. 51–83, 2021.

[3] I. M. Nasir, M. Raza, J. H. Shah, S. H. Wang, U. Tariq *et al.,* "HAREDNet: A deep learning based architecture for autonomous video surveillance by recognizing human actions," *Computers and Electrical Engineering*, vol. 99, no. 1, pp. 1–16, 2022.

[4] M. Raza, J. H. Shah, M. A. Khan and A. Rehman, "Human action recognition using machine learning in uncontrolled environment," *Artificial Intelligence and Data Analytics*, vol. 12, no. 3, pp. 182–187, 2021.

[5] P. Pareek and A. Thakkar, "A survey on video-based human action recognition: Recent updates, datasets, challenges, and applications," *Artificial Intelligence Review*, vol. 54, no. 3, pp. 2259–2322, 2021.

[6] A. Sarkar, A. Banerjee, P. K. Singh and R. Sarkar, "3D human action recognition: Through the eyes of researchers," *Expert Systems with Applications*, vol. 1, no. 2, pp. 11–42, 2022.

[7] I. U. Khan, S. Afzal and J. W. Lee, "Human activity recognition via hybrid deep learning based model," *Sensors*, vol. 22, no. 1, pp. 323–349, 2022.

[8] Z. Fu, X. He, E. Wang, J. Huo, J. Huang *et al.,* "Personalized human activity recognition based on integrated wearable sensor and transfer learning," *Sensors*, vol. 21, no. 3, pp. 885–903, 2021.

[9] H. Wang, B. Yu, K. Xia, J. Li and X. Zuo, "Skeleton edge motion networks for human action recognition," *Neurocomputing*, vol. 423, no. 3, pp. 1–12, 2021.

[10] Y. B. Cheng, X. Chen, D. Zhang and L. Lin, "Motion-transformer: Self-supervised pre-training for skeleton-based action recognition," *Multimedia Tools and Applications*, vol. 4, no. 1, pp. 1–32, 2021.

[11] K. Liu, L. Gao, N. M. Khan, L. Qi and L. Guan, "Integrating vertex and edge features with graph convolutional networks for skeleton-based action recognition," *Neurocomputing*, vol. 466, no. 13, pp. 190–201, 2021.

[12] T. Xue and H. Liu, "Hidden markov model and its application in human activity recognition and fall detection: A review," *Communications, Signal Processing, and Systems*, vol. 1, no. 1, pp. 863–89, 2022.

[13] C. Yin, J. Chen, X. Miao, H. Jiang and D. Chen, "Device-free human activity recognition with low-resolution infrared array sensor using long short-term memory neural network," *Sensors*, vol. 21, no. 10, pp. 3551–3561, 2021.

[14] A. Anagnostis, L. Benos, D. Tsaopoulos, A. Tagarakis, N. Tsolakis *et al.,* "Human activity recognition through recurrent neural networks for human–robot interaction in agriculture," *Applied Sciences*, vol. 11, no. 5, pp. 2188–2205, 2021.

[15] W. Ding, C. Ding, G. Li and K. Liu, "Skeleton-based square grid for human action recognition with 3D convolutional neural network," *IEEE Access*, vol. 9, no. 3, pp. 54078–54089, 2021.

[16] D. Tran, L. Bourdev, R. Fergus, L. Torresani and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," *Image and Vision Computing*, vol. 13, no. 5, pp. 4489–4497, 2019.

[17] A. Graves, A. R. Mohamed and G. Hinton, "Speech recognition with deep recurrent neural networks," *Acoustics, Speech and Signal Processing*, vol. 1, no. 1, pp. 6645–6649, 2013.

[18] A. Graves, "Long short-term memory," in *Supervised Sequence Labelling with Recurrent Neural Networks*, 1st ed., vol. 385. Berlin, DEU: Springer, pp. 37–45, 2012.

[19] C. Feichtenhofer, A. Pinz and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," *Computer Vision and Pattern Recognition*, vol. 2, no. 1, pp. 1933–1941, 2012.

[20] L. Sun, K. Jia, D. Y. Yeung and B. E. Shi, "Human action recognition using factorized spatio-temporal convolutional networks," *Computer Vision and Applications*, vol. 1, no. 1, pp. 4597–4605, 2015.

[21] V. Veeriah, N. Zhuang and G. J. Qi, "Differential recurrent neural networks for action recognition," *Computer Vision amd Applications*, vol. 1, no. 1, pp. 4041–4049, 2015.

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh *et al.,* "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 1409–1556, 2014.

[24] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," *Computer Vision and Pattern Recognition*, vol. 2, no. 1, pp. 4510–4520, 2018.

[25] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," *Computer Vision and Pattern Recognition*, vol. 1, no. 13, pp. 770–778, 2016.

[26] D. Jia, W. Dong, R. Socher, L. J. Li, K. Li *et al.,* "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conf. on Computer Vision and Pattern Recognition*, IEEE, pp. 248–255, 2009.

[27] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," *Machine Learning Tools*, vol. 1, no. 1, pp. 6105–6114, 2019.

[28] J. Redmon, "Darknet: Open source neural networks in c," 1st ed., vol. 3. New York, USA: Springer, pp. 152–183, 2013.

[29] G. Huang, Z. Liu, L. V. D. Maaten and K. Q. Weinberger, "Densely connected convolutional networks," *Computer Vision and Pattern Recognition*, vol. 3, no. 12, pp. 4700–4708, 2017.

[30] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *Computer Vision and Pattern Recognition*, vol. 3, no. 11, pp. 1251–1258, 2017.

[31] C. Szegedy, S. Ioffe, V. Vanhoucke and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," *Artificial Intelligence*, vol. 1, no. 1, pp. 1–13, 2017.

[32] B. Zoph, V. Vasudevan, J. Shlens and Q. V. Le, "Learning transferable architectures for scalable image recognition," *Computer Vision and Pattern Recognition*, vol. 3, no. 11, pp. 8697–8710, 2018.

[33] O. Abedinia, N. Amjady and A. Ghasemi, "A new metaheuristic algorithm based on shark smell optimization," *Complexity*, vol. 21, no. 5, pp. 97–116, 2016.

[34] S. Wroe, D. R. Huber, M. Lowry, C. McHenry, K. Moreno *et al.,* "Three-dimensional computer analysis of white shark jaw mechanics: How hard can a great white bite?," *Journal of Zoology*, vol. 276, no. 4, pp. 336–342, 2008.

[35] K. Soomro and A. R. Zamir, "Action recognition in realistic sports videos," *Computer Vision in Sports*, vol. 1, no. 1, pp. 181–208, 2014.

[36] M. Marszalek, I. Laptev and C. Schmid, "Actions in context," *Computer Vision and Pattern Recognition*, vol. 1, no. 1, pp. 2929–2936, 2009.

[37] J. Zhao, "Sports motion feature extraction and recognition based on a modified histogram of oriented gradients with speeded up robust features," *Journal of Computers*, vol. 33, no. 1, pp. 63–70, 2022.

[38] A. A. Alibari, J. S. Alzahrani, A. Qahmash, M. Maray, M. Alghamdi *et al.,* "Quantum water strider algorithm with hybrid-deep-learning-based activity recognition for human–computer interaction," *Applied Sciences*, vol. 12, no. 14, pp. 6848, 2022.

[39] K. Muhammad, A. Ullah, A. S. Imran, M. Sajjad, M. Kiran *et al.,* "Human action recognition using attention based LSTM network with dilated CNN features," *Future Generation Computer Systems*, vol. 125, no. 1, pp. 820–830, 2021.

[40] S. Kiran, M. A. Khan, M. Y. Javed, M. Alhaisoni, U. Tariq *et al.,* "Multi-layered deep learning features fusion for human action recognition," *Computers, Materials & Continua*, vol. 69, no. 3, pp. 4061–4075, 2021.

[41] F. Afza, M. A. Khan, M. Sharif, S. Kadry, G. Manogaran *et al.,* "A framework of human action recognition using length control features fusion and weighted entropy-variances based feature selection," *Image and Vision Computing*, vol. 106, no. 1, pp. 104090–104105, 2021.

[42] W. Zou, S. Zhuo, Y. Tang, S. Tian, X. Li *et al.,* "STA3D: Spatiotemporally attentive 3D network for video saliency prediction," *Pattern Recognition Letters*, vol. 147, no. 3, pp. 78–84, 2021.

[43] J. Chen, Z. Li, Y. Jin, D. Ren and H. Ling, "Video saliency prediction via spatio-temporal reasoning," *Neurocomputing*, vol. 462, no. 2, pp. 59–68, 2021.

[44] C. A. Aly, F. S. Abas and G. H. Ann, "Robust video content analysis schemes for human action recognition," *Science Progress*, vol. 104, no. 2, pp. 5480–5501, 2021.

[45] A. Ullah, K. Muhammad, W. Ding, V. Palade, I. U. Haq *et al.,* "Efficient activity recognition using lightweight CNN and DS-GRU network for surveillance applications," *Applied Soft Computing*, vol. 103, no. 1, pp. 107–122, 2021.

[46] C. Wu, Y. Li, Y. Zhang and B. Liu, "Double constrained bag of words for human action recognition," *Signal Processing: Image Communication*, vol. 98, no. 3, pp. 1163–1199, 2021.