Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.038120
Article

Human Gait Recognition Based on Sequential Deep Learning and Best Features Selection

Ch Avais Hanif[1], Muhammad Ali Mughal[1,*], Muhammad Attique Khan[2], Usman Tariq[3], Ye Jin Kim[4] and Jae-Hyuk Cha[4]

[1]Department of Electrical Engineering, HITEC University, Taxila, Pakistan
[2]Department of Computer Science, HITEC University, Taxila, Pakistan
[3]College of Computer Engineering and Science, Prince Sattam Bin Abdulaziz University, Al-Kharaj, 11942, Saudi Arabia
[4]Department of Computer Science, Hanyang University, Seoul, 04763, Korea

*Corresponding Author: Muhammad Ali Mughal. Email: ali.mughal@hitecuni.edu.pk
Received: 28 November 2022; Accepted: 22 February 2023

Abstract: Gait recognition is an active research area that uses a walking theme to identify the subject correctly. Human Gait Recognition (*HGR*) is performed without any cooperation from the individual. However, in practice, it remains a challenging task under diverse walking sequences due to the covariant factors such as normal walking and walking with wearing a coat. Researchers, over the years, have worked on successfully identifying subjects using different techniques, but there is still room for improvement in accuracy due to these covariant factors. This paper proposes an automated model-free framework for human gait recognition in this article. There are a few critical steps in the proposed method. Firstly, optical flow-based motion region estimation and dynamic coordinates-based cropping are performed. The second step involves training a fine-tuned pre-trained MobileNetV2 model on both original and optical flow cropped frames; the training has been conducted using static hyperparameters. The third step proposed a fusion technique known as normal distribution serially fusion. In the fourth step, a better optimization algorithm is applied to select the best features, which are then classified using a Bi-Layered neural network. Three publicly available datasets, CASIA A, CASIA B, and CASIA C, were used in the experimental process and obtained average accuracies of 99.6%, 91.6%, and 95.02%, respectively. The proposed framework has achieved improved accuracy compared to the other methods.

Keywords: Human gait recognition; optical flow; deep learning features; fusion; feature selection

1 Introduction

Human gait recognition is a biometric application that goals to identify pedestrians by their walking pattern [1,2]. In video surveillance, gait is a distinctive biological trait with numerous

applications, such as forensic identification, criminal investigation, and crime prevention [3]. The significant advantage of human gait recognition is that it facilitates the identification of a person from a distance [4]. Another advantage of the technique is that it can be utilized for low-resolution video sequences. In human motion, gait describes the temporal dynamic and spatial statics. Gait recognition has proven to be more efficient than other biometrics, such as iris, face, and fingerprint, which require more pixels to identify humans from a distance [5]. The viability of such a technique in the circumstances such as the COVID-19 pandemic cannot be over-emphasized. Surveillance systems augmented with such features can contribute to an excellent deal for ensuring our public security [6].

Nowadays, scientists and researchers use machine learning (ML) and Deep learning (DL) models in several applications [7], including agriculture [8], cyber security [9], environment [10], medicine [11], and text sentiment analyses [12]. Gait recognition is a biometric approach and has wildly succeeded in deep learning. The primary catalyst of this success is the massive amount of open-source data that is publicly available and is suitable for deep learning (DL) models. DL features are directly obtained from the sequential input instead of silhouette images [13–16]. Traditional techniques for these images are more suitable for handcrafted features, such as template-based techniques [17]. With the critical contribution of DL methods, recognition performance has been substantially improved. Once, gauged against commonly used benchmarks of performance, the feasibility of Gait recognition as impressive tools for safety of public is evident. The recent maximum recognition accuracy on CASIA-B dataset [18] is 93% on all selected angles using deep learning; therefore, still a gap is available in the improvement of accuracy [19].

However, based on recent studies, researchers faced several challenges that affected the accuracy of the introduced methods and the computational time [20]. These included humans wearing different clothes, changing camera perspectives, and carrying objects like bags. To tackle these problems, appearance-based [21,22] and model-based [23] methods have been introduced. The model-based techniques extract features from motion patterns and body structure. These are insensitive to external factors like carrying, wearing, and clothing variations. But making a precise body model is complex and computationally inefficient. The practical techniques of model-based are template-based long short-term memory (LSTM), video-based [21,22], gait energy image (GEI), and motion history images. An LSTM-based network that maintains the spatiotemporal gait information was proposed [21]; researchers incorporated a person's gait sequences in various scenarios to directly extract the global features from videos because they predicted the sequential limitations of gait were not necessary for recognition. Handcrafted engineering is always time-consuming and improved by deep learning [24,25]. Fig. 1 shows different processes of gait recognition, such as handcrafted features, deep learning through GEI images, and deep end-to-end understanding. The end-to-end deep learning process is better with the inherent demerit of many chances to extract irrelevant features. Based on this figure, still several challenges exists including ignoring the overlapped body components and wrongly foreground segmentation [26].

This article proposes a sequential end-to-end framework for human gait recognition. Our significant contributions include the following:

- Optical flow-based motion region extraction that later cropped using dynamic coordinates.
- Trained two models using fixed hyperparameters on original frames and optical flow cropped frames.
- A serially normal distribution-based fusion approach.
- An improved monarch butterfly optimization algorithm for the best feature selection.

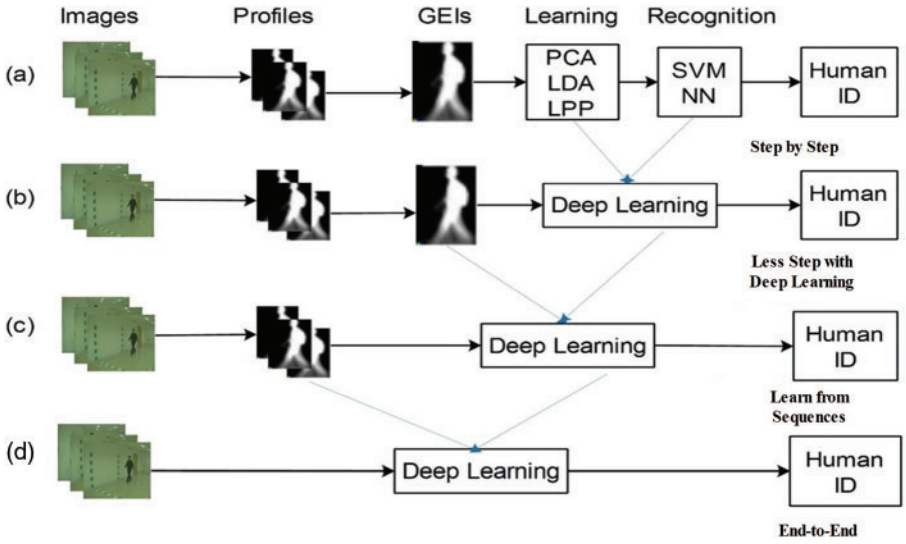

Figure 1: The four typical workflows on deep gait recognition [26]

The rest of this article is organized as follows: Related work of this manuscript is discussed under Section 2. Section 3 describes the proposed methodology. In Section 4, the results have been discussed in detail. And finally, Section 5 of the manuscript represents its conclusion.

2 Related Work

In computer vision, several techniques have been introduced for video surveillance applications, such as dehazing [27], action recognition [28,29], and gait recognition. The literature has introduced a variety of HGR employing deep learning techniques. Mehmood et al. [30] presented a hybrid technique to solve the change in variation problem for accurate gait recognition. DenseNet 201 was utilized to compute the gait attributes for the image frames. Two layers such as FC1000 and Avg._pool were used for the feature extraction that later merged through a parallel approach. Firefly and Kurtosis-based feature selection technique was introduced that not only improved the accuracy but also minimized the computational time during the testing phase (classification phase).

A multilayer feature fusion and accurate region of interest (ROI) segmentation technique was presented for HGR by Sharif et al. [31]. In this approach, ROI was segmented at the initial phase and later utilized for feature extraction. This process was executed slowly, which was the limitation of this work. Liao et al. [23] presented a pose gait model for HGR. In this method, they tried to solve the differences in human gait. The data was captured from various perspectives using the 3D models and extracted 3D body joints for feature extraction. Rani et al. [32] presented an artificial neural network (ANN) based HGR method for identifying a person through a walking pattern. The background subtraction did through the image processing technique that was later converted into binary for silhouette extraction. The system is evaluated through the self-similarity-based method and on the CASIA-B dataset, which showed improvement in accuracy. Deng et al. [33] suggested a reliable gait recognition technique based on global and local image entropy features. Binary walking silhouettes were extracted and later used for the global and local entropy features. Both local and global entropy features were combined and fused with deep learning features for a better informative matrix. The CASIA B dataset was employed in this method for the experimental process and obtained

improved accuracy. Anusha et al. [34] suggested a method for HGR based on the multi-level features. They extracted the low-level features through the texture, spatial and gradient information, which were later combined in one matrix. The experiments were conducted on five different datasets, CASIA A, CASIA B, CMV MoBo, rotal dataset (KTH) and OU-ISIR D video dataset, and obtained improved accuracy of 99.8%, 99.7%, 92.2% and 93.3%, respectively.

Sharif et al. [35] suggested a novel framework for HGR. They performed three key steps such as (a) capturing videos in real-time, (b) extracting features using transfer learning on a deep model named ResNet 101, and (c) features selected through the Kurtosis-Controlled entropy (KcE) approach that followed the correlation-based feature fusion. The presented method was tested in real-time and selected angles of the CASIA-B dataset and obtained an accuracy of 96.6% and 95.26%, respectively. Mehmood et al. [36] presented a hybrid technique using deep learning and selecting the best features to overcome the problems of HGR. The methods consisted of four significant steps: preprocessing of video frames, Pre-trained convolutional neural network (CNN) model named VGG16 used for the features extraction, removing the unnecessary features, and last, classification. Principal Score and Kurtoseis-based technique were utilized for reducing the irrelevant features that was the main strength of this work. Fusion was performed at the end and performed classification through one against all support vector machine (SVM) classifiers.

Khan et al. [37] presented a deep learning (DL) and Improved Ant Colony Optimization (IACO) framework for HGR. The proposed method consisted of four main steps such as (a) normalization of the database in video frames, (b) two pre-trained models were selected named InceptionV3 and ResNet101, and (c) features were extracted that were later optimized using an IACO approach. The experiments were performed on three angles such as 0, 18, and 180, of the CASIA-B dataset and obtained improved accuracy. For multi-view HGR, Zhao et al. [38] presented a graph-based method. The data were captured in a single view using the Spider web graph connected with other views of gait data concurrently. This process was complex, but the performance was improved on the captured datasets. Finally, a novel gait-recognition technique called Conv-LSTM was developed by Wang et al. [39]. They first introduced GEI-based frame extraction for each gait cycle. Then, they analyzed the cross-covariance of a single subject. Ultimately, they designed a Conv-LSTM model for the final gait recognition process. The CASIA-B and OU-ISIR datasets were utilized during the experiments and acquired an accuracy of 93% and 95%, respectively. Finally, Arshad et al. [40] described a fuzzy entropy-controlled skewness (FEcS) and deep neural networks-based framework for HGR. They focused on improving accuracy and reducing the computational time during the classification process. The boundaries of this work were the selection of deep models for feature extraction from raw images. Moreover, several other techniques have been introduced for HGR, such as the ensemble-based technique [41], the multi-source information fusion approach [42], and named a few more [43].

In summary, the above techniques faced the challenge of complex datasets, huge similarity among several gaits of different subjects, and selection of essential features. Moreover, they tried reducing the computational time by employing feature selection techniques. However, still, due to covariant factors and the addition of video frames, this challenge is active for more research. Also, several researchers used only a few angles of the CASIA-B dataset for the experimental process; however, there was a considerable gap in accuracy when the experimentation was conducted on all angles. Based on these challenges, a clear gap is observed that is considered in this work. This paper proposes an improved optimization technique that enhances the accuracy and decreases the computational time during the classification phase to tackle the abovementioned issues.

3 Proposed Methodology

In this section, the proposed deep learning and improved optimization-based framework have been presented for HGR. Fig. 2 shows the framework of the proposed HGR. This figure illustrates that inputs are passed to fine-tuned MobileNet V2 architecture in two ways, i.e., raw images and optical-flow-based motion estimation. First, the estimated motion regions are cropped later for the deep learning model training. Deep transfer learning-based training is performed with fixed hyperparameters. Features are extracted from both trained models that are later fused using a proposed high-index fusion approach. An improved butterfly optimization technique is chosen later, and the best features are selected. The selected features are classified using a Bi-Layered neural network.

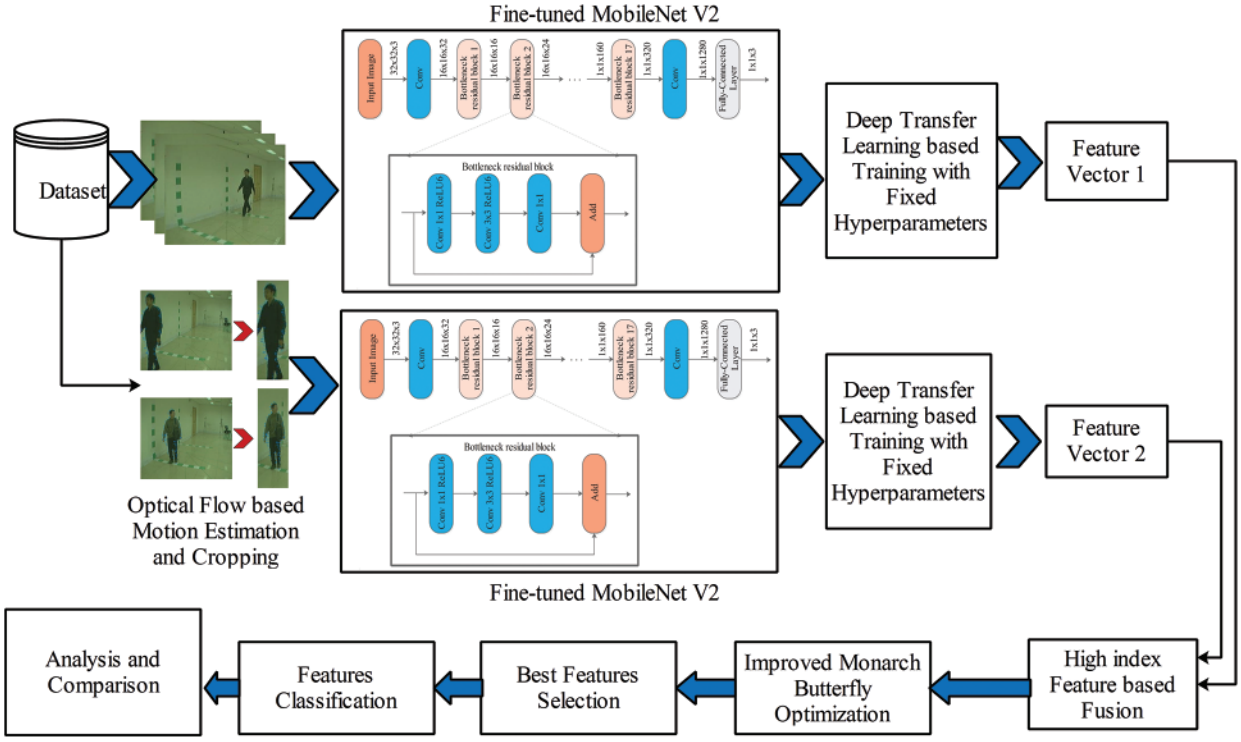

Figure 2: Proposed architecture for human gait recognition using deep learning and improved optimization algorithm

3.1 Motion Extraction and Cropping

In this article, as presented in Fig. 2, motion is extracted from the original video sequences and later cropped for input to the selected pre-trained fine-tuned model. The primary purpose of this phase is to get the temporal information of the subject during the movement that is later fused with raw features to get better accuracy. The Horn-Schunck optical flow (OF) method is employed for motion extraction. This method estimated the motion region that four coordinates have cropped. Mathematically, OF is defined as follows:

$$F_u x + F_v y + F_t = 0 \tag{1}$$

where, F_u, F_v, and F_t denote the spatiotemporal image brightness derivatives. The horizontal and vertical OF are represented by x and y, respectively. The Horn-Schunck method estimates the velocity

for $[x,\ y]^T$ as follows:

$$e = \iint (F_u x + F_v y + F_t)^2 \, dudv + \alpha \iint \left\{ \left(\frac{dx}{du}\right)^2 + \left(\frac{dx}{dv}\right)^2 + \left(\frac{dy}{du}\right)^2 + \left(\frac{dy}{dv}\right)^2 \right\} dudv \tag{2}$$

where, $\frac{dx}{du}$ and $\frac{dx}{dv}$ are two spatiotemporal derivatives, and α denotes the global smoothing term, respectively. This equation can be further minimized as follows:

$$x_{u,v}^{k+1} = x_{u,v}^{-k} - \frac{F_u \left[F_u x_{u,v}^{-k} + F_v y_{u,v}^{-k} + F_t\right]}{\alpha^2 + F_u^2 + F_v^2} \tag{3}$$

$$y_{u,v}^{k+1} = y_{u,v}^{-k} - \frac{F_v \left[F_u x_{u,v}^{-k} + F_v y_{u,v}^{-k} + F_t\right]}{\alpha^2 + F_u^2 + F_v^2} \tag{4}$$

where $\left[x_{u,v}^k\ y_{u,v}^k\right]$ denotes the velocity estimation for each pixel (u, v). $\left[x_{u,v}^{-k} y_{u,v}^{-k}\right]$ denotes the neighborhood average of $x_{u,v}^k$ and $y_{u,v}^k$. The initial value of $k = 0$. To solve u and v, already defined filters are utilized [44]. Sample visual effects of this process are shown in Fig. 3. The motion regions are further cropped through dynamic coordinates, and visual images are illustrated in Fig. 4. These cropped images are later passed to a pre-trained fine-tuned model for the feature extraction.

Figure 3: Optical flow-based motion estimation

3.2 Fine-Tuned MobileNet-V2 Features

In 2018, Google introduced MobileNetV2, which contains 53 deep layers [45]. This network showed improved performance for object recognition, segmentation, and classification [46]. This network accepts an input image of 224 × 224. A contracting path (left side) and a classifier head make up the model (right side). By repeated application, two 3 × 3 convolutions (unpadded convolutions), each followed by a rectified linear unit (ReLU), and a 2 × 2 max pooling operation with stride 2 for downsampling. A set of fully connected layers is produced from repeatedly performing these three processing steps, referred to as "blocks," which makes the network deep (classifier stage). Convolution layers compute filters repeatedly applied over the whole dataset to increase training efficiency, resulting in locally weighted sums (referred to as "feature maps") at every layer. The nonlinear layers then enhance the nonlinear properties of the feature maps. Finally, the largest element in the rectified feature

map is chosen by employing max pooling. Instead of only using the largest element, one may use the average pooling [47]. Fig. 5 shows the architecture of MobileNet-V2. This network was initially trained using the ImageNet dataset. There are 1000 object classes in this dataset; therefore, the network output layer contains 1000 outputs. We fine-tuned this network and removed the last three layers. Afterward, three new layers are added and trained on original and cropped OF-based frames using transfer learning (TL).

Figure 4: Cropped motion estimated regions for model training

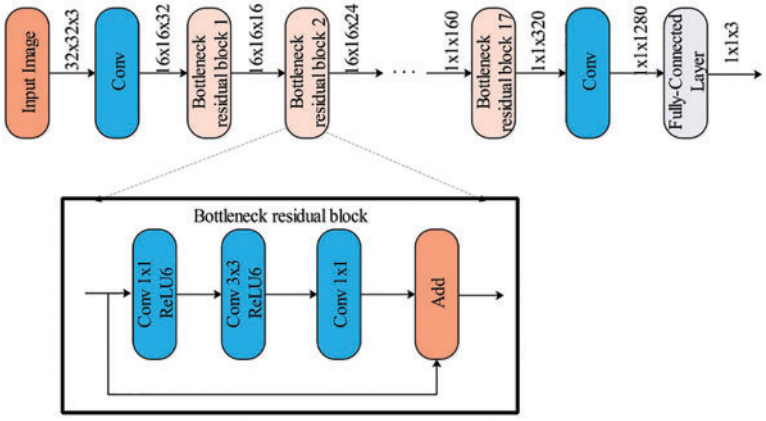

Figure 5: The main architecture of MobileNet-V2

During the training through Deep TL, fixed hyperparameters, such as a learning rate of 0.05, epochs 100, the momentum of 0.6, cross-entropy loss function, and stochastic gradient descent (SGD) set as an optimizer, have been initialized. Training is performed in two phases. In the first phase, original video frames are employed, and training is conducted. The trained model is further utilized and extracted deep features from the average pooling layer. On this layer, $N \times 1280$ features are extracted. In the second phase, cropped OF frames are employed, and performed training. Similar to the first phase, deep TL-based training is conducted, and features are extracted on the average pooling layer and obtained a vector of dimensional $N \times 1280$. A complete process of deep TL is illustrated in Fig. 6.

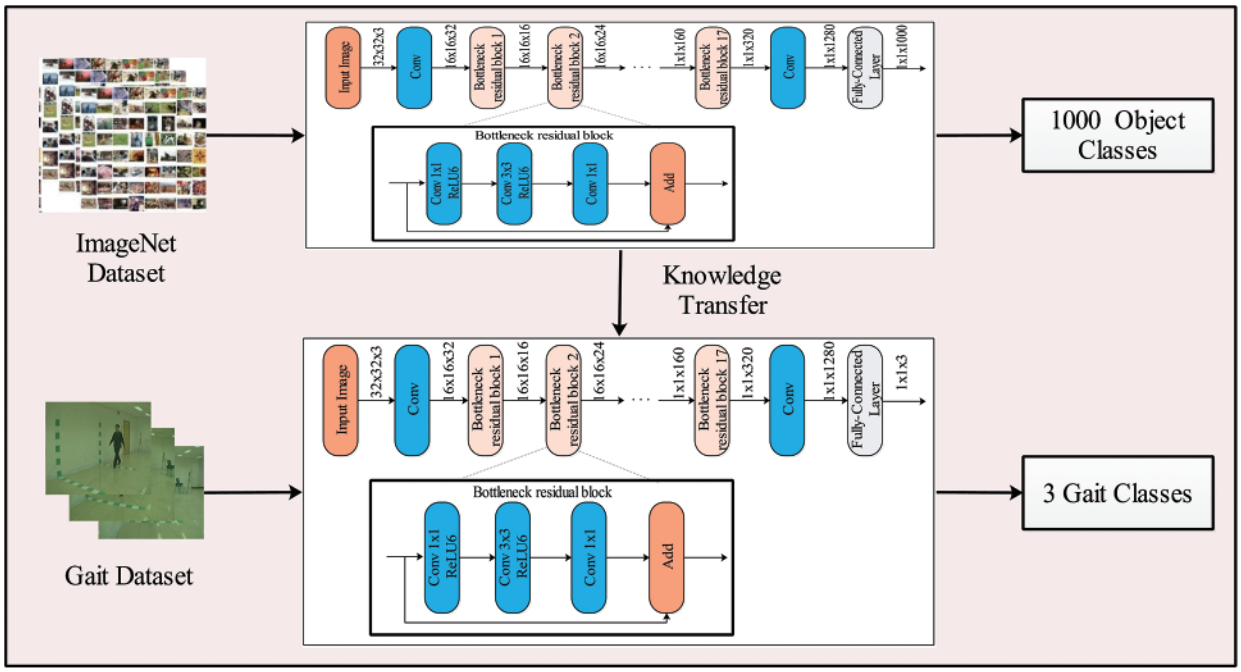

Figure 6: Process of deep transfer learning for training and features extraction on gait datasets

3.3 Features Fusion

Feature fusion is a process that improves the accuracy of an object by combing different characteristics of the same thing. This article proposed a new normal distribution along a precision-based serial approach for feature fusion. The proposed method consists of three steps: i) serial fusion of both vectors, ii) computation of normal distribution and iii) threshold-based final selection. The serial fusion is defined as follows:

$$f_1(k) = \begin{pmatrix} vec_1 \\ vec_2 \end{pmatrix}_{(N \times K_1,\, N \times K_2)} \tag{5}$$

where, vec_1 and vec_2 are originally extracted feature vectors of dimension $N \times 1280$ and $N \times 1280$, respectively. After that, a normal distribution formulation based single point is selected and put into a threshold function.

$$S(f_1(k)) = \frac{\rho}{2\pi} e^{-\rho \frac{(f_1(k)-\mu)^2}{2}} \tag{6}$$

where ρ denotes the precision and is computed as:

$$\rho = \frac{1}{\sigma^2} \tag{7}$$

$$\sigma^2 = E\left[(f_1(k) - \mu)^2\right] \tag{8}$$

Based on the value of S, a threshold is defined for the final fused vector.

$$T(f_1) = \begin{cases} \widetilde{FV} \; for \; S \geq f_1(k) \\ Ignore, \; Elsewhere \end{cases} \tag{9}$$

where, $\widetilde{FV}$ is a fused vector of dimension $N \times 1650$ that is further refined by the improved optimization algorithm.

3.4 Improved Monarch Butterfly Optimization

This article employed an improved monarch butterfly optimization (IMBO) algorithm for the best feature selection. The purpose of this algorithm is to improve accuracy and reduce computational time. Initially, the population can be classified as $Ceil(\beta \times MP)$. The monarch butterflies [48] are moved to Land 1 and Land 2 based on the following shifting procedure:

$$y_{l,m}^{i+1} = y_{b_1,m}^{i} \tag{10}$$

Here, $y_{l,m}^{i+1}$ represented the m^{th} element of y_l at $i+1$ generation and $y_{b1,m}^{i}$ represent the m^{th} element for y_{b_1}, and the current generation is denoted by i. Randomly, the butterfly b_1 is chosen from subpopulation 1. In the above equation, $b \leq \delta$, $y_{l,m}^{i+1}$ that is computed as follows:

$$b = rand \times peri \tag{11}$$

In the main MBO approach, the shifting period is presented by *peri*, and its value is set to 1.2, *rand* represents the random number between 0 and 1. If $b > \delta$, $y_{b_1,m}^{i}$, then Eq. (2) can be written as follows:

$$y_{l,m}^{i+1} = y_{b_2,m}^{i} \tag{12}$$

Here, $y_{b_1,m}^{i}$ represent the m^{th} element of y_{b_1} and b_2 *a* butterfly is selected randomly from subpopulation 2. In the next phase, the update the butterfly positions if $rand \leq p$ for each element k:

$$y_{k,m}^{i+1} = y_{best,m}^{i} \tag{13}$$

where, m^{th} element of y_k at generation $i+1$ and $y_{best,m}^{i}$ represent the m^{th} element of the butterfly that is the fittest y_{best}. If $rand > p$ then it allows for updating as:

$$y_{k,m}^{i+1} = y_{b_3,m}^{i} \tag{14}$$

Here, $y_{b_3,m}^{i}$ represent the m^{th} element for y_{b_3}, where, $b_3 \in \{1, 2, \ldots MP_2\}$.

Also, if $rand > BR$, then it can be updated as follows:

$$y_{k,m}^{i+1} = y_{k,m}^{i+1} + \beta \times (sy_m - 0.5) \tag{15}$$

Here, BR represents the adjustments of the butterfly rate. For butterfly k, the notation s_y shows the walk step, and a Levy flight can fund that.

$$s_y = Levy\left(y_k^i\right) \tag{16}$$

$$\beta = \frac{D_{max}}{i^2} \tag{17}$$

Here, β represents the weighting factor while D_{max} is the *max* walk step. In the improved algorithm, we first used a new equation to update the butterflies:

$$y_{l,new}^{i+1} = \begin{cases} y_l^{i+1}, f\left(y_l^{i+1}\right) < f\left(y_l^i\right) \\ y_l^i, \ else \end{cases} \tag{18}$$

Here, $y_{l,new}^{i+1}$ represents the individual butterfly newly created for the next generation, $f\left(y_l^i\right)$ and $f\left(y_l^{i+1}\right)$ are the fitness function for the y_l^i and y_l^{i+1} butterfly. The second improvement is made in Eq. (15) by selecting a dynamic value instead of a static value like 0.5. We computed the mean deviation of input features and updated this equation.

$$y_{k,m}^{i+1} = y_{k,m}^{i+1} + \beta \times (sy_m - MD) \tag{19}$$

The fine k-nearest neighbor (KNN) is employed as a fitness function and computes the mean error rate for each iteration. Until all iterations have been completed, this process is repeated. In this work, the initial number of iterations is 200. This algorithm returns an output feature matrix of $N \times 870$. The resultant matrix is finally passed to Bi-Layered NN for final classification.

4 Results and Discussion

The detailed experimental process of the proposed framework has been conducted in this section using visual graphs and well-defined performance measures. For the experimental process, three publicly accessible datasets, including CASIA-A, CASIA-B, and CASIA-C, have been used. Half the frames of each dataset have been employed for training the model, and the rest have been used for testing. Features are extracted, fused, selected, and passed to classification algorithms for final accuracy. All the results are computed through 10-fold cross-validation. Several classifiers have been employed for the performance comparison with Bi-Layered Neural Networks. The performance of each classifier is evaluated by accuracy and computational time (in seconds). MATLAB2022a simulates the entire framework on a personal computer with 16 GB of RAM and an 8 GB graphics card.

4.1 CASIA-A Dataset Results

The results of the CASIA-A dataset have been discussed in this sub-section. This dataset consists of angles 0, 45, and 90 degrees. For each angle, accuracy is computed separately, as shown in Table 1. This table represents that the Bi-Layered NN obtained the best accuracy for all three angles for 99.4%, 99.5%, and 99.9%. The computational time of this classifier is also recorded. For each angle, the noted time is 66.1274 (s), 71.1005 (s), and 70.5429 (s), respectively. The obtained accuracy on each angle of Bi-Layered NN is further confirmed by a confusion matrix, illustrated in Fig. 7. Moreover, this accuracy is further compared with a few other state-of-the-art techniques such as Cubic SVM, Fine-KNN, and named a few more. The Cubic SVM performed the second and obtained an accuracy of 95.8%, 96.0%, and 96.7%, respectively. Based on the results and discussion, we can claim that the performance of the proposed framework is well on the Bi-Layered NN classifier using the CASIA-A dataset.

Table 1: Gait recognition results using proposed architecture for CASIA A dataset

Classifier	Angle			Performance measure	
	0	45	90	Avg accuracy (%)	Time (s)
Bi-Layered neural network	✓			**99.4**	**66.1274**
		✓		**99.5**	**71.1005**
			✓	**99.9**	**70.5429**
Cubic SVM	✓			95.8	81.9236
		✓		96.0	80.1125
			✓	96.7	92.5006
Fine-KNN	✓			91.5	80.5639
		✓		94.2	82.9960
			✓	93.8	81.3629
Ensemble baggage tree	✓			93.5	92.5296
		✓		94.8	96.3004
			✓	96.5	95.1529
Decision tree	✓			86.5	62.5296
		✓		89.2	60.1123
			✓	91.8	59.2042

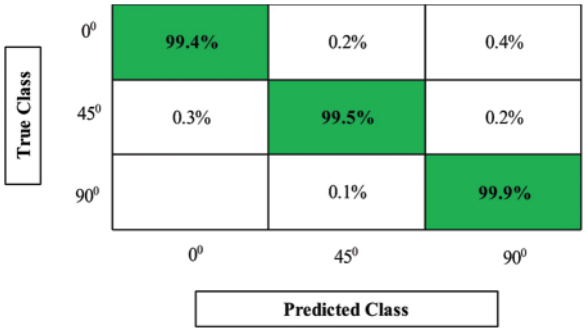

Figure 7: Confusion matrix Bi-Layered Neural Network using the proposed method for the CASIA-A dataset

4.2 CASIA B Dataset Results

The results of this dataset have been described in this section. This dataset includes three classes such as walking with a bag (BG), normal walking (NM), and walking with carrying a coat (CL). The total angles in this dataset are 11, starting from 0 and ending at 180. The difference between each angle is 18 degrees. We computed the results for each angle separately, as presented in Table 2. This table presents that the Bi-Layered NN obtained better accuracy for all angles than the other listed classifiers mentioned in this table. The accuracy of the NM class is in the range of 93.0–98.5, whereas

the average accuracy is 96.9%. The BG class accuracy range is 90.1–95.9, whereas the average accuracy is 93.6%.

Table 2: Gait recognition results using proposed architecture on the CASIA-B dataset

Classifier	Class	Angle											Mean
		0	18	36	54	72	90	108	126	144	162	180	
Bi-Layered neural network	NM	97.4	98.0	95.5	93.0	98.5	97.6	97.3	97.0	98.5	95.1	98.0	**96.9**
	BG	94.6	95.8	92.3	93.9	94.1	90.1	94.2	92.7	92.6	93.4	95.9	**93.6**
	CL	80.1	83.6	83.4	86.0	80.1	90.1	87.1	82.4	81.0	89.5	84.2	**84.3**
Cubic SVM	NM	93.5	95.2	93.4	91.2	95.2	94.3	94.1	93.2	92.9	91.6	93.9	93.5
	BG	92.1	91.4	90.0	92.5	91.4	89.7	91.8	89.6	89.2	90.1	92.3	90.9
	CL	74.8	81.0	80.6	83.1	76.3	86.1	84.7	80.5	73.3	85.5	81.6	80.7
Fine-KNN	NM	90.2	92.6	91.4	89.5	93.6	92.5	93.4	93.2	91.5	90.5	91.9	91.8
	BG	90.6	90.3	87.4	91.2	90.3	89.1	90.5	89.2	86.0	88.4	90.4	89.4
	CL	72.5	80.2	78.6	80.3	74.9	84.8	84.2	80.1	75.6	83.4	80.2	79.5
Ensemble baggage tree	NM	90.6	91.3	90.5	88.0	91.9	90.1	91.7	89.6	88.7	90.3	90.8	90.3
	BG	91.5	88.4	85.3	89.2	86.3	87.5	88.2	86.6	83.9	88.6	91.2	87.9
	CL	70.2	76.7	77.4	79.5	70.3	80.5	82.2	80.6	76.3	82.1	80.5	77.8
Decision tree	NM	89.4	91.5	90.2	89.6	90.2	90.3	90.8	88.5	89.6	90.1	87.5	89.8
	BG	90.2	88.7	86.2	88.3	88.4	89.3	87.5	87.2	84.3	90.1	90.4	88.2
	CL	71.3	74.5	76.3	78.2	71.3	81.4	82.5	81.3	77.5	80.9	77.6	77.5

Similarly, the accuracy for CL-class is also computed, and the accuracy range is 80.1–90.1, whereas the average accuracy is 84.3%. The accuracy of Cubic SVM is second best such as 93.5%, 90.9%, and 80.7%, respectively. Based on the results, we can claim that the proposed framework performed well, but still, there is much room to improve accuracy for BG and CL. The main issue is still similar to walking with a coat and carrying a bag.

4.3 CASIA-C Dataset Results

The results of this dataset have been discussed in this section. This dataset contains four classes such as quick walk (QW), normal walk (NW), slow walk (SW), and normal walk with carrying a bag (CB). The average accuracy for each class is separately computed, as presented in Table 3. In this table, it is noted that the accuracy of Bi-Layered NN is better than the other mentioned classifier in this table. For this classifier, the average accuracy of CB, SW, NW, and QW is 99.2%, 91.5%, 93.6%, and 95.8%, respectively. The computational time is also noted for each class, as mentioned in this table, as 45.1162 (s), 49.2504 (s), 41.6602 (s), and 39.1056 (s), respectively. The Cubic SVM classifier achieves the second-best accuracy for all four classes. Based on the results, we can analyze that the similarity between SW and NW is very high; therefore, the accuracy is degraded.

Moreover, the accuracy of QW class can be further improved. In addition, Fig. 8 shows the confusion matrix of Bi-Layered NN. This figure shows that the error rate of SW and NW is higher than in the other classes.

Table 3: Gait recognition results using proposed architecture on the CASIA-C dataset

Classifier	Angle				Performance measure	
	CB	SW	NW	QW	Avg accuracy (%)	Time (s)
Bi-Layered neural network	✓				**99.2**	45.1162
		✓			**91.5**	49.2504
			✓		**93.6**	41.6602
				✓	**95.8**	39.1056
Cubic SVM	✓				96.5	51.0429
		✓			87.2	53.6605
			✓		89.5	48.1040
				✓	91.2	45.2096
Fine-KNN	✓				95.2	50.4215
		✓			88.5	51.0049
			✓		86.4	46.2291
				✓	90.9	43.2510
Ensemble baggage tree	✓				93.2	63.2914
		✓			88.4	66.3910
			✓		83.5	60.5990
				✓	85.8	56.1129
Decision tree	✓				89.4	41.6629
		✓			81.8	39.0100
			✓		84.2	31.2259
				✓	82.0	27.0429

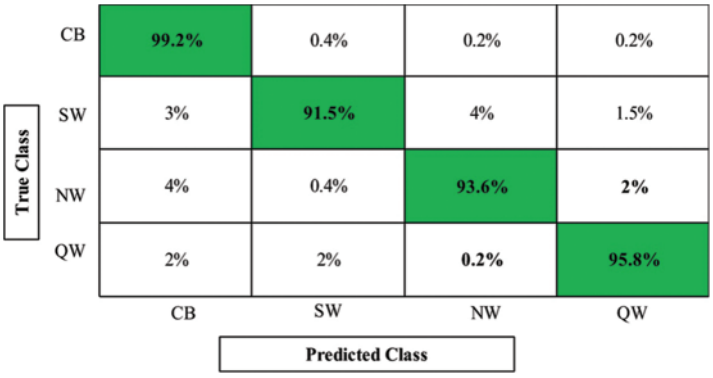

Figure 8: Confusion matrix of Bi-Layered NN using the proposed method for the CASIA-C dataset

4.4 Analysis

A short analysis is conducted at the end of the proposed framework and other possible methods; the investigation is done for all three selected datasets. Several methods have been selected and implemented on chosen datasets. The average accuracy is recorded for each method, as mentioned in Table 4. This table provides the accuracy values for CASIA A and CASIA C datasets. The implemented methods are OF-MobilenetV2, OR-MobilenetV2, OF-AlexNet, OR-AlexNet, OF-ResNet50, OR-ResNet50, OF-Densenet201, OR-Densenet201, Fusion (OF-MobilenetV2, OR-MobilenetV2), Fusion (OF-AlexNet, OR-AlexNet), Fusion (OF-ResNet50, OR-ResNet50), and Fusion (OF-Densenet201, OR-Densenet201). Compared with these methods, the proposed framework shows improved accuracy. The fusion of MobilenetV2 performed second best after the proposed framework. Similarly, Table 5 presented the analysis of the CASIA-B dataset on above listed methods. The proposed framework achieved better accuracy than other listed methods. Moreover, in comparison with the recent technique [6], it is shown that the proposed framework offers improved accuracy.

Table 4: Comparison of the proposed architecture in terms of accuracy rate with several deep learning and fusion techniques on CASIA A and CASIA C datasets. ∗ OF represents optical flow, and OR represents original images

CASIA A dataset		CASIA C dataset	
Method	Accuracy (avg)	Method	Accuracy (avg)
Proposed	**99.6**	**Proposed**	**95.02**
OF-MobilenetV2	94.2	OF-MobilenetV2	91.71
OR-MobilenetV2	96.3	OR-MobilenetV2	92.30
OF-AlexNet	93.1	OF-AlexNet	87.65
OR-AlexNet	92.6	OR-AlexNet	91.10
OF-ResNet50	94.4	OF-ResNet50	89.34
OR-ResNet50	95.7	OR-ResNet50	90.50
OF-Densenet201	97.3	OF-Densenet201	90.45
OR-Densenet201	95.8	OR-Densenet201	91.14
Fusion (OF-MobilenetV2, OR-MobilenetV2)	97.5	Fusion (OF-MobilenetV2, OR-MobilenetV2)	93.24
Fusion (OF-AlexNet, OR-AlexNet)	95.2	Fusion (OF-AlexNet, OR-AlexNet)	90.45
Fusion (OF-ResNet50, OR-ResNet50)	96.9	Fusion (OF-ResNet50, OR-ResNet50)	91.36
Fusion (OF-Densenet201, OR-Densenet201)	97.3	Fusion (OF-Densenet201, OR-Densenet201)	92.59

Table 5: Comparison of proposed architecture accuracy with several available techniques

Method	Accuracy (avg)		
	NM	BG	CL
Proposed	**96.9**	**93.6**	**84.30**
OF-MobilenetV2	91.56	88.13	77.50
OR-MobilenetV2	90.77	88.30	80.21
OF-AlexNet	86.50	84.74	74.24
OR-AlexNet	85.14	83.70	75.44
OF-ResNet50	86.95	87.90	72.57
OR-ResNet50	85.70	88.14	79.36
OF-Densenet201	89.41	85.30	76.15
OR-Densenet201	87.60	86.74	80.04
Fusion (OF-MobilenetV2, OR-MobilenetV2)	93.4	90.2	81.7
Fusion (OF-AlexNet, OR-AlexNet)	88.9	88.2	79.6
Fusion (OF-ResNet50, OR-ResNet50)	90.3	89.6	80.9
Fusion (OF-Densenet201, OR-Densenet201)	92.4	90.5	81.8

5 Conclusion

Gait recognition is a critical biometric application in which humans are identified by their walking patterns. This article proposes deep sequential learning and an improved MBO-based framework for HGR. First, the optical flow-based motion regions were extracted and cropped through dynamic coordinates. The advantage of this step is that it enhances the information for each subject under different walk patterns. Also, the raw frames were used separately to train a deep model. Then, features were extracted from the optical flow frames-based and raw frames-based trained models. Finally, the extracted features were fused using a proposed normal distribution-based approach that improved the accuracy.

Further, an improved optimization algorithm decreases the computational time during the classification process and improves accuracy. The main limitation of this framework is the reduction in some features during the fusion and selection processes. However, there exist chances that the reduced features may have some critical points that can be beneficial in improving the accuracy of the proposed framework for the CASIA-B dataset. This problem shall be considered in the future, and a more optimized approach will be designed. Moreover, the fusion technique shall be employed that neglect the redundant information.

Funding Statement: This work was supported by "Human Resources Program in Energy Technology" of the Korea Institute of Energy Technology Evaluation and Planning (KETEP), granted financial resources from the Ministry of Trade, Industry & Energy, Republic of Korea. (No. 20204010600090).

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

References

[1] C. Shen, S. Yu, J. Wang, G. Q. Huang and L. Wang, "A comprehensive survey on deep gait recognition: Algorithms, datasets and challenges," *Results in Engineering*, vol. 3, no. 2, pp. 1–21, 2022.

[2] S. Sarkar, P. J. Phillips, Z. Liu and I. R. Vega, "The humanid gait challenge problem: Data sets, performance, and analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 2, pp. 162–177, 2005.

[3] H. Li, Y. Qiu, H. Zhao, J. Zhan and R. Chen, "GaitSlice: A gait recognition model based on spatio-temporal slice features," *Pattern Recognition*, vol. 124, no. 11, pp. 108453, 2022.

[4] M. S. Nixon, T. Tan and R. Chellappa, "Human identification based on gait," *Science & Business Media*, vol. 4, no. 2, pp. 1–21, 2010.

[5] E. Vahdani and Y. Tian, "Deep learning-based action detection in untrimmed videos: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 1–11, 2022.

[6] H. Arshad, R. Damaševičius, A. Alqahtani, S. Alsubai and A. Binbusayyis, "Human gait analysis: A sequential framework of lightweight deep learning and improved moth-flame optimization algorithm," *Computational Intelligence and Neuroscience*, vol. 2022, no. 3, pp. 1–23, 2022.

[7] L. Tianyi, S. Riaz, Z. Xuande and A. Mirza, "Federated learning based nonlinear two-stage framework for full-reference image quality assessment: An application for biometric," *Image and Vision Computing*, vol. 128, no. 13, pp. 104588, 2022.

[8] M. A. Haq, "CNN based automated weed detection system using UAV imagery," *Computer System Science and Engineering*, vol. 42, no. 6, pp. 837–849, 2022.

[9] S. Qiu, H. Zhao, N. Jiang, Z. Wang and L. Liu, "Multi-sensor information fusion based on machine learning for real applications in human activity recognition: State-of-the-art and research challenges," *Information Fusion*, vol. 80, no. 6, pp. 241–265, 2022.

[10] G. Revathy, S. A. Alghamdi, S. M. Alahmari and M. A. Haq, "Sentiment analysis using machine learning: Progress in the machine intelligence for data science," *Sustainable Energy Technologies and Assessments*, vol. 53, no. 5, pp. 102557, 2022.

[11] B. Santosh Kumar, M. A. Haq, P. Sreenivasulu and D. Siva, "Fine-tuned convolutional neural network for different cardiac view classification," *The Journal of Supercomputing*, vol. 3, no. 2, pp. 1–18, 2022.

[12] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras and H. Janicke, "Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning," *IEEE Access*, vol. 10, no. 6, pp. 40281–40306, 2022.

[13] H. Chao, K. Wang, Y. He, J. Zhang and J. Feng, "GaitSet: Cross-view gait recognition through utilizing gait as a deep set," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 4, no. 2, pp. 1–8, 2021.

[14] C. Fan, Y. Peng, C. Cao, X. Liu and S. Hou, "Gaitpart: Temporal part-based model for gait recognition," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, NY, USA, pp. 14225–14233, 2020.

[15] B. Lin, S. Zhang and X. Yu, "Gait recognition via effective global-local feature representation and local temporal aggregation," in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision*, NY, USA, pp. 14648–14656, 2021.

[16] X. Li, Y. Makihara, C. Xu, Y. Yagi and M. Ren, "Joint intensity transformer network for gait recognition robust against clothing and carrying status," *IEEE Transactions on Information Forensics and Security*, vol. 14, pp. 3102–3115, 2019.

[17] Z. Lv, X. Xing, K. Wang and D. Guan, "Class energy image analysis for video sensor-based gait recognition: A review," *Sensors*, vol. 15, pp. 932–964, 2015.

[18] S. Yu, D. Tan and T. Tan, "A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition," in *18th Int. Conf. on Pattern Recognition (ICPR'06)*, NY, USA, pp. 441–444, 2006.

[19] C. Shen, B. Lin, S. Zhang and G. Q. Huang, "Gait recognition with mask-based regularization," *ArXiv Preprint*, vol. 5, no. 2, pp. 1–21, 2022.

[20] H. Arshad, M. I. Sharif, M. Yasmin and J. M. R. Tavares, "A multilevel paradigm for deep convolutional neural network features selection with an application to human gait recognition," *Expert Systems*, vol. 39, pp. e12541, 2022.
[21] Z. Zhang, L. Tran, F. Liu and X. Liu, "On learning disentangled representations for gait recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 1–9, 2020.
[22] H. Chao, Y. He, J. Zhang and J. Feng, "Gaitset: Regarding gait as a set for cross-view gait recognition," in *Proc. of the AAAI Conf. on Artificial Intelligence*, NY, USA, pp. 8126–8133, 2019.
[23] R. Liao, S. Yu, W. An and Y. Huang, "A model-based gait recognition method with body pose and human prior knowledge," *Pattern Recognition*, vol. 98, no. 4, pp. 107069, 2020.
[24] K. Shiraga, Y. Makihara, D. Muramatsu and Y. Yagi, "Geinet: View-invariant gait recognition using a convolutional neural network," in *2016 Int. Conf. on Biometrics (ICB)*, NY, USA, pp. 1–8, 2016.
[25] Z. Wu, Y. Huang, L. Wang and T. Tan, "A comprehensive study on cross-view gait based human identification with deep CNNs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 209–226, 2016.
[26] C. Song, Y. Huang, Y. Huang, N. Jia and L. Wang, "GaitNet: An end-to-end network for gait based human identification," *Pattern Recognition*, vol. 96, pp. 106988, 2019.
[27] K. H. Abdulkareem, N. Arbaiy, Z. H. Arif and S. Kadry, "Mapping and deep analysis of image dehazing: Coherent taxonomy, datasets, open challenges, motivations, and recommendations," *International Journal of Interactive Multimedia & Artificial Intelligence*, vol. 7, pp. 1–21, 2021.
[28] A. Mujahid, M. J. Awan, A. Yasin and R. Damaševičius, "Real-time hand gesture recognition based on deep learning YOLOv3 model," *Applied Sciences*, vol. 11, pp. 4164, 2021.
[29] M. N. Akbar, F. Riaz, A. B. Awan and S. Rehman, "A hybrid duo-deep learning and best features based framework for action recognition," *Computers, Materials & Continua*, vol. 73, no. 6, pp. 2555–2576, 2022.
[30] A. Mehmood, M. Sharif, S. A. Khan and M. Shaheen, "Prosperous human gait recognition: An end-to-end system based on pre-trained CNN features selection," *Multimedia Tools and Applications*, vol. 23, pp. 1–21, 2020.
[31] M. Sharif, M. Yasmim, T. Saba and U. J. Tanik, "A machine learning method with threshold based parallel feature fusion and feature selection for automated gait recognition," *Journal of Organizational and End User Computing (JOEUC)*, vol. 32, pp. 67–92, 2020.
[32] M. P. Rani and G. Arumugam, "An efficient gait recognition system for human identification using modified ICA," *International Journal of Computer Science and Information Technology*, vol. 2, no. 4, pp. 55–67, 2010.
[33] M. Deng, Y. Sun, Z. Fan and X. Feng, "Human gait recognition by fusing global and local image entropy features with neural networks," *Journal of Electronic Imaging*, vol. 31, pp. 013034, 2022.
[34] R. Anusha and C. Jaidhar, "Human gait recognition based on histogram of oriented gradients and haralick texture descriptor," *Multimedia Tools and Applications*, vol. 79, pp. 8213–8234, 2020.
[35] M. I. Sharif, M. Nazir, S. Alsubai and A. Binbusayyis, "Deep learning and kurtosis-controlled, entropy-based framework for human gait recognition using video sequences," *Electronics*, vol. 11, pp. 334, 2022.
[36] A. Mehmood, U. Tariq, C. Jeong, Y. Nam and R. Mostafa, "Human gait recognition: A deep learning and best feature selection framework," *Computers*, vol. 70, pp. 343–360, 2022.
[37] A. Khan, M. Y. Javed, M. Alhaisoni, U. Tariq and S. Kadry, "Human gait recognition using deep learning and improved ant colony optimization," *Computer, Material and Continua*, vol. 71, no. 2, pp. 300–320, 2022.
[38] A. Zhao, J. Li and M. Ahmed, "SpiderNet: A spiderweb graph neural network for multi-view gait recognition," *Knowledge-Based Systems*, vol. 206, pp. 106273, 2020.
[39] X. Wang and W. Q. Yan, "Human gait recognition based on frame-by-frame gait energy images and convolutional long short-term memory," *International Journal of Neural Systems*, vol. 30, pp. 1950027, 2020.
[40] H. Arshad, M. I. Sharif, M. Yasmin, J. M. R. Tavares and Y. D. Zhang, "A multilevel paradigm for deep convolutional neural network features selection with an application to human gait recognition," *Expert Systems*, vol. 31, pp. e12541, 2020.

[41] M. Derlatka and M. Borowska, "Ensemble of heterogeneous base classifiers for human gait recognition," *Sensors*, vol. 23, no. 5, pp. 508, 2023.
[42] Y. Wang, Q. Song, T. Ma, N. Yao and R. Liu, "Research on human gait phase recognition algorithm based on multi-source information fusion," *Electronics*, vol. 12, pp. 193, 2023.
[43] Y. V. Altamirano-Flores, I. H. Lopez-Nava and I. González, "Emotion recognition from human gait using machine learning algorithms," in *Int. Conf. on Ubiquitous Computing and Ambient Intelligence*, NY, USA, pp. 77–88, 2022.
[44] J. L. Barron, D. J. Fleet and S. S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, pp. 43–77, 1994.
[45] M. Sandler, A. Howard, M. Zhu and L. C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, NY, USA, pp. 4510–4520, 2018.
[46] M. Toğaçar, B. Ergen and Z. Cömert, "COVID-19 detection using deep learning models to exploit social mimic optimization and structured chest X-ray images using fuzzy color and stacking approaches," *Computers in Biology and Medicine*, vol. 121, pp. 103805, 2020.
[47] M. Akay, Y. Du, C. L. Sershen and S. Assassi, "Deep learning classification of systemic sclerosis skin using the MobileNetV2 model," *IEEE Open Journal of Engineering in Medicine and Biology*, vol. 2, no. 4, pp. 104–110, 2021.
[48] G. G. Wang, S. Deb and Z. Cui, "Monarch butterfly optimization," *Neural Computing and Applications*, vol. 31, no. 5, pp. 1995–2014, 2019.