Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 8238375, 13 pages
https://doi.org/10.1155/2022/8238375

Research Article

Human Gait Analysis: A Sequential Framework of Lightweight Deep Learning and Improved Moth-Flame Optimization Algorithm

Muhammad Attique Khan,[1] **Habiba Arshad**,[2] **Robertas Damaševičius**,[3] **Abdullah Alqahtani**,[4] **Shtwai Alsubai**,[4] **Adel Binbusayyis**,[4] **Yunyoung Nam**,[5] **and Byeong-Gwon Kang**[5]

[1]*Department of Computer Science, HITEC University, Taxila, Pakistan*
[2]*Department of Computer Science, University of Wah, Wah Cantt, Pakistan*
[3]*Department of Software Engineering, Kaunas University of Technology, Kaunas, Lithuania*
[4]*College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj, Saudi Arabia*
[5]*Department of ICT Convergence, Soonchunhyang University, Asan 31538, Republic of Korea*

Correspondence should be addressed to Byeong-Gwon Kang; bgkang@sch.ac.kr

Received 16 April 2022; Revised 23 May 2022; Accepted 3 June 2022; Published 14 July 2022

Academic Editor: Muhammad Ahmad

Human gait recognition has emerged as a branch of biometric identification in the last decade, focusing on individuals based on several characteristics such as movement, time, and clothing. It is also great for video surveillance applications. The main issue with these techniques is the loss of accuracy and time caused by traditional feature extraction and classification. With advances in deep learning for a variety of applications, particularly video surveillance and biometrics, we proposed a lightweight deep learning method for human gait recognition in this work. The proposed method includes sequential steps–pretrained deep models selection of features classification. Two lightweight pretrained models are initially considered and fine-tuned in terms of additional layers and freezing some middle layers. Following that, models were trained using deep transfer learning, and features were engineered on fully connected and average pooling layers. The fusion is performed using discriminant correlation analysis, which is then optimized using an improved moth-flame optimization algorithm. For final classification, the final optimum features are classified using an extreme learning machine (ELM). The experiments were carried out on two publicly available datasets, CASIA B and TUM GAID, and yielded an average accuracy of 91.20 and 98.60%, respectively. When compared to recent state-of-the-art techniques, the proposed method is found to be more accurate.

1. Introduction

Person recognition and identification using gait have great importance in the field of machine learning and computer vision [1]. Gait is the walking behavior of a person but to recognize a person by gait from distance and in less illuminated environment it becomes very complicated and difficult [2]. Moreover, as compared to other traditional biometric techniques such as fingerprint, face detection, and iris detection, it does not require direct contact of a person [3]. Due to these discriminative factors, it has taken a lot of attention from researchers and it is used to apply in various applications like security surveillance, dubious person detection, and forensics [4, 5].

In early research, gait recognition was categorized in to two main categories such as model-based and appearance-based [6]. The prior categories are more costly to implement the human model using high-resolution videos and give low average results as compared to modern categories; hence, researchers focus on using modern categories for gait feature detection [7]. In the model-based method, prior information is used to extract the moving features of the human body [8]. Furthermore, in this method, the movement of human body is examined using changing factors like gait path, position of joints, and the torso [9]. This is a challenging method, due to its high computational complications. In the model-free method, gait cycle is used to extract the features from a silhouette, and it is simple to implement due to less computational cost [10].

There are various machine learning and computer vision techniques that are used to overcome the covariate factors like angular deviations [11], lightning conditions [12], and clothing and carrying bags [6, 13], but there exist various challenges in extracting useful features that affect the optimal accuracy results. Feature extraction is considered as the most important step in recognizing gait traits [14], such as if the extracted features are related to the problem, then the system will be able to correctly recognize the human gait patterns. In contrast, if irrelevant features are evaluated, then the system performance will go down and it will not give optimal recognition results [10]. In past, various types of features are used like shape-based features [15], geometrical features [16], and statistical features [17]. Deep features are extracted using deep convolutional neural network techniques to overcome these challenges. Deep learning techniques, rather than manual feature extraction, extract automated features from raw images [18, 19]. In this work, we proposed a sequential lightweight deep learning architecture for human gait recognition. Our major contributions are listed as follows:

(i) Two pretrained deep learning models are modified namely VGG-19 and MobileNet-V2 based on the target dataset classes and adjusted their weights. Then, both models are trained using transfer learning without freezing any layer and obtained newly trained models.

(ii) Feature engineering is performed on fully connected layer 7 (VGG-19) and global pooling layer (MobileNet-V2) and fused by employing discriminant correlation analysis (DCA)

(iii) A modified moth-flame optimization algorithm is developed for the selection of optimum features that are finally classified using extreme learning machine (ELM)

The rest of the article is organized as follows: Section 2 describes the manuscript's related work. Section 3 discusses the specifics of selected datasets. The proposed methodology is presented in Section 4. Section 5 discusses and explains the experimental results. Finally, Section 6 brings the entire manuscript that followed the references to a close.

2. Related Works

Identification of human through gait is the most biometric application, and researchers have made extensive studies for it by extracting feature values [20]. In literature, various machine learning and computer vision-based techniques are implemented for human gait recognition [21]. Liao et al. [22] presented Pose-gait model-based gait recognition method. In this approach, the human 3D pose is estimated using CNN, and then spatiotemporal features extracted from the 3D pose are used for the improvement in recognition. Two publicly available datasets CASIA B and CASIA E are used for experimentation, and it gives auspicious results in the presence of covariate factors. Sanjay et al. [23] introduced an automated approach for human gait recognition in the presence of a covariate situation. In the first step, basic distinct stances in the gait cycle are detected which are used to compute the gait features related to these detected stances and it is termed as dynamic gait energy image (DGEI). Then generative adversarial network (GAN) is used to detect the corresponding dynamic gait energy image. These extracted features are then used to compare with the gallery sequences. Finally, GAN-created DGEI is used for final recognition. Publicly available datasets such as CASIA B, TUM Gait, and OU-ISIR TreadMill B are used to validate the presented approach, and it gives considerably improved results as compared to existing methods. Chen et al. [24] introduced a method for cross-view gait recognition using deep learning. Multiview gait generative adversarial network is introduced for creating fake gait data samples for extension in existing data. The method is then used to train each instance of each view involved in single or multiple datasets. Domain alignment using projected maximum mean dependency (PMMD) is utilized to minimize the effect of distribution divergence. CASIA B and OUMVLP are used for experimentation, and the achieved results show that the introduced method gives better results than existing methods. Hou et al. [25] presented a set residual network-based gait recognition model to detect more discriminative features from the silhouettes. Set residual block is introduced to extract the silhouette level and two-level features in a parallel sequence, and then the residual connection is applied to join the two-level features. Moreover, an efficient method is applied to utilize features from the deep layers. Two datasets CASIA B and OUMVLP are used for experimentation. The applied approach gives consistent results as compared to existing methods. Gul et al. [13] introduced a machine vision method to extract the distinct gait features from covariate factors. Spatiotemporal gait features are extracted, and these features are used to train the 3D CNN model to overcome these challenges. Then, the holistic method is used by the model to implement the distinct gait features in the form of gait energy images. Two publicly available datasets OULP and CASIA B are used to test the validity of the introduced method with large gender and age differences. The presented approach gives promising results using CASIA B dataset as compared to existing methods. The methods presented above concentrated on both spatial and temporal data. None of them focused on feature fusion or optimization of extracted features to achieve better results in the shortest amount of time. As a result, in this article, we proposed a lightweight deep learning framework for human gait recognition that not only improves accuracy but also reduces a system's computational time.

3. Datasets

3.1. TUM GAID Dataset. TUM Gait from Audio, Image and Depth (GAID) [26] dataset consists of RGB audio, video, and depth. It consists of 305 subjects carried out in two indoor walking sequences in which four distinct situations are captured without any view variation through Microsoft Kinect like six normal walk videos ($n1$–$n6$), two videos with carrying a bag ($b1$–$b2$), and two walking videos wearing coating shoes ($s1$-$s2$), and there is an elapsed time instance in which 32 subjects were recorded wearing distinct cloths. A few sample images are shown in Figure 1.

3.2. CASIA B Dataset. CASIA B [27] is a multiview and indoor gait dataset in which 124 subjects are included in recording session of which 93 are male participants and 31 female participants. This dataset considers major factors for gait recognition, that is variation in view angle, clothing, and carrying situations separately. For all, subject videos are captured through a USB camera from 11 different views that include six normal walking videos (NM), two walking videos with wearing a coat (CL), and two walking videos with carrying a bag (BG). A few sample images are shown in Figure 2.

4. Methodology

The proposed lightweight (LW) human gait recognition framework has been presented in this section with detailed mathematical formulations and flow diagrams. The main architecture is shown in Figure 3. In this figure, it is illustrated that the proposed method consists of some important steps such as modification of pretrained CNN models, training of modified models using transfer learning (TL), feature engineering on global average pooling layers, fusion of extracted deep features using discriminant correlation analysis (DCA), selection of best features using the modified moth-flame optimization algorithm, and finally classification using the extreme learning machine (ELM). The details of each step are given.

4.1. Convolutional Neural Network (CNN). The convolutional neural network (CNN) has become an important recognition task in the domain of computer vision. The CNN architecture is employed for feature extraction of an image based on several hidden layers. CNNs have many layers, including convolutional, pooling, fully connected, and others. The convolution layer (CL) is the most important layer of a CNN that performed a 2D convolution process on the input and the kernels through a forward pass. The kernel weights in every CL are assigned randomly and their values are changed at each step by applying the loss function through network training. In the end, the resultant learned kernels may identify some types of shapes within the input images. In CL, three different types of steps are performed like convolution, stack, and nonlinear activation function.

Suppose, we have an input matrix M and an output Z of the CL, and there are some set of kernels $K_l, \forall l \in [1, \ldots, L]$, then the output of the convolution process $O(l)$ after step 1 is represented as

$$O(l) = M \otimes K_l, \quad \forall l \in [1, \ldots, L], \tag{1}$$

where $\otimes$ refers to the convolution process, which is the product of filter and inputs. Second, all $O(l)$ activation maps are combined to create a novel 3D activation map.

$$R = Q(O(1), \ldots, O(L)), \tag{2}$$

where Q represents the combination of operations with channel direction, and L is the total number of filters. Third, the 3D activation map R is given as input into the activation function and gives the resultant activation map as

$$Z = \mathrm{NLAF}(R). \tag{3}$$

The size Q of three main matrices (input, filters, and result) is taken as

$$Q(i) = \begin{cases} T_A \times U_A \times V_A, & i = I, \\ T_P \times U_P \times V_P, & i = K_l, \forall l \in [1, \ldots, L], \\ T_N \times U_N \times V_N, & i = Z, \end{cases} \tag{4}$$

where the variables (T, U, V) represent the size of height, width, and channels of the activation map, and the subscripts A, P, and N represent input, filter, and output, respectively. It contains two equalities. First, $V_A = V_P$ refers the channel of input V_A equals to the channels of filter V_P. Second, $V_N = L$ refers the channels of output V_N equals the number of filters L. Suppose Y represents padding, S represents stride, so the result of T_N, U_N, V_N can be evaluated as

$$\begin{aligned} T_N &= 1 + d_{de}\left[\frac{(2 \times Y + T_A - T_P)}{S}\right], \\ U_N &= 1 + d_{de}\left[\frac{(2 \times Y + U_A - U_P)}{S}\right], \end{aligned} \tag{5}$$

where d_{de} is the floor function. The nonlinear activation Υ generally selects the rectified linear unit (ReLU) function [28].

$$\begin{aligned} \gamma_{\mathrm{ReLU}}(r_{mn}) &= \mathrm{ReLU}(r_{mn}) \\ &= \max(0, r_{mn}), \end{aligned} \tag{6}$$

where $r_{mn} \in R$ is the component of the activation map R. At present, ReLU is the mostly used NLAF as compared to the traditional hyperbolic tangent (HT) and sigmoid function (SM) function, that are computed as

$$\begin{aligned} \gamma_{\mathrm{HT}}(r_{mn}) &= \tanh(r_{mn}) = \frac{(f^{r_{mn}} - f^{-r_{mn}})}{(f^{r_{mn}} + f^{-r_{mn}})}, \\ \gamma_{\mathrm{SM}}(r_{mn}) &= (1 + f^{-r_{mn}})^{-1}. \end{aligned} \tag{7}$$

4.2. Transfer Learning. Transfer learning (TL) is the branch of machine learning that transfer the knowledge of one domain to a different domain within less computational time. Given a

FIGURE 1: Example images from the TUM GAID gait dataset [26].

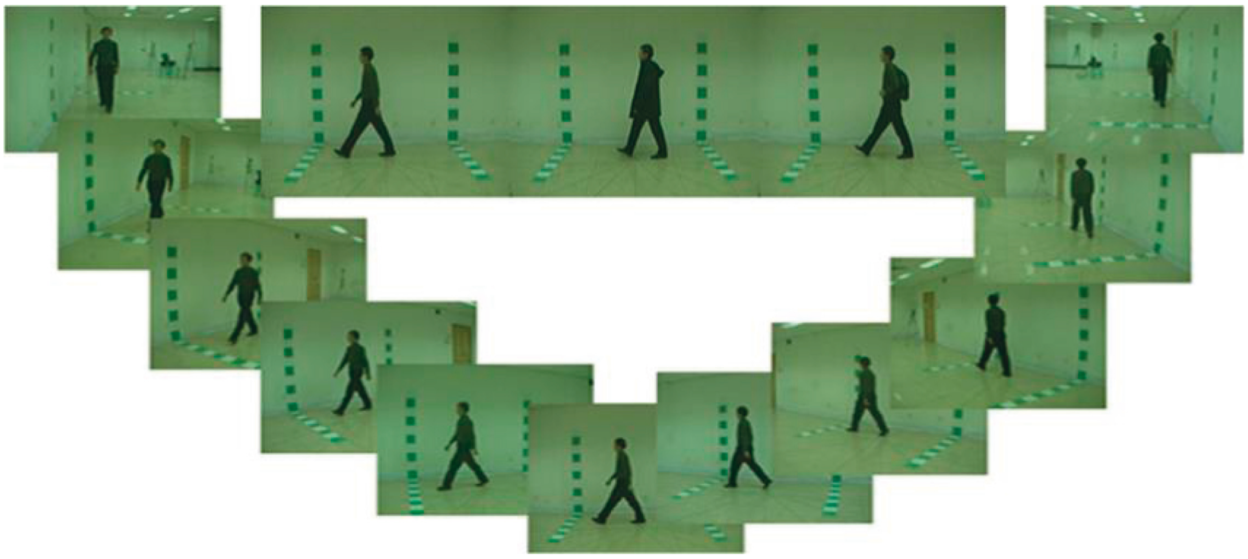
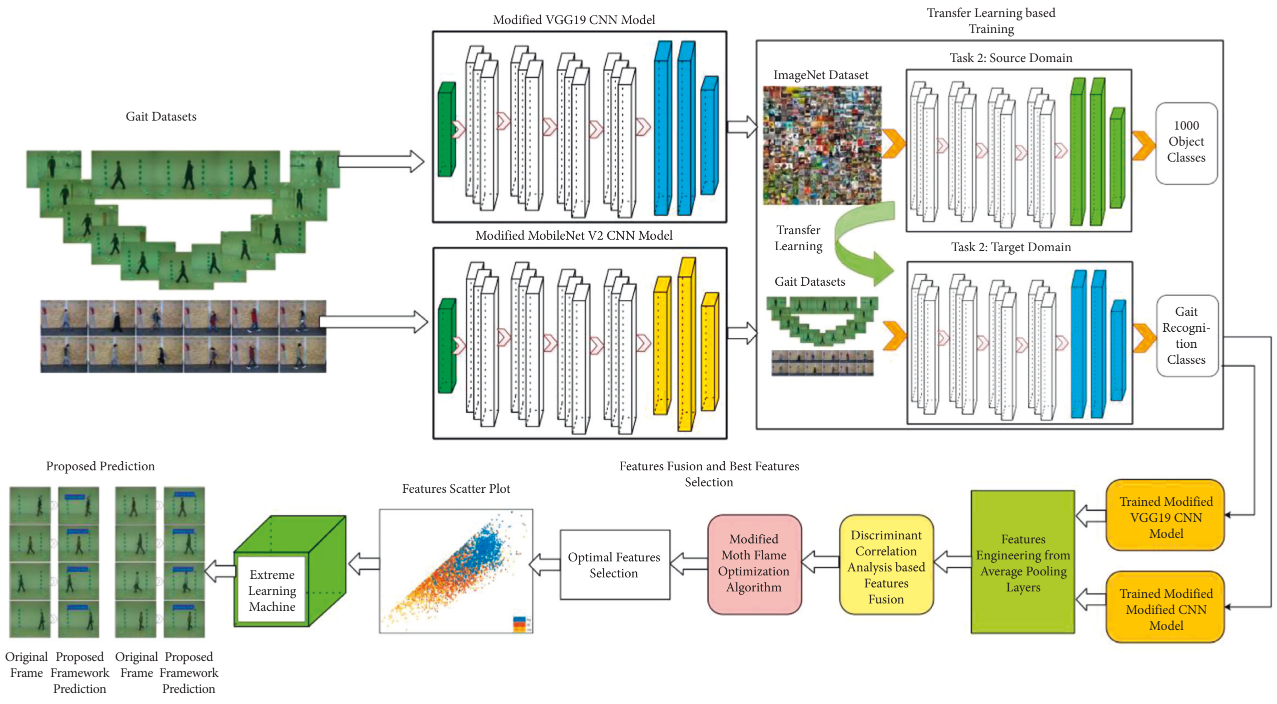


FIGURE 2: Sample images of the CASIA B gait dataset [27].


Gait Datasets
Modified VGG19 CNN Model
Modified MobileNet V2 CNN Model
Transfer Learning based Training
Task 2: Source Domain
ImageNet Dataset
1000 Object Classes
Transfer Learning
Task 2: Target Domain
Gait Datasets
Gait Recogni-tion Classes
Trained Modified VGG19 CNN Model
Trained Modified Modified CNN Model
Features Engineering from Average Pooling Layers
Discriminant Correlation Analysis based Features Fusion
Modified Moth Flame Optimization Algorithm
Features Fusion and Best Features Selection
Optimal Features Selection
Features Scatter Plot
Extreme Learning Machine
Proposed Prediction
Original Frame
Proposed Framework Prediction
Original Frame
Proposed Framework Prediction
FIGURE 3: Proposed main flow of human gait recognition using lightweight deep learning architecture.

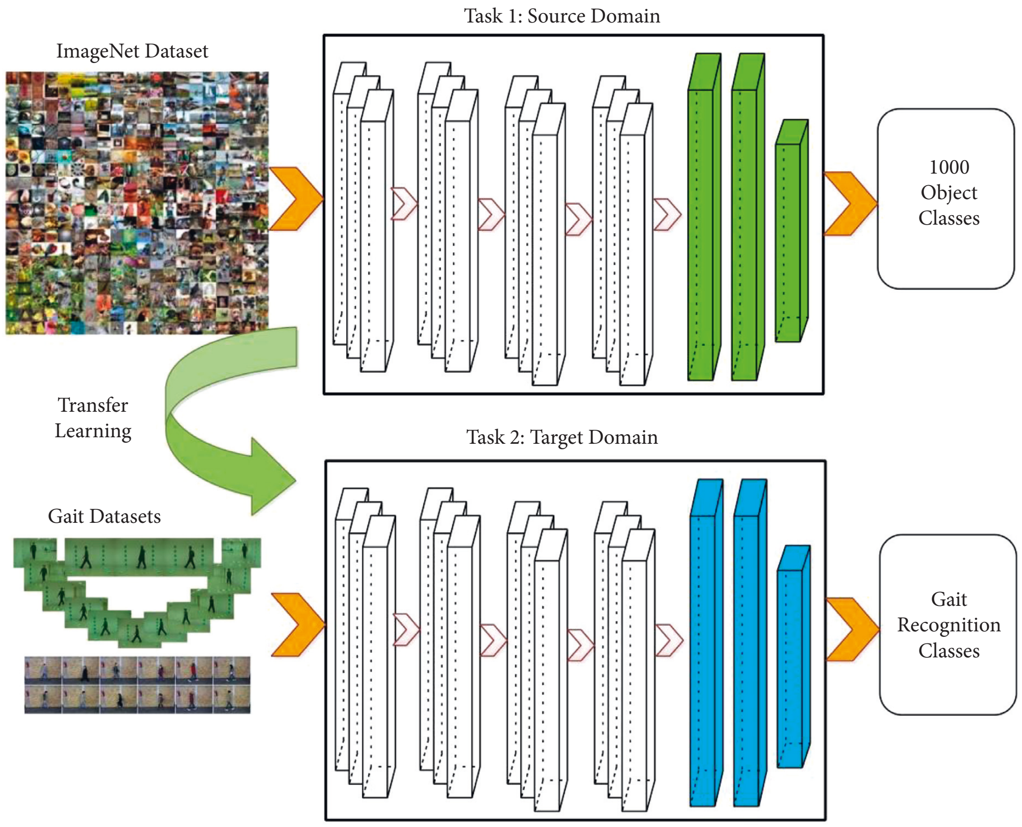


FIGURE 4: Transfer learning-based training of modified CNN models for human gait recognition.

source domain $\xi^S = \{F^S, P^S(\phi^S)\}$ with the source task $\lambda^S = \{\widetilde{F}^S, f^S(\cdot)\}$ and a target domain $\xi^T = \{F^T, P^T(\phi^T)\}$ with the target task, $\lambda^T = \{\widetilde{F}^T, f^T(.)\}$ aims to learn a better mapping function $f^T(.)$ for the target task λ^T for the knowledge transfer from the source domain ξ^S and task λ^S. Hence, the TL is expressed as follows:

$$\xi^S \neq \xi^T \text{ or } \lambda^S \neq \lambda^T, \text{ where } \phi = v|v_i \in F, \quad i = 1, 2, 3, \ldots N. \tag{8}$$

Hence, DTL is defined as follows: given a TL task $f^{S\longrightarrow T}(.)$: $F^T \longrightarrow \widetilde{F}^T$ based on $[\xi^S, \xi^T, \lambda^S, \lambda^T]$, DTL objectives to acquire the $f^{S\longrightarrow T}(.)$ by leveraging powerful DL process. Visually, the DTL process is shown in Figure 4.

4.3. Modified VGG-19 Model. Visual geometry group (VGG)-19 [29] is the modified version of VGG, and it consists of 19 layers with 16 convolutional layers of 64, 128, and 256, and 512 filter sizes with stride length and padding is 1 pixel on each side. The convolutional layer consists of 5 sets, two of them contain 64 filters, the next set contains 2 convolutional layers with 128 filters, the next set contains 4 convolutional layers with 256 filters, and the last two sets contain 4 convolutional layers with 512 filters each. Then, a max pooling layer with a 2×2 filter size and stride rate of 2 pixels are used after each set of convolutional layers. The output is then passed to the fully connected layer. Three fully connected (FC) layers and one softmax layer are used for the classification. In this work, we removed the last fully connected (FC) layer and added a new FC layer. Then, several hyperparameters are employed such as training rate, epochs, mini batch size, optimizer, and training loss. Based on these hyperparameters, we trained the modified model from scratch through TL and obtained a new model for only gait recognition task. Later, this modified model is used for the feature engineering task.

4.4. Modified MobileNet-V2 Model. MobileNet-V2 [30] is a lightweight CNN-based model specially designed for mobile devices. This architecture can perform well on small datasets as it can overcome the effect of overfitting and it also optimized the memory consumption. In this network, 17 inverted residuals are used between two convolutional layers and one FC layer. So, the depth of the network consists of 53 convolutional layers and one FC layer. The working of this architecture is based on two concepts that include depth-wise separable convolution and the inverted residual methods. In this architecture, a full convolutional layer is replaced with a factorized version that divides the convolution into two separate groups. The first layer is named as depth-wise convolution; its function is to do lightweight filtering by using one convolutional filter on each input channel. The second layer is named as point-wise convolutional layer, which is used for creating new features by computing linear models of the input channels. The depthwise convolutional layers contains 2 convolutional layers: the first layer contains a 3×3 filter size, while the other one has a 1×1 filter size. The other two regular convolutional layers have filter sizes of 3×3 and 1×1. Moreover, in this architecture, ReLU6 is used instead of ReLU as it is more efficient for less accurate computation. Dropout and batch normalization are then applied, where layer activators are standardized to mean zero and unit variance, and then, linear transformation is applied [31].

In this work, we removed the last layer and added a new FC layer. Then, several hyperparameters are employed such as training rate (0.005), epochs (100), mini batch size (32), optimizer (Adam), and training loss. Based on these hyperparameters, we trained the modified model from scratch through TL and obtained a new model for only gait recognition task.

4.5. Feature Engineering. Feature engineering is applied on global average pooling layers of both models and obtained two feature vectors of dimension $\times K_1$ and $N\times K_2$. Mathematically, this process is defined as follows:

$$\begin{aligned} K_1 &= \text{activation}\,(\text{layer}),\\ K_2 &= \text{activation}\,(\text{layer}), \end{aligned} \tag{9}$$

where K_1 and K_2 represent the length of feature vectors, and layer defines the selected one like global average pooling. Thereafter, the fusion process is performed using discriminant canonical correlation analysis approach.

4.6. Discriminative Canonical Correlation Analysis-Based Fusion. In this work, the DCCA fusion approach is employed for feature fusion. By applying canonical correlation analysis (CCA), the correlated features $l_m^v m_t$ and $l_n^v n_t$, $t=1,\ldots,z$, are extracted and merged for identification. Though the features extracted from related class samples are not utilized, resultantly it becomes the constraint of the recognition capabilities of CCA. Moreover, the basic concept to introduce CCA is for modeling instead of recognition, and correlation β refers to the certainty among $l_m^v m_t$ and $l_n^v n_t$, $t=1,\ldots,z$. CCA was more often utilized for modeling and estimation, for instance image extraction and parameter prediction. If the extracted features are for recognition, then the class description of the instances should be utilized to get more discriminatory features. Finally, the class description was fused with the CCA framework for cooperated feature extraction and presented an innovative approach of fused feature extraction for multimodal recognition, termed as discriminative canonical correlation analysis (DCCA). Mathematically, this approach is defined as follows.

Suppose z pairs of mean-normalized pairwise instances $\{(m_t,n_t)\}_{t=1}^{z}\in\xi^a\times\xi^b$ access from p classes, DCCA can be systematically represented in the below optimization problem:

$$\max_{l_m,l_n}\left(l_m^v S_l l_n-\mu . l_m^v S_h l_n\right), \tag{10}$$

$$\begin{aligned} \text{s.t.}\, l_m^v MM^v l_m &= 1,\\ l_n^v NN^v l_n &= 1, \end{aligned} \tag{11}$$

where the parameters S_l and S_h are used to compute the inside-class association and between-class association, respectively (detailed description is given below), $\mu>0$ adjustable metric that shows the comparative significance of the inside-class association $l_m^v S_l l_n$ contrasted with the between-class association $l_m^v S_h l_n$. Moreover, the limitation value represents the scale limitation on l_m, l_n.

$$\begin{aligned} M &= \left[m_1^{(1)},\ldots,m_{z_1}^{(1)},\ldots,m_1^{(p)},\ldots,m_{z_c}^{(p)}\right],\\ N &= \left[n_1^{(1)},\ldots,n_{z_1}^{(1)},\ldots,n_1^{(p)},\ldots,n_{z_c}^{(p)}\right],\\ f_{z_t} &= \left[\underbrace{0,\ldots 0,}_{\sum} k^{t-1} z_k \underbrace{1,\ldots 1,}_{z_t} \underbrace{0,\ldots 0}_{z-\sum_{k\le 1}^{t-1} z_k}\right]^v \in Q^z,\\ 1_z &\le [1,\ldots 1]^V \in Q^z, \end{aligned} \tag{12}$$

where $m_k^{(t)}$ refers the kth instance in the tth class, similarly $n_k^{(t)}$, and z_t shows the number of instances of $m_k^{(t)}$ or $n_k^{(t)}$ in the tth class. The matrix S_l is represented as

$$\begin{aligned} S_l &= \sum_{t=1}^{p}\sum_{d=1}^{z_t}\sum_{y=1}^{z_t} m_d^{(t)} n_y^{(t)V}\\ &= \sum_{t=1}^{p}\left(Mf_{z_t}\right)\left(Nf_{z_t}\right)\\ &= MGN^V, \end{aligned} \tag{13}$$

$$G=\begin{bmatrix} 1_{z_1\times z_1} & & & & \\ & \ddots & & & \\ & & 1_{z_t\times z_t} & & \\ & & & \ddots & \\ & & & & 1_{z_p\times z_p} \end{bmatrix}\in\xi^{z\times z}, \tag{14}$$

where G represents a proportionate, positively defined, block crosswise matrix, and Matrix $(G)=p$. In contrast, the matrix S_h is represented as

$$\begin{aligned} S_h &= \sum_{t=1}^{p}\sum_{\substack{k=1\\ k\neq t}}^{p}\sum_{d=1}^{z_t}\sum_{y=1}^{z_k} m_d^{(t)} n_y^{(t)V}\\ &= \sum_{t=1}^{p}\sum_{k=1}^{p}\sum_{d=1}^{z_t}\sum_{y=1}^{z_k} m_d^{(t)} n_y^{(t)V}-\sum_{t=1}^{p}\sum_{k=1}^{z_t}\sum_{d=1}^{z_t} m_d^{(t)} n_y^{(t)V}\\ &= \left(M1_z\right)\left(N1_z\right)^V-\text{MGN}^V\\ &= -\text{MGN}^V. \end{aligned} \tag{15}$$

The "=" in the end holds because mean normalization is applied on the instances, hence both $M1_z = 0$ and $N1_z = 0$ holds. In contrast between equations (13) and (15), the only difference among S_l and S_h is a single negative sign, thus the objective of equation (10) will be $(1+\mu)l_m^v S_l l_n$, and this enhancement issue is free from parameter μ, consequently μ can be excluded. Hence, DCCA can be represented as

$$\begin{gathered}\max_{l_m,l_n}\left(l_m^v \mathrm{MGN}^V l_n\right),\\ \text{s.t.}\ l_m^v MM^v l_m = 1, l_n^v NN^v l_n = 1.\end{gathered} \tag{16}$$

By applying the Lagrangian multiplier technique, it becomes very simple to get main equation of DCCA, which is represented as

$$\begin{pmatrix} - & \mathrm{MGN}^V \\ \mathrm{NGM}^v & - \end{pmatrix}\begin{pmatrix} l_m \\ l_n \end{pmatrix} = \vartheta\begin{pmatrix} MM^V & \\ & YY^V \end{pmatrix}\begin{pmatrix} l_m \\ l_n \end{pmatrix}. \tag{17}$$

When the vector pairs (l_{mt}, l_{nt}), $i = 1, \ldots, g$, adjacent to the first g largest generalized eigenvalues and attained, let $L_m = [l_{m1}, \ldots, l_{mg}]$, $L_n = [l_{n1}, \ldots, l_{ng}]$, then both feature extraction and the feature fusion can be performed using FFS-I and II, respectively, where g fulfills the limitations $g \le \min(a, b)$ and $g \le p$. The formulation returned a fused vector of dimension $N \times K_3$, where $K_3 \in (K_1, K_2)$. Later on, this resultant vector is further improved using a modified moth-flame optimization algorithm.

4.7. Moth-Flame Optimization Algorithm (MFO). Several nature-inspired optimization algorithms have been introduced in the literature for best feature selection such as genetic algorithm, particle swarm optimization, and moth-flame optimization [32]. The improved moth-flame optimization algorithm is utilized in this work for the best feature selection. Originally, the MFO algorithm was presented by Mirjalili [33]. It is under the populace-based metaheuristics algorithm. In this procedure, first the data flow of MFO begins by randomly generating moths within the resultant space. Then it calculates the positional (i.e., fitness) value of each moth and label the best position by flame. Afterwards, changing the moth place depends on a whole movement function used to attain a better position labeled by a flame. Moreover, it updates the new best positions of the individual. The previous process (i.e., updating of moths' location and generating the new location) until it meets the resultant criteria. The MFO algorithm consists of three major steps that are as follows.

4.7.1. Creating the Initial Population of Moths. As stated in [33], it is supposed that an individual moth can fly in 1D, 2D, 3D, or in hyper-dimensional position. The matrix of moths can be represented as

$$H = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & \cdots & h_{1,a} \\ h_{2,1} & h_{2,2} & \cdots & \cdots & h_{2,a} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{m,1} & h_{m,1} & \cdots & \cdots & h_{m,a} \end{bmatrix}, \tag{18}$$

where m represents the number of moths' and a represents the number of dimensions in the resultant region. Moreover, the fitness values for entire moths' stored in an array are represented as

$$VH = \begin{matrix} VH_1 \\ VH_2 \\ \vdots \\ VH_m \end{matrix}. \tag{19}$$

The remaining elements in the algorithm are flames that are represented using D-dimensional space with their fitness/position value function in the following matrix set as

$$\begin{gathered} P = \begin{bmatrix} P_{1,1} & P_{1,2} & \cdots & \cdots & P_{1,a} \\ P_{2,1} & P_{2,2} & \cdots & \cdots & P_{2,a} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ P_{m,1} & P_{m,1} & \cdots & \cdots & P_{m,a} \end{bmatrix}, \\ VP = \begin{bmatrix} VP_1 \\ VP_2 \\ \vdots \\ VP_m \end{bmatrix}. \end{gathered} \tag{20}$$

It is important to note that moths and flames both are solutions. The moths are the real search agents that revolve around the search area, while flames are the moth's best position that is obtained yet. Hence, an individual moth hunts around a flame and updates it when it finds the best solution. Following this procedure, a moth never misses its best solution.

4.7.2. Updating Moths' Location/Positions. MFO utilizes three distinct functions to convergent the global optimum of the optimization issues. Mathematically, it is defined as follows:

$$\mathrm{MFO} = (L, M, E), \tag{21}$$

where L represents the first random positions of the moths (πl: $\varnothing \longrightarrow \{H, VH\}$), M represents that the motion of the moths in the search is (M: $H \longrightarrow H$), and E represents end of the search process (E: $H \longrightarrow$ true, false). The equation given below represents L function, which is used for the implementation of random distribution:

$$H(x, y) = (UB(x) - LB(y) \times \mathrm{rand} + LB(x), \tag{22}$$

where UB and LB refers to the upper and lower bound variables, respectively. As discussed before, the moths fly in the search area by means of transverse direction. There are

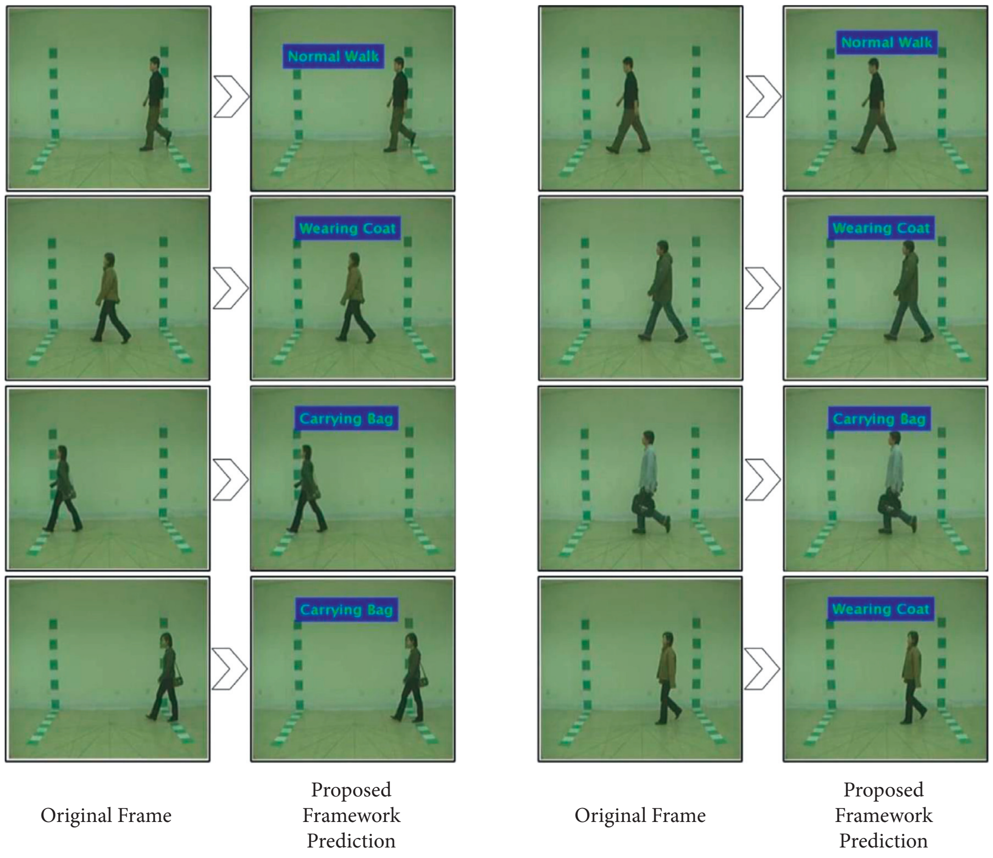


FIGURE 5: Proposed architecture labeled results on the CASIA B dataset.

three conditions that should be followed when applying a logarithmic spiral: (i) The spiral starting point should start from the moth; (ii) the spiral endpoint should be the location of the flame, and (iii) variation in the range of spiral should not extend from the search area. Thus, in the MFO algorithm, the logarithmic spiral can be defined as

$$S\left(H_x, P_y\right) = R_x . z^{qb} . \cos(2\pi b) + P_y, \tag{23}$$

where R_x represents space between the xth moth and yth flame (computed by equation 24), q represents a solution to define the shape of the logarithmic spiral, b represents a random range between [−1, 1].

$$R_x = \left|P_y - H_x\right|. \tag{24}$$

In MFO, the equalization among exploitation and examination is affirmed by the spiral motion of the moth near the flame in the search area. Moreover, to escape from falling in the trap of the local goal, the best solution has been kept in each step, and the moths fly around the flames by means of VP and VH matrices. Then, the update criteria are defined as follows:

4.7.3. Updating the Size of Flames. This part highlights to augment the manipulation of the MFO algorithm (i.e., updating the moths' location in m various positions in the search area may minimize the chance of exploitation of the best optimal solutions). However, minimizing the extent of flames helps to overcome this problem using the following equation:

$$\text{FLAME NO} = \text{ROUND}\left(O - c \times \frac{O - c}{I}\right), \tag{25}$$

where O refers to the maximum number of flames, c refers the current number of iterations, and I represents the maximum number of iterations. This equation returns the best features; however, during the analysis stage, it is observed that the best selected features contain some redundant information; therefore, we tried to overcome this problem and speedup the selection process based on Newton Raphson (NR) formulation. Mathematically, the NR method is defined as follows:

$$\delta_n = \delta_{n-1} - \frac{M\left(\delta_{n-1}\right)}{M'\left(\delta_{n-1}\right)}, \tag{26}$$

where $\delta_n \in Sf$ and Sf represent the selected features of moth-flame. Through the above formulation, a stop value is obtained that added in equation 25 for final selection.

$$\text{FLAME NO} = \text{ROUND}\left(\delta - cf \times \frac{\delta - cf}{I}\right). \tag{27}$$

The final selected features are passed to the extreme learning machine (ELM) for classification. A few visual predicted frames are shown in Figure 5.

Table 1: Proposed classification results of human gait recognition on the CASIA B dataset.

Method	Class	0°	18°	36°	54°	72°	90°	108°	126°	144°	162°	180°	Mean
LightweightDeep-ELM	NM	97.1	98.2	95	93.8	98.1	97.5	98.3	97.2	98	94	98.6	**96.89**
	BG	94.2	95.3	91	92.7	93	89.7	94.8	93.1	92.8	91.8	95.4	**93.07**
	CL	78.8	83.5	82.1	86.3	78.5	90.2	85.9	82	81.3	88.3	83.4	**83.66**
LightweightDeep-SVM	NM	96.2	97.5	96.1	92.8	97.9	96.2	98	97.5	97.1	93.2	98	96.40
	BG	92.5	93.6	93	92.2	94.1	87.6	94	93.5	92.1	92.4	93.4	92.58
	CL	79	82.8	81.5	86	79.1	87.4	83.9	82.3	80.7	87.5	82.1	82.93
LightweightDeep-FKNN	NM	93.2	94.5	92.9	93.1	94.6	91.8	93.5	94.8	94.5	93	94.1	93.63
	BG	87	89.1	90.5	89.6	91.4	82.6	90.8	89.4	87	89.3	90.7	88.85
	CL	73.5	78.4	77.2	81.6	75.3	82.2	80	77.5	76.1	82.4	78.5	78.42
LightweightDeep-EBT	NM	92.8	93.3	90.4	91.5	95	92.6	92.4	93.1	94	93.7	92.9	92.88
	BG	88.5	87.4	90.1	90.5	91.8	82	90.2	90.4	86.1	89.8	91.9	88.97
	CL	72.6	80.1	77	80.4	75	81.6	80.6	76.2	78.5	82	78.1	78.37
LightweightDeep-DT	NM	87.4	88.9	90.1	91.6	93	90.5	88	91.2	91.3	87.4	90.5	89.99
	BG	81.5	82.6	87.5	83.8	90.4	80.6	90	87.9	85.3	84	90.1	85.79
	CL	69.8	72.1	77	78.5	72.7	80	78.3	72.5	72.9	80.1	80.2	75.82

Bold values indicate the best values.

5. Experimental Results and Analysis

In this section of the proposed method, the detailed experimental process is presented in the form of tables and graphs. The proposed method is tested using two publicly available datasets, CASIA B and TUM GAID. Section 2 contains more information on both datasets. Instead of 70 : 30, the selected datasets are divided 50 : 50 for training and testing. The main reason for this portioning is to make the validation process more equitable. All of the results are based on 10-fold cross-validation. For the classification results, several classifiers are used, including the extreme learning machine (ELM), support vector machine (SVM), KNN, ensemble tree (EBT), and decision trees (DTs). The entire proposed framework is implemented on MATLAB 2021b using Personal Desktop Computer Corei7, 32 GB RAM, and 8 GB graphics card.

5.1. Results

5.1.1. CASIA B Dataset Results. The proposed method results for the CASIA B dataset are presented in the form of numerical values and a time plot in this section. Table 1 provides the classification results of the CASIA B dataset from all perspectives. Normally, researchers choose only a few angles, but in this work, we chose all 11 angles to test the capability of the proposed algorithm. Each angle has three classes: normal walk (NM), walk with a bag (BG), and walk while wearing a coat (WC) (CL). On this dataset, ELM performed better, with average accuracies of 96.89, 93.07, and 83.66% for NM, BG, and CL, respectively. For each angle, the obtained accuracy is above 90% that shows the proposed method effectiveness. A comparison of ELM with other classifiers such as SVM, FKNN, EBT, and DT shows that ELM performed better than all of them. Moreover, the time is also noted of each classifier as shown in Figure 6. From this figure, it is observed that the ELM and DT classifiers executed fast than the other listed methods.

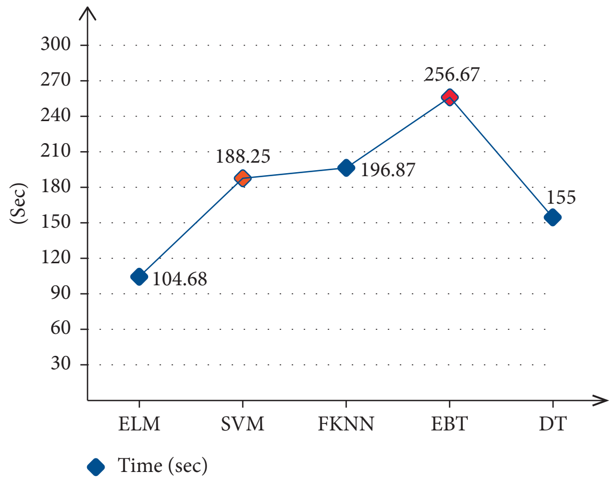


Figure 6: Average classification time of selected classifiers on the CASIA B dataset using the proposed method.

5.1.2. TUM GAID Dataset Results. The results of the proposed method on the TUM GAID dataset are given in Table 2. In this table, accuracy is computed of each class of the selected dataset such as normal walk, walk with a bag, and walk with shoes. Moreover, the average accuracy of each classifier is also computed. Many classifies are selected and ELM shows the better average accuracy of 98.60%. The rest of the classifiers obtained average accuracies of 97.25, 96.73, 96.91, and 96.26%, respectively. The computational time of each classifier is also computed and plotted in Figure 7. It can be seen from this figure that the ELM has a minimum computation time of 86.43 (sec) compared to the rest of the classifiers. Hence, overall, ELM classifier performed better using the proposed method on the TUM GAID dataset.

5.2. Discussion and Comparison. A detailed analysis of the proposed framework has been conducted in this section based on confidence interval and standard error means (SEM). As given in Tables 3 and 4, the proposed

Table 2: Proposed classification results of human gait recognition on the TUM GAID dataset.

Classifier	Class-based accuracy (%)			Mean accuracy (%)
	Normal walk	Walk with a bag	Walk with shoes	
LightweightDeep-ELM	**99.64**	**98.52**	**97.65**	**98.60**
LightweightDeep-SVM	98.58	96.92	96.25	97.25
LightweightDeep-FKNN	98.63	96.21	95.37	96.73
LightweightDeep-EBT	98.12	96.82	95.80	96.91
LightweightDeep-DT	97.52	96.03	95.24	96.26

Bold values indicate the best values.

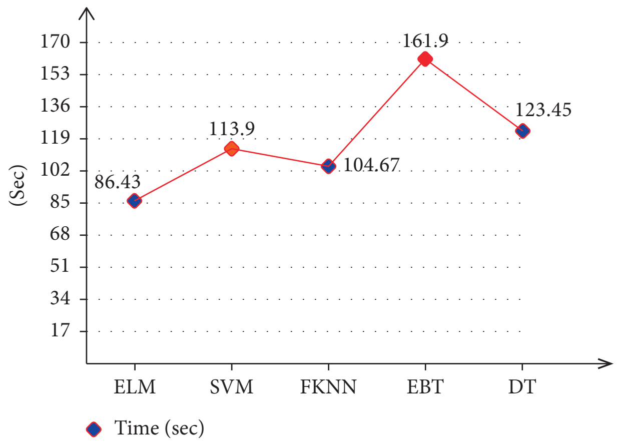


Figure 7: Average classification time of selected classifiers on the TUM GAID dataset using the proposed method.

Table 3: Confidence interval-based analysis of proposed framework on the CASIA B dataset.

Confidence level	Margin of error
Normal walk	
68.3%, $\sigma_{\overline{x}}$	96.005 ± 0.626 ($\pm$0.65%)
90%, 1.645 $\sigma_{\overline{x}}$	96.005 ± 1.029 ($\pm$1.07%)
95%, $1.960\sigma_{\overline{x}}$	96.005 ± 1.227 ($\pm$1.28%)
99%, $2.576\sigma_{\overline{x}}$	96.005 ± 1.612 ($\pm$1.68%)
Walk with a bag	
68.3%, $\sigma_{\overline{x}}$	92.59 ± 0.346 ($\pm$0.37%)
90%, $1.645\sigma_{\overline{x}}$	92.59 ± 0.57 ($\pm$0.62%)
95%, $1.960\sigma_{\overline{x}}$	92.59 ± 0.679 ($\pm$0.73%)
99%, $2.576\sigma_{\overline{x}}$	92.59 ± 0.893 ($\pm$0.96%)
Walk with a coat	
68.3%, $\sigma_{\overline{x}}$	82.53 ± 0.799 ($\pm$0.97%)
90%, $1.645\sigma_{\overline{x}}$	82.53 ± 1.314 ($\pm$1.59%)
95%, $1.960\sigma_{\overline{x}}$	82.53 ± 1.566 ($\pm$1.90%)
99%, $2.576\sigma_{\overline{x}}$	82.53 ± 2.058 ($\pm$2.49%)

LightweightDeep-ELM framework gives the better accuracy than other combinations on the CASIA B dataset. Similarly, the proposed framework (LightweightDeep-ELM) also obtained better results on the TUM GAID dataset. Moreover, the average computational time of each classifier for both datasets is also shown in Figures 6 and 7. The ELM execution time is minimum than the rest of the selected classifiers. To further analyze the performance of the ELM classifier, the proposed framework is executed 500 times and computed two values—minimum accuracy and maximum accuracy.

Table 4: Confidence interval-based analysis of proposed framework on the TUM GAID dataset.

Confidence level	Margin of error
Normal walk	
68.3%, $\sigma_{\overline{x}}$	98.767 ± 0.62 ($\pm$0.63%)
90%, $1.645\sigma_{\overline{x}}$	98.767 ± 1.02 ($\pm$1.03%)
95%, $1.960\sigma_{\overline{x}}$	98.767 ± 1.215 ($\pm$1.23%)
99%, $2.576\sigma_{\overline{x}}$	98.767 ± 1.597 ($\pm$1.62%)
Walk with a bag	
68.3%, $\sigma_{\overline{x}}$	97.71 ± 0.573 ($\pm$0.59%)
90%, $1.645\sigma_{\overline{x}}$	97.71 ± 0.942 ($\pm$0.96%)
95%, $1.960\sigma_{\overline{x}}$	97.71 ± 1.123 ($\pm$1.15%)
99%, $2.576\sigma_{\overline{x}}$	97.71 ± 1.475 ($\pm$1.51%)
Walk with a coat	
68.3%, $\sigma_{\overline{x}}$	96.925 ± 0.513 ($\pm$0.53%)
90%, $1.645\sigma_{\overline{x}}$	96.925 ± 0.843 ($\pm$0.87%)
95%, $1.960\sigma_{\overline{x}}$	96.925 ± 1.005 ($\pm$1.04%)
99%, $2.576\sigma_{\overline{x}}$	96.925 ± 1.321 ($\pm$1.36%)

Based on the minimum and maximum accuracy, the standard error mean is computed. Through SEM, a confidence error is obtained that shows the consistency of proposed framework.

Table 3 provides the confidence interval-based analysis of the CASIA B dataset. Confidence level and margin of error (MoE) are calculated for each class such as walk, bag, and coat. We selected several confidence levels such as 68.3%, $\sigma_{\overline{x}}$; 90%, $1.645\sigma_{\overline{x}}$; 95%, $1.960\sigma_{\overline{x}}$; and 99%, $2.576\sigma_{\overline{x}}$ and obtained MOE for each is noted as given below. Based on the MoE, it is observed that the proposed framework showed consistent performance on the CASIA B dataset after 500 iterations. Similarly, Table 4 provides the proposed confidence interval-based analysis of the TUM GAID dataset. From this table, it is also confirmed that the proposed method's accuracy is consistent after the numbers of iterations.

At the end, a detailed comparison is conducted with recent techniques for both selected datasets such as CASIA B and TUM GAID. Table 5 provides the comparison of the proposed method accuracy with recent techniques on the CASIA B dataset. In this table, the authors of [34] obtained an average accuracy of 51.4% on the CASIA B dataset. The authors in [35] improved the average accuracy and reached to 84.2% that was later further improved by [36] of 87.5%. Recently, the authors of [37] obtained an average accuracy of 89.66% on the CASIA B dataset that is improved then the previous noted techniques. Our method achieved an accuracy of 91.20% on the CASIA B dataset that is improved

Table 5: Comparison of proposed method results on the CASIA B dataset with recent techniques.

Reference	Year	Datasets	NM	BG	CL	Mean (%)
[34]	2018	CASIA B	68.1	54.7	31.5	51.4
[35]	2019	CASIA B	95.0	87.2	70.4	84.2
[36]	2022	CASIA B	96.0	91.6	74.8	87.5
[37]	2022	CASIA B	96	92	81	89.66
Proposed		CASIA B	**96.89**	**93.07**	**83.66**	**91.20**

Bold values indicate the best values.

Table 6: Comparison of proposed method results with recent techniques on TUM GAID.

Reference	Year	Datasets	*N*	*B*	*S*	Mean (%)
[26]	2014	TUM GAID	99.4	59.4	94.5	84.4
[38]	2017	TUM GAID	98.7	91.1	94.5	96.7
[39]	2017	TUM GAID	99.7	98.1	95.8	97.9
[40]	2021	TUM GAID	99.4	97.4	96.4	97.73
Proposed		TUM GAID	**99.64**	**98.52**	**97.65**	**98.60**

Bold values indicate the best values.

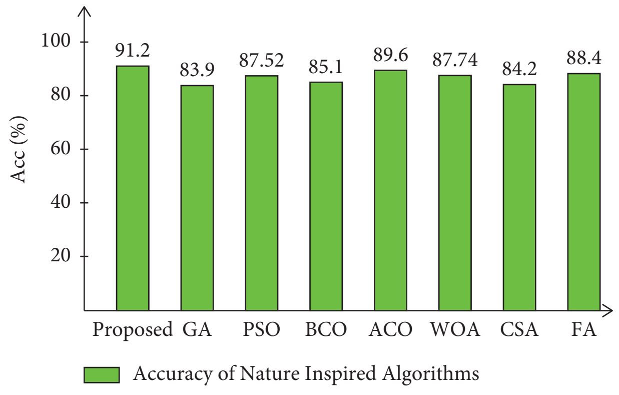


Figure 8: Comparison of proposed optimization accuracy with several other nature-inspired algorithms.

than the existing techniques. Similarly, the comparison of the TUM GAID dataset is given in Table 6. In this table, it is noted that the recently achieved accuracies were 84.4%, 96.7%, 97.9%, and 97.73%. Our proposed method obtained an accuracy of 98.60% that is improved than the recent state-of-the-art (SOTA) techniques.

Finally, the improved moth-flame optimization algorithm is compared to several other nature-inspired algorithms (Figure 8) such as the genetic algorithm, particle swarm optimization, bee colony optimization, ant colony optimization, whale optimization, crow search, and firefly algorithm. This graph shows that the proposed optimization algorithm outperforms the other compared algorithms in terms of accuracy. Moreover, the gait is important for several purposes such as assisting those suffering from Parkinson's disease [41, 42]. In this work, we used Adam as an optimizer [18] during the training of deep learning models instead of stochastic gradient descent (SGD). For the gait recognition task, SGD is not performed better than Adam due to a high number of video frames. As we know, Adam is known to be computationally fast, requires less memory, and needs little tuning.

6. Conclusion

Human gait recognition using lightweight deep learning models and improved moth-flame optimization algorithm has been presented in this work. Two lightweight pretrained CNN models were fine-tuned and deep transfer learning based trained. Features are extracted from the global average pooling layer and fused using a new approach named DCCA. Furthermore, an optimization algorithm is developed for the selection of the best features. The proposed method was compared based on several classifiers such as ELM and SVM and found ELM is more suitable based on accuracy and time. Two publicly available datasets were employed for the validation process and achieved an improved average accuracy of 91.20 and 98.60%. The key findings of this work are as follows: (i) freezing few middle layers can train a model with less time but it is a chance to sacrifice the better accuracy; (ii) fusion of lightweight models features using DCCA approach is time-consuming but at the end, better information in the form of features is obtained; and (iii) improved optimization algorithm provides the better accuracy and reduces the computational time. In the future, a new scratch-based CNN model will be developed for human gait recognition.

Data Availability

The datasets used in this work are publicly available (http://www.cbsr.ia.ac.cn/english/Gait%20Databases.asp).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

All authors contributed equally in this work and have read and agreed to the published version of the manuscript.

Acknowledgments

This research was supported by Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (P0012724, The Competency Development Program for Industry Specialist) and the Soonchunhyang University Research Fund.

References

[1] M. Chaa, Z. Akhtar, and A. Lati, "Contactless person recognition using 2D and 3D finger knuckle patterns," *Multimedia Tools and Applications*, vol. 81, pp. 8671–8689, 2022.
[2] F. Gao, T. Tian, T. Yao, and Q. Zhang, "Human gait recognition based on multiple feature combination and parameter optimization algorithms," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 6693206, 14 pages, 2021.
[3] S. K, F. F. Wahid, and R. G, "Statistical features from frame aggregation and differences for human gait recognition," *Multimedia Tools and Applications*, vol. 80, no. 12, pp. 18345–18364, 2021.

[4] H. Arshad, M. A. Khan, M. I. Sharif et al., "A multilevel paradigm for deep convolutional neural network features selection with an application to human gait recognition," *Expert Systems*, vol. 2020, Article ID e12541, 18 pages, 2020.

[5] X. Li, Y. Makihara, C. Xu, Y. Yagi, and M. Ren, "Gait recognition via semi-supervised disentangled representation learning to identity and covariate features," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13309–13319, IEEE, Seattle, WA, USA, June 2020.

[6] M. Kumar, N. Singh, R. Kumar, S. Goel, and K. Kumar, "Gait recognition based on vision systems: a systematic survey," *Journal of Visual Communication and Image Representation*, vol. 75, Article ID 103052,, 2021.

[7] A. Liaqat, M. A. Khan, M. Sharif et al., "Gastric tract infections detection and classification from wireless capsule endoscopy using computer vision techniques: a review," *Current medical imaging*, vol. 16, no. 10, pp. 1229–1242, 2020.

[8] T. Teepe, A. Khan, J. Gilg, F. Herzog, S. Hörmann, and G. Rigoll, "GaitGraph: graph convolutional network for skeleton-based gait recognition," in *Proceedings of the 2021 IEEE International Conference on Image Processing (ICIP)*, pp. 2314–2318, IEEE, Anchorage, AK, USA, September 2021.

[9] L. Yao, W. Kusakunniran, Q. Wu, J. Zhang, Z. Tang, and W. Yang, "Robust gait recognition using hybrid descriptors based on skeleton gait energy image," *Pattern Recognition Letters*, vol. 150, pp. 289–296, 2021.

[10] A. Mehmood, M. A. Khan, M. Sharif et al., "Prosperous human gait recognition: an end-to-end system based on pre-trained CNN features selection," *Multimedia Tools and Applications*, vol. 202021 pages, 2020.

[11] A. Sepas-Moghaddam and A. Etemad, "Deep gait recognition: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1, 2022.

[12] S. Hou, X. Liu, C. Cao, and Y. Huang, "Gait quality aware network: toward the interpretability of silhouette-based gait recognition," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–11, 2022.

[13] S. Gul, M. I. Malik, G. M. Khan, and F. Shafait, "Multi-view gait recognition system using spatio-temporal features and deep learning," *Expert Systems with Applications*, vol. 179, Article ID 115057, 2021.

[14] F. Saleem, M. A. Khan, M. Alhaisoni et al., "Human gait recognition: a single stream optimal deep learning features fusion," *Sensors*, vol. 21, no. 22, p. 7584, 2021.

[15] G. Liu, S. Zhong, and T. Li, "Gait recognition method of temporal-spatial HOG features in critical separation of Fourier correction points," *Future Generation Computer Systems*, vol. 94, pp. 11–15, 2019.

[16] X. Wang, S. Feng, and W. Q. Yan, "Human gait recognition based on self-adaptive hidden markov model," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 3, pp. 963–972, 2021.

[17] S. Das Choudhury and T. Tjahjadi, "Robust view-invariant multiscale gait recognition," *Pattern Recognition*, vol. 48, no. 3, pp. 798–811, 2015.

[18] M. N. Y. Ali, M. G. Sarowar, M. L. Rahman, J. Chaki, N. Dey, and J. M. R. S. Tavares, "Adam deep learning with SOM for human sentiment classification," *International Journal of Ambient Computing and Intelligence*, vol. 10, no. 3, pp. 92–116, 2019.

[19] M. I. Sharif, M. A. Khan, A. Alqahtani et al., "Deep learning and kurtosis-controlled, entropy-based framework for human gait recognition using video sequences," *Electronics*, vol. 11, no. 3, p. 334, 2022.

[20] V. B. Semwal, A. Mazumdar, A. Jha, N. Gaud, and V. Bijalwan, "Speed, cloth and pose invariant gait recognition-based person identification," *Machine Learning: Theoretical Foundations and Practical Applications*, vol. 87, pp. 39–56, 2021.

[21] A. Zhao, J. Dong, J. Li, L. Qi, and H. Zhou, "Associated spatio-temporal capsule network for gait recognition," *IEEE Transactions on Multimedia*, vol. 24, pp. 846–860, 2022.

[22] 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, Article ID 107069, 2020.

[23] S. K. Gupta and P. Chattopadhyay, "Gait recognition in the presence of co-variate conditions," *Neurocomputing*, vol. 454, pp. 76–87, 2021.

[24] X. Chen, X. Luo, J. Weng, W. Luo, H. Li, and Q. Tian, "Multi-view gait image generation for cross-view gait recognition," *IEEE Transactions on Image Processing*, vol. 30, pp. 3041–3055, 2021.

[25] S. Hou, X. Liu, C. Cao, and Y. Huang, "Set residual network for silhouette-based gait recognition," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 3, pp. 384–393, 2021.

[26] M. Hofmann, J. Geiger, S. Bachmann, B. Schuller, and G. Rigoll, "The tum gait from audio, image and depth (gaid) database: multimodal recognition of subjects and traits," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 195–206, 2014.

[27] S. Yu, D. Tan, and T. Tan, "A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)*, pp. 441–444, IEEE, Hong Kong, China, 20-24 August 2006.

[28] D. R. Nayak, D. Das, R. Dash, S. Majhi, and B. Majhi, "Deep extreme learning machine with leaky rectified linear unit for multiclass classification of pathological brain images," *Multimedia Tools and Applications*, vol. 79, no. 21-22, pp. 15381–15396, 2020.

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[30] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, IEEE, Salt Lake City, UT, USA, June 2018.

[31] C. Buiu, V.-R. Dănăilă, and C. N. Răduţă, "MobileNetV2 ensemble for cervical precancerous lesions classification," *Processes*, vol. 8, no. 5, p. 595, 2020.

[32] N. Dey, A. S. Ashour, and S. Bhattacharyya, *Applied Nature-Inspired Computing: Algorithms and Case Studies*, Springer, Berlin, Germany, 2020.

[33] S. Mirjalili, "Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm," *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015.

[34] Y. He, J. Zhang, H. Shan, and L. Wang, "Multi-task GANs for view-specific feature learning in gait recognition," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 1, pp. 102–113, 2019.

[35] H. Chao, Y. He, J. Zhang, and J. Feng, "Gaitset: regarding gait as a set for cross-view gait recognition," in *Proceedings of the*

AAAI conference on artificial intelligence, pp. 8126–8133, Hawaii, HI, USA, January 2019.

[36] F. Han, X. Li, J. Zhao, and F. Shen, “A unified perspective of classification-based loss and distance-based loss for cross-view gait recognition,” *Pattern Recognition*, vol. 125, Article ID 108519, 2022.

[37] H. Li, Y. Qiu, H. Zhao et al., “Gaitslice: a gait recognition model based on spatio-temporal slice features,” *Pattern Recognition*, vol. 124, Article ID 108453, 2022.

[38] F. M. Castro, M. J. Marín-Jiménez, N. Guil, S. López-Tapia, and N. P. de la Blanca, “Evaluation of CNN architectures for gait recognition based on optical flow maps,” in *Proceedings of the International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 1–5, IEEE, Darmstadt, Germany, September 2017.

[39] F. M. Castro, M. J. Marín-Jiménez, N. Guil, and N. Pérez de la Blanca, “Automatic learning of gait signatures for people identification,” in *Proceedings of the International Work-Conference On Artificial Neural Networks*, pp. 257–270, Salamanca, Spain, June 2017.

[40] R. Delgado-Escano, F. M. Castro, N. Guil, and M. J. Marin-Jimenez, “Gaitcopy: disentangling appearance for gait recognition by signature copy,” *IEEE Access*, vol. 9, pp. 164339–164347, 2021.

[41] A. El-Attar, A. S. Ashour, N. Dey, H. Abdelkader, M. M. Abd El-Naby, and R. S. Sherratt, “Discrete wavelet transform-based freezing of gait detection in parkinson’s disease,” *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 33, no. 4, pp. 543–559, 2021.

[42] A. El-Attar, A. S. Ashour, N. Dey, H. A. El-Kader, M. M. A. El-Naby, and F. Shi, “Hybrid DWT-FFT features for detecting freezing of gait in parkinson’s disease,” in *Information Technology and Intelligent Transportation Systems*, pp. 117–126, IOS Press, Amsterdam, Netherlands, 2019.