Article

Combined Regularized Discriminant Analysis and Swarm Intelligence Techniques for Gait Recognition

Tomasz Krzeszowski *[ID] **and Krzysztof Wiktorowicz** [ID]

Faculty of Electrical and Computer Engineering, Rzeszow University of Technology,
al. Powstancow Warszawy 12, 35-959 Rzeszow, Poland; kwiktor@prz.edu.pl
* Correspondence: tkrzeszo@prz.edu.pl

Abstract: In the gait recognition problem, most studies are devoted to developing gait descriptors rather than introducing new classification methods. This paper proposes hybrid methods that combine regularized discriminant analysis (RDA) and swarm intelligence techniques for gait recognition. The purpose of this study is to develop strategies that will achieve better gait recognition results than those achieved by classical classification methods. In our approach, particle swarm optimization (PSO), grey wolf optimization (GWO), and whale optimization algorithm (WOA) are used. These techniques tune the observation weights and hyperparameters of the RDA method to minimize the objective function. The experiments conducted on the GPJATK dataset proved the validity of the proposed concept.

1. Introduction

Biometric authentication (also known as biometrics) refers to identifying or verifying individuals based on their biological or behavioral traits [1]. There are many different biometric traits among which can be distinguished the face, iris, fingerprint, palm print, voice, signature, or gait. Typically, gait is a manifestation of an individual's walking style; hence, its recognition means identifying a person by his/her way of walking. The major advantages of gait are: noninvasive, can be captured at a distance, hard to conceal, and non-cooperative. These advantages make it an ideal trait for visual surveillance systems [2]. However, the recognition performance of existing methods is limited by the influence of a large number of covariate factors affecting both appearance and dynamics of the gait, e.g., variations in footwear and clothing, viewpoint variations, changes in the characteristics of the surface on which movement occurs, various carrying conditions, injuries affecting movement, and so on. These are the reasons why gait recognition has been extensively studied in recent years.

Gait recognition methods [3] can be categorized as model-free (appearance-based) [2,4–12] and model-based [13–22]. Most gait recognition studies are based on model-free approaches that employ the whole motion pattern of the human body. Several techniques were proposed to characterize this motion pattern, such as the gait energy image (GEI) [2,5,7,10], which is a spatio-temporal gait representation, GEI region bounded by legs (RBL) [8], human body contours [9], and dense optical flow field [4]. These methods are strongly based on silhouette extraction and therefore are not resistant to changing clothes or carrying luggage. It is also worth noting that most of them can only achieve correct results from a specific point of view, usually side view [4,5,7–9,16]. The recent research on model-free methods has focused on eliminating these drawbacks [6,10–12]. Model-based methods infer gait signature directly by modeling the underlying kinematics of human motion. The methods of this approach initially focused on using only static body parameters for recognition, such as stride

length, which were updated over time [13,14]. Yam et al. [14] have extended this concept by analyzing the movement of the legs and the angles between them. In Ref. [15], the authors proposed a method that uses a motion-based model and elliptic Fourier descriptors to extract the key features of gait. Deng et al. [18] proposed a method that combines spatio-temporal and kinematic gait features. The fusion of two different features gives a comprehensive characterization of gait dynamics, which is less sensitive to walking conditions. In [21], the authors presented a gait recognition method that uses a 3D model of the human body and particle swarm optimization to obtain gait features. The number of obtained features was reduced using the multilinear principal component analysis (MPCA).

In the recognition process, various classification techniques are used; most often, they are classical methods such as k-nearest neighbors (kNN) [2,4,10,14–17,19,23], multilayer perceptron (MLP) [21], support vector machine [9,16,24] linear discriminant analysis (LDA) [16], and radial basis function neural networks [18]. The main focus in these papers is on developing descriptors that better describe gait features, rather than introducing new classification methods with better recognition ability. This is a traditional approach that is based on a clear separation between the descriptors and classifier model. On the other hand, in the papers of recent years, the introduction of new classification methods such as deep learning [11,20,22] or hybrid methods [12], in which the description and classification steps cannot be easily distinguished, is increasingly visible. In [20], the authors utilized a 3D convolutional neural network (CNN) and long short-term memory neural networks for training the classification models. They then used a grey wolf optimizer to tune the fusion parameters of each modality to boost the recognition performance of the system. Chao et al. [11] proposed a deep learning model called GaitSet. In this method, the CNN is used to extract frame-level features from each silhouette independently. Next, an operation called set pooling is used to aggregate frame-level features into a single set-level feature. In the end, a structure called horizontal pyramid mapping is used to map the set-level feature into a more discriminative space to obtain the final representation. The proposed method can extract spatial and temporal information more effectively than other methods regarding gait as a template or sequence. In turn, in the paper [12], the authors used a hybrid approach and combine the improved local coupled extreme learning machine and PSO for the classification process. A Gabor filter was used to extract gait features from the GEI and linear discriminant analysis was used to dimensionality reduction.

From the literature review, it is seen that most studies considering the traditional approach are devoted to developing gait descriptors, rather than introducing new classification methods. In this paper, we propose hybrid methods that combine the RDA and swarm intelligence techniques for gait recognition. In our approach, the PSO, GWO, and WOA are used to tune the observation weights and hyperparameters of the RDA model. To the best of our knowledge, the GWO and WOA algorithms have not been used before for this purpose. In the learning process, the confusion value is used as an objective function. The proposed methods are tested on a database of 414 gait cycles belonging to 32 different persons [21]. Summarizing, the main contributions of this paper can be stated as:

- proposing a combination of regularized discriminant analysis and particle swarm optimization for gait recognition,
- proposing a combination of regularized discriminant analysis and grey wolf optimization,
- proposing a combination of regularized discriminant analysis and whale optimization algorithm,
- comparing and improving the results obtained in the paper [21].

The structure of this article is as follows: Section 2 contains the description of the dataset and methods used in the recognition process. In particular, the structure of the gait recognition system, building classification models, and swarm intelligence methods are described. The experimental results are presented in Section 3. The results obtained by eight methods are presented, of which three were proposed by the authors. Section 4 contains the discussion of the achieved results. Finally, the conclusions are given in Section 5.

2. Material and Methods

2.1. Gait Dataset

The publicly available gait dataset (GPJATK) was used in the experiments [21]. The dataset consists of 166 data sequences (414 gait cycles) representing 32 people (10 women and 22 men). The sequences are divided into three subsets: 128 sequences (325 gait cycles) in which each of 32 individuals was dressed in his/her clothes; 24 sequences (58 gait cycles) in which 6 of 32 individuals (person #26–#31) changed clothes; and 14 sequences (31 gait cycles) in which 7 of the individuals (person #26–#32) had a backpack on his/her back. Each sequence contains video data (960x540@25fps) recorded using four calibrated and synchronized cameras and data from a markerless and marker-based motion capture systems. The synchronization between videos and motion capture data has been realized using Vicon MX Giganet. Our research is based on data obtained by a markerless motion capture system [25], which uses the annealed PSO algorithm in the motion capture process and data from four synchronized and calibrated cameras.

2.2. Gait Recognition System

A typical model-based system for gait recognition is presented in Figure 1. Such a system consists of gait capture, a feature extraction module, and a classifier. The objective of the system is to determine the identity of a gait sample using a database consisting of gait patterns from a set of known subjects. In the first step, one or more video-cameras are used to register the user's image in a scene. In a preprocessing phase, image processing, i.e., background subtraction, body silhouette extraction, and edges extraction, is performed. The kinematic model of human motion is used in the next step to extract gait features that will define a gait signature. In the used gait dataset [21], each gait cycle is treated as a data sample represented by a third-order tensor with the dimension $32 \times 11 \times 3$. The first dimension, equal to 32, is the average time of the gait cycle. The motion data was filtered using a moving average of length nine samples to the original data. The second dimension of the tensor is equal to the number of bones (excluding pelvis), i.e., 10 plus one element for storing a person height and distance between ankles. The third dimension relates to three angles, except the 11th vector that contains a person's height, distance between ankles, and value of zero to maintain alignment with the rest of the vectors. Such a gait signature is then reduced using the MPCA algorithm [26]. The last element of the system is the classifier block that we focus in this article.
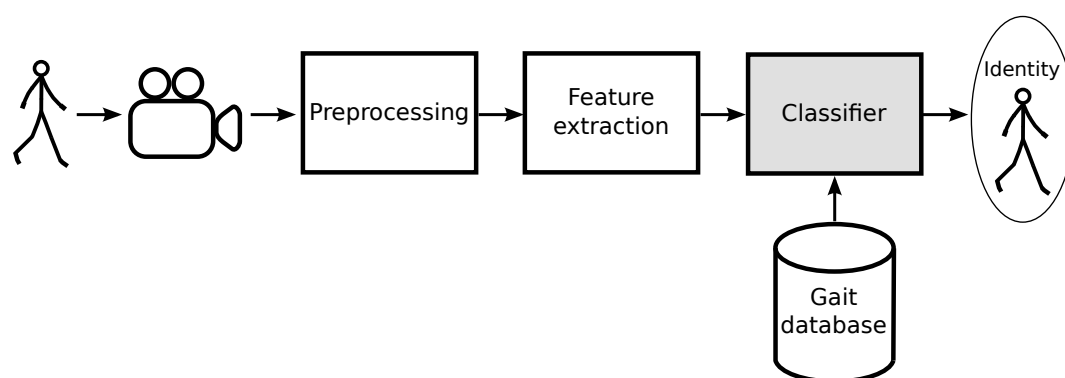
**Figure 1.** Structure of a gait recognition system.

2.3. Building Classification Model for Gait Recognition

Building a gait classification model involves two main stages, which include training the model and testing it. For this purpose, the gait sequence database is divided into three sets: the training set, validation set, and test set (Figure 2). These three datasets are commonly used in different stages of the model building. Separating the dataset into these three subsets is used to avoid overfitting of

the model. Initially, the model is fit on the training set, which is a set of observations used to fit the parameters of the RDA model. It should be noted that, in [21], only two sets were defined: training and testing. However, in the proposed approach, due to the optimization of the classifier parameters, an additional validation set is separated from the training data. The validation dataset is used for an evaluation of the fitted model while training the model's hyperparameters. After building the model, the test set is used for testing, that is, for predicting the classifier's output for data that has never been used in the training phase. Model training is carried out using one of the hybrid methods, in which swarm intelligence techniques optimize the observation weights and hyperparameters of the RDA. This problem is well suited for swarm optimization techniques because it creates a large search space to be explored. This search space is determined by the number of observation weights, the number of hyperparameters, and by the fact that all these variables are real values in the specified intervals. During the optimization, the following objective function, expressed as the confusion value, is used:

$$\text{objective function (confusion)} = \frac{\text{number of samples misclassified}}{\text{number of all samples}} \tag{1}$$

where the samples are taken from the validation set. This objective function is minimized using one of the swarm intelligence methods, i.e., particle swarm optimization, grey wolf optimization, or whale optimization algorithm.
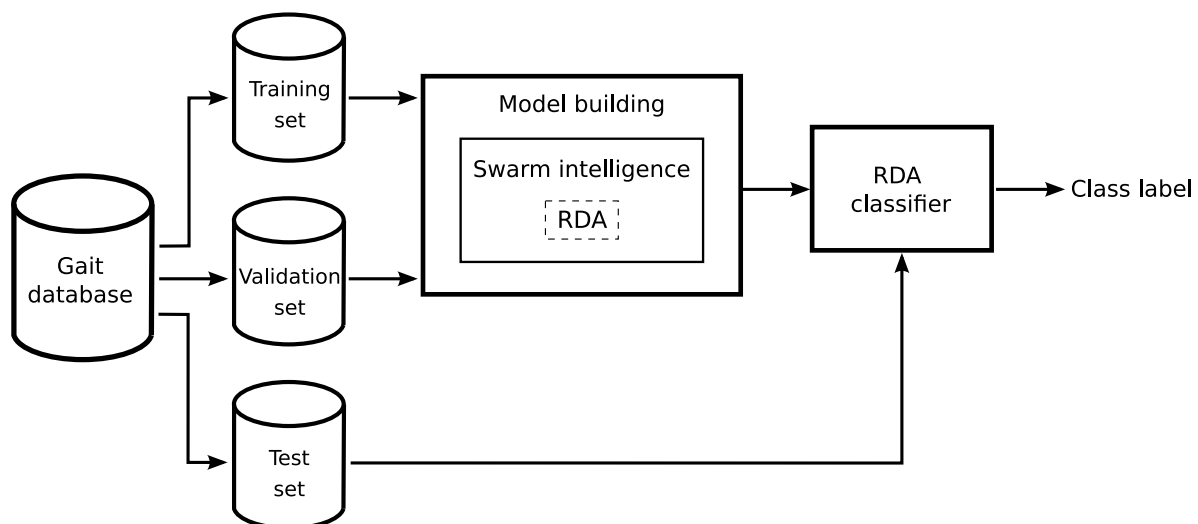
Figure 2. The idea of building the RDA classification model.

After building the RDA classifier, it is used to determine the correct classified ratio (CCR). The CCR is a ratio of correctly classified samples to the total number of samples in the test subset.

2.4. Regularized Discriminant Analysis

Linear discriminant analysis was developed by Sir Ronald Fisher in 1936 [27]. The original method proposed by Fisher was described for a 2-class problem, and it was in 1948 generalized as multi-class problems by Rao [28]. The LDA is a transformation technique used in statistics and machine learning to find linear combinations of features that separate classes of objects. The combinations obtained by this method may be used as:

- dimensionality reduction and feature extraction before classification,
- a linear classifier (considered in this paper).

The LDA consists of statistical properties of data calculated for each class. For a single variable, these are the mean and the variance of the variable. For multiple variables, these are the means and the covariance matrix. These statistical properties are estimated from the data and used to formulate

an equation for making predictions. It should be emphasized that the use of the LDA is not associated with problems when the number of observations is greater than the dimension of each observation. Problems arise when the opposite is true, which makes the covariance matrix singular and cannot be inverted. To resolve this problem, instead of using the covariance matrix directly, a regularization of this matrix is used. This approach is applied to the regularized discriminant analysis method, in which the regularized covariance matrix $\widehat{\boldsymbol{\Sigma}}_\gamma$ is given by [29]:

$$\widehat{\boldsymbol{\Sigma}}_\gamma = (1 - \gamma)\widehat{\boldsymbol{\Sigma}} + \gamma\mathbf{I} \tag{2}$$

where $\widehat{\boldsymbol{\Sigma}}$ is the covariance matrix, $\mathbf{I}$ is the identity matrix, and $\gamma \in [0, 1]$ is the amount of regularization. The RDA introduces regularization into the covariance matrix estimate, enabling a solution to be obtained and allowing different influences of variables on the classification model. In addition to the parameter γ, the RDA model uses the parameter δ that acts as a threshold: if a model coefficient has the magnitude smaller than δ, the RDA sets this coefficient to zero, and the corresponding predictor can be eliminated from the model.

The output of the RDA classifier $\widehat{y}$ is calculated so as to minimize the classification cost [30]:

$$\widehat{y} = \underset{y=1,\dots,K}{\arg\min} \sum_{k=1}^{K} \widehat{P}(x|k)C(y|k) \tag{3}$$

where K is the number of classes, $\widehat{P}(x|k)$ is the posterior probability of class k for observation x, $C(y|k)$ is the cost of classifying an observation as y when its true class is k. The RDA used in this paper constructs weighted classifiers using the following scheme. Suppose $\mathbf{M}$ is an N-by-K class membership matrix such that $M_{nk} = 1$ if observation n is from class k, $M_{nk} = 0$, otherwise. The estimate of the class mean for weighted data with positive weights w_n is [30]

$$\widehat{\mu}_k = \frac{\sum_{n=1}^{N} M_{nk}w_n x_n}{\sum_{n=1}^{N} M_{nk}w_n} \tag{4}$$

The estimate of the covariance matrix is

$$\widehat{\Sigma} = \frac{\sum_{n=1}^{N} \sum_{k=1}^{K} M_{nk}w_n(x_n - \widehat{\mu}_k)(x_n - \widehat{\mu}_k)^T}{1 - \sum_{k=1}^{K} \frac{W_k^{(2)}}{W_k}} \tag{5}$$

where $W_k = \sum_{n=1}^{N} M_{nk}w_n$ is the sum of the weights for class k, and $W_k^{(2)} = \sum_{n=1}^{N} M_{nk}w_n^2$ is the sum of squared weights per class.

2.5. Particle Swarm Optimization

A particle swarm optimization algorithm was developed by Kennedy and Eberhart [31]. This algorithm is based on the social behavior of organisms living in large groups. In the PSO, a group of agents called particles forms a swarm, where each particle represents a point in a multidimensional space. The particles explore this space in order to find the optimal solution. Each particle in the swarm is attracted both to its best position and the best position found by other particles. The best solution is obtained by minimizing the objective function.

Each particle has its position ($\mathbf{x}$) and velocity ($\mathbf{v}$). The velocity $\mathbf{v}_k$ of the kth particle is determined using the following equation [31]:

$$\mathbf{v}_k(t+1) = \omega\mathbf{v}_k(t) + c_1\mathbf{r}_1(\mathbf{pbest}_k(t) - \mathbf{x}_k(t)) + c_2\mathbf{r}_2(\mathbf{gbest}(t) - \mathbf{x}_k(t)) \tag{6}$$

where t is the current iteration number, ω is the inertia weight, $\mathbf{r}_1$, $\mathbf{r}_2$ are vectors of random numbers in the range [0,1], c_1 is the cognitive coefficient, and c_2 is the social coefficient. It is seen that the update

of the velocity is a weighted sum of the previous velocity $\mathbf{v}_k(t)$, the difference between the current position and the personal best position (**pbest**), and the difference between the current position and the global best position (**gbest**). The position $\mathbf{x}_k$ of the kth particle is updated according to the equation

$$\mathbf{x}_k(t+1) = \mathbf{x}_k(t) + \mathbf{v}_k(t) \tag{7}$$

After updating the velocity and the position, the objective function is calculated to determine the personal and global positions.

2.6. Grey Wolf Optimization

A grey wolf optimizer is another swarm intelligence technique used to solve optimization problems [32]. The GWO algorithm is inspired by the behavior and hierarchy of grey wolves in nature, searching for the optimal way to attack their prey. In the hierarchy of grey wolves, the most dominating is alpha (α), which leads the entire group. The other wolves are beta (β) and delta (δ), which help to control the rest of the wolves considered as omega (ω). The omega wolves have the lowest ranking in the hierarchy. The main phases of grey wolf hunting are: (a) tracking, chasing, and approaching; (b) chasing, encircling, and harassing; (c) attacking.

2.6.1. Encircling Prey

Grey wolves encircle the prey during the hunt, which can be mathematically modeled by the following equation [32]:

$$\mathbf{X}(t+1) = \mathbf{X}_p(t) - \mathbf{A} \cdot \mathbf{D} \tag{8}$$

where

$$\mathbf{D} = |\mathbf{C} \cdot \mathbf{X}_p(t) - \mathbf{X}(t)| \tag{9}$$

and $\mathbf{X}(t)$ is the current position of a grey wolf at iteration t, $\mathbf{X}_p(t)$ is the position of the prey, and "$\cdot$" is an element-by-element multiplication. The coefficient vectors $\mathbf{A}$, $\mathbf{C}$ are determined as follows:

$$\mathbf{A} = 2\mathbf{a} \cdot \mathbf{r}_1 - \mathbf{a} \tag{10}$$

$$\mathbf{C} = 2\mathbf{r}_2 \tag{11}$$

where components of $\mathbf{a}$ are linearly decreased from 2 to 0 through iterations and $\mathbf{r}_1$, $\mathbf{r}_2$ are random vectors in $[0, 1]$.

2.6.2. Hunting

In the GWO, we assume that the α, β, and δ are the best solutions for the entire population. Therefore, the other wolves should update their position according to the positions of the three agents. The following formula is used to calculate the positions of search agents [32]:

$$\mathbf{X}(t+1) = \frac{1}{3}(\mathbf{X}_1(t) + \mathbf{X}_2(t) + \mathbf{X}_3(t)) \tag{12}$$

where

$$\mathbf{X}_1 = \mathbf{X}_\alpha(t) - \mathbf{A}_1 \cdot \mathbf{D}_\alpha \tag{13}$$

$$\mathbf{X}_2 = \mathbf{X}_\beta(t) - \mathbf{A}_2 \cdot \mathbf{D}_\beta \tag{14}$$

$$\mathbf{X}_3 = \mathbf{X}_\delta(t) - \mathbf{A}_3 \cdot \mathbf{D}_\delta \tag{15}$$

and

$$D_\alpha = |C_1 \cdot X_\alpha - X| \tag{16}$$

$$D_\beta = |C_2 \cdot X_\beta - X| \tag{17}$$

$$D_\delta = |C_3 \cdot X_\delta - X| \tag{18}$$

The vectors A_1, A_2, A_3 are obtained using Equation (10), while C_1, C_2, C_3 are obtained using Equation (11).

2.6.3. Attacking Prey (Exploitation) and Search for Prey (Exploration)

The grey wolves start the attack, once the prey stops moving. To model the process of approaching the prey, the GWO linearly decrease all the values of **a** from 2 to 0 according to the equation [32]

$$\mathbf{a} = 2 - \frac{2t}{T_{max}} \tag{19}$$

where T_{max} is the total number of iterations of the algorithm. The change of **a** affects the coefficient vector **A**, which controls the behavior of search agents. If $|A| < 1$, the wolf will move towards the prey; on the other hand, if $|A| > 1$, the wolf will diverge from the prey in order to find new better prey. In addition, the vector **C** which contains a random value in the range [0,2] is employed to help the algorithm to avoid being trapped in the local optima.

2.7. Whale Optimization Algorithm

The whale optimization algorithm is a nature-inspired metaheuristic technique for solving optimization problems [33]. This algorithm mimics the social behavior of humpback whales realized in the bubble-net hunting strategy. The WOA is based on three operators to simulate the search for prey, encircling prey, and bubble-net foraging.

2.7.1. Encircling Prey

Humpback whales encircle pray after recognizing its position. In this phase, the search agents attempt to change their locations towards the best search agents. This behavior is expressed by the following formula [33]:

$$X(t+1) = X^*(t) - A \cdot D \tag{20}$$

where

$$D = |C \cdot X^*(t) - X(t)| \tag{21}$$

and $X(t)$ is the current position vector at iteration t, $X^*(t)$ is the best position obtained so far, "·" is an element-by-element multiplication. The coefficient **A**, **C** are determined from the formulas:

$$A = 2a \cdot r_1 - a \tag{22}$$

$$C = 2r_2 \tag{23}$$

where components of **a** are linearly decreased from 2 to 0 through iterations and r_1, r_2 are random vectors in $[0, 1]$.

2.7.2. Bubble-Net Attacking (Exploitation Phase)

In this phase, humpback whales swim around the prey within a helix-shaped path. To model this behavior, it is assumed that there is a 50% chance to choose between the shrinking encircling or spiral movements [33]:

$$\mathbf{X}(t+1) = \begin{cases} \mathbf{X}^*(t) - \mathbf{A} \cdot \mathbf{D} & \text{if } p < 0.5 \\ \mathbf{D}^* \cdot \exp(bk) \cdot \cos(2\pi k) + \mathbf{X}^*(t) & \text{if } p \geq 0.5 \end{cases} \tag{24}$$

where $\mathbf{D}^* = |\mathbf{X}^*(t) - \mathbf{X}(t)|$ is the distance of the ith whale to the prey, b defines the shape of the spiral, k is a random number in $[-1, 1]$, and p is a random number in $[0, 1]$.

2.7.3. Search for Prey (Exploration Phase)

In this phase, humpback whales search the pray according to the position of each other. The location of a search agent is calculated according to randomly selected search agent rather than the best search agent as in the exploitation phase. The mathematical model is determined as follows [33]:

$$\mathbf{X}(t+1) = \mathbf{X}_{rand}(t) - \mathbf{A} \cdot \mathbf{D} \tag{25}$$

where

$$\mathbf{D} = |\mathbf{C} \cdot \mathbf{X}_{rand}(t) - \mathbf{X}(t)| \tag{26}$$

and $\mathbf{X}_{rand}$ is the random position vector chosen from the current population.

2.8. Integration of Swarm Intelligence Techniques with Regularized Discriminant Analysis

The idea of integrating swarm intelligence methods with the RDA classifier is shown in Figure 3. This figure presents in the form of a block diagram main stages of optimization of the RDA model using swarm algorithms and the method of determining the objective function. The task of particle swarm optimization, grey wolf optimization, or whale optimization algorithm is to select the RDA parameters, which are [30]:

- $w_1, w_2, \ldots, w_n$ — the observation weights,
- δ — the linear coefficient threshold,
- γ — the parameter for regularizing the covariance matrix of the predictors,

where n is the number of observations in the training set. For this purpose, the observation weights and two hyperparameters δ, γ are placed as elements of the agent (candidate) vector, which has the form

$$\boxed{\; w_1 \mid w_2 \mid \ldots \mid w_n \mid \delta \mid \gamma \;} \tag{27}$$

At the beginning of the algorithm, the agents are initialized. In the next step, the value of the objective function for all agents is calculated. Then, the stop condition is checked, if it is not reached, the agents are updated and the value of the objective function is recalculated. The swarm optimization algorithm generates many hypothetical solutions (which are represented by agents) and the best solution is selected in the optimization process. These operations are repeated until the stop condition is reached. In the objective function block, the RDA model is determined for the parameters proposed by the agent and for the training data. The output of this model is then calculated on the validation data and the value of the objective function is determined based on formula (1). The weights $w_1, w_2, \ldots, w_n$, hyperparameters δ and γ of the RDA classifier are limited during optimization in given ranges (see Section 3). The algorithm returns the optimal parameters of the RDA classifier in the best agent vector. This result is included in the block 'Return the global best solution' in Figure 3. On this basis, the objective function value for the test data are calculated.
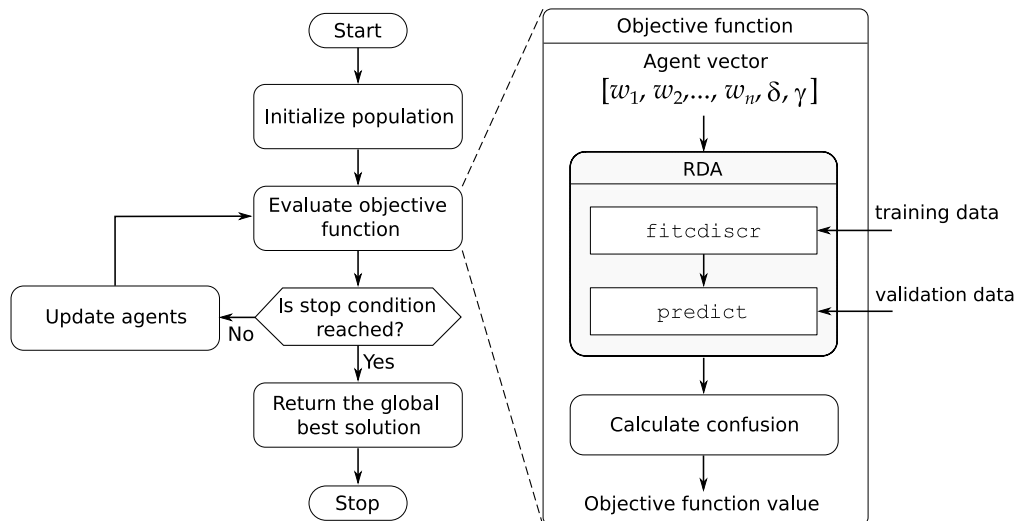
Figure 3. The idea of integration of swarm intelligence techniques with the RDA.

All the proposed hybrid methods have been implemented in Matlab equipped with additional toolboxes. Using the function `fitcdiscr` from the Statistics and Machine Learning Toolbox [30], the RDA classifier model is created, while the function `predict` from the same package is used to determine class predictions (Figure 3). The PSO method has been implemented using the function `particleswarm` from the Global Optimization Toolbox [34] and the GWO and WOA methods using software developed by Mirjalili [32,33].

3. Results

The division of the gait dataset into training, validation, and test sets in four experiments is presented in Table 1. In the first experiment (Set #1) and the fourth experiment (Set #4), all persons in the collection were wearing clothes number 1. In the second experiment (Set #2), persons in the training and validation sets were in clothing number 1, while in the test set in clothing number 2. In the third experiment (Set #3), persons in the training and validation sets were in clothing number 1, while in the test set they wore a backpack. The number of identities in training/validation/test subsets was: in Set #1—32/32/32, Set #2—32/32/6, Set #3—32/32/7, and Set #4—32/32/32. The samples were not repeated in the subsets. The procedure of separating the samples from the training set to the validation set was as follows: for Sets #1 and #4, one sample was taken for each class (the last sample was always taken), for Sets #2 and #3, two samples were taken for each class because there were more training data (the last two samples for each class were taken). Experiments 1–3 were performed in such a way that the classification model and its testing error were determined 10 times, and then the average of the results was calculated. In the fourth experiment (Set #4), the 10-fold cross-validation was performed to obtain an average score. In this method, the original dataset is partitioned into 10 equal size subsets. A single subset is retained as the test data, and the remaining nine subsets are used as training data. The cross-validation process is repeated 10 times (the folds) for the test data, and the 10 results are averaged. In the RDA-PSO, RDA-GWO, and RDA-WOA methods, the number of agents was equal to 30, and the number of iterations was equal to 25. These parameters of optimization techniques were selected experimentally. The weights of the observations were limited to the range $[10^{-8}, 1]$, while the hyperparameters δ and γ were limited to the range $[0, 1]$.

Table 1. Division of the gait dataset into training, validation, and test sets.

Experiment	Subset	Classical Methods	Hybrid Methods
1: Set #1	train	169	137
	validation	–	32
	test	156	156
2: Set #2	train	325	261
	validation	–	64
	test	58	58
3: Set #3	train	325	261
	validation	–	64
	test	31	31
4: Set #4	train	90% ($\approx$ 293)	80% ($\approx$ 261)
	validation	–	10% ($\approx$ 32)
	test	10% ($\approx$ 32)	10% ($\approx$ 32)

Table 2 contains the correct classified ratio of the gait recognition for the considered methods. These are four classical methods (kNN [35], NB [35], support vector machines with sequential minimal optimization (SMO) [36], and MLP [35]) taken from [21], linear discriminant analysis (non-regularized) [27], and three proposed hybrid methods. Figure 4 presents the confusion matrices for the best models in the considered experiments. These matrices show the percentage of correct class recognition by the arrangement of the colored elements. The closer the color is to dark red on the diagonal, the more accurate the class recognition. The colors changing from dark red to white mean worse and worse recognition.

Table 2. Correct classified ratio [%].

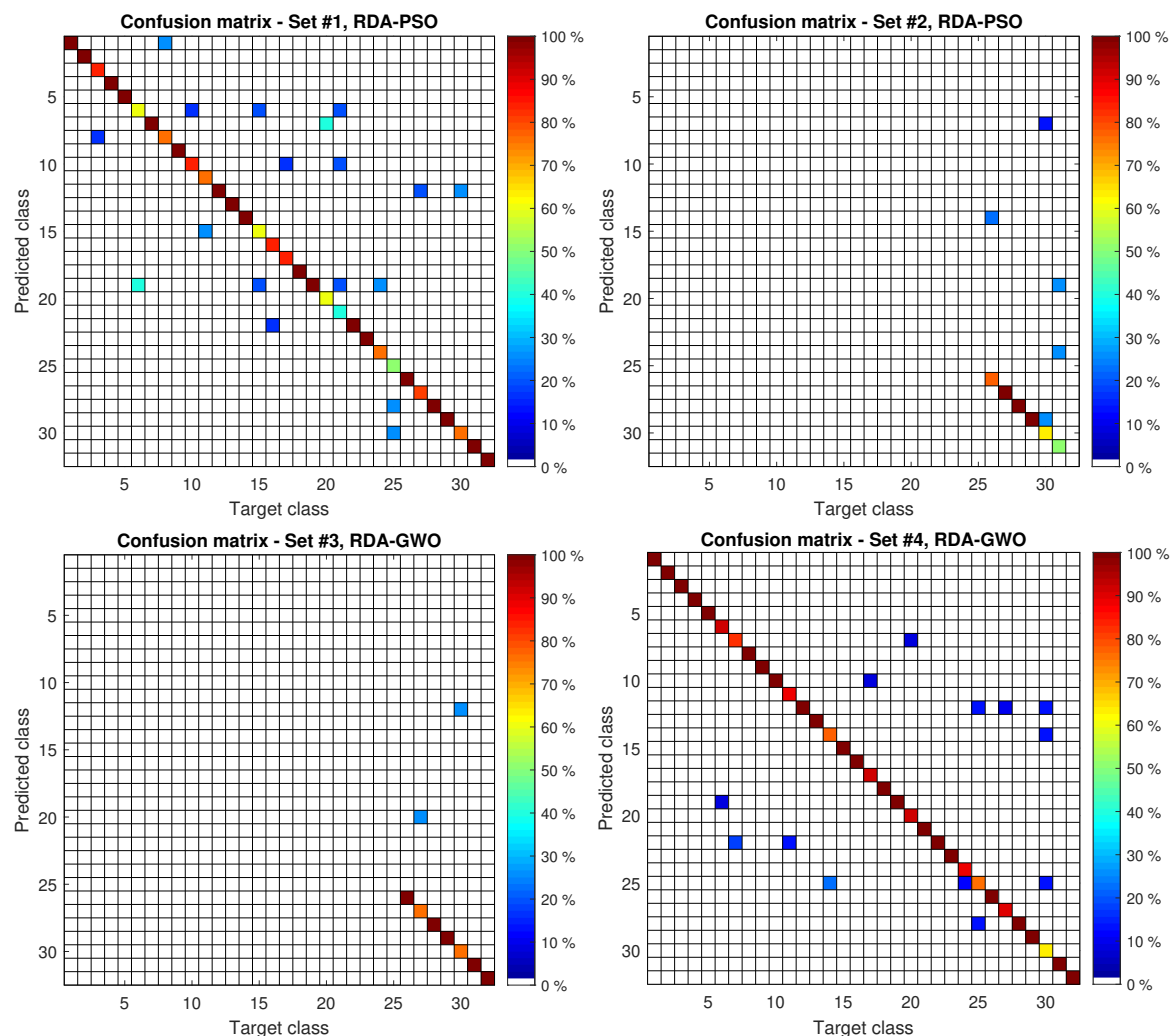| Experiment	kNN [21]	NB [21]	SMO [21]	MLP [21]	LDA	RDA-PSO	RDA-GWO	RDA-WOA
1: Set #1	47.44	55.77	67.95	80.13	45.51	**87.05**	86.28	86.92
2: Set #2	37.93	56.90	63.79	75.86	79.31	**85.34**	84.48	84.48
3: Set #3	38.71	70.97	67.74	77.42	**93.55**	88.39	**93.55**	**93.55**
4: Set #4	56.92	79.69	84.31	89.85	91.99	95.07	**95.39**	95.09

The best result is marked in bold font.

Figure 4. Confusion matrices for best models in each experiment.

4. Discussion

To compare the results, the classical methods taken from the paper [21] and the LDA were considered. This comparison with the methods proposed by the authors (RDA-PSO, RDA-GWO, RDA-WOA) is provided in Table 2. It can be seen that, in Experiment 1 and Experiment 2, the RDA-PSO method proposed by the authors has the highest CCR index. In Experiment 3, which considered clothing with a backpack, three of the analyzed methods (LDA, RDA-GWO, RDA-WOA) achieved the same CCR at the level of 94%. In the final fourth experiment, the method proposed by the authors was again the best, in this case combining RDA with GWO. It should also be noted that the LDA obtained the worst result of all methods for Experiment 1; this is most likely due to insufficient training data. However, the proposed methods obtained very good results for this experiment, at the level of about 86–87%. The use of the regularized discriminant analysis combined with swarm intelligence techniques improved the quality of gait recognition. The developed methods outperformed the classical methods and improved the recognition results achieved by the best of them (MLP) by 6 to 16%, depending on the experiment. When comparing the proposed hybrid methods between each other, it should be noted that the results are inconclusive and it seems that these methods are equivalent in this application.

When analyzing the confusion matrix shown in Figure 4, it can be seen that most people are recognized with high accuracy, but there are classes with which the methods have difficulties. For example, in Experiment 1 (RDA-PSO method) for classes 6, 15, 20, 21, and 25, the recognition efficiency drops to 40–60%. It is most likely caused by too little training data (about five gait cycles for each class).

When 10-fold cross-validation is used (about nine gait cycles for each class), the recognition efficiency increases significantly. It should also be noted that the data used in experiments were recorded with a markerless motion capture system, which is not perfect and generates noise [21]. It certainly has an impact on the achieved results. For Experiment 2, for which learning and testing were performed on the sequences for which the clothes were changed, a significant deterioration in the results for persons #30 and #31 can be observed. When analyzing these video sequences, it can be noticed that the heavy shoes changed to sandals. This could be the cause of the deterioration in the results in this experiment. On the other hand, for person #29, there was a change of footwear from sports shoes to shoes with a heel, and still 100% detection rate was achieved.

5. Conclusions

The hybrid methods that combine regularized discriminant analysis and swarm intelligence techniques for gait recognition have been proposed. In the presented approach, particle swarm optimization, grey wolf optimization, and whale optimization algorithm are used. These techniques optimize the observation weights and hyperparameters of the regularized discriminant analysis. The proposed methods were compared with five methods found in the literature. In the learning process, the confusion value was used as an objective function. The conducted experiments on the GPJATK dataset proved the validity of the proposed concept. Future work will focus on improving the proposed concept by replacing the MPCA method with another method of dimensionality reduction. Moreover, some work will be carried out to add new features to gait signatures.

References

1. Jain, A.K.; Flynn, P.; Ross, A.A. *Handbook of Biometrics*; Springer US: Boston, MA, USA, 2008; doi:10.1007/978-0-387-71041-9.
2. Matovski, D.S.; Nixon, M.S.; Mahmoodi, S.; Carter, J.N. The Effect of Time on Gait Recognition Performance. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 543–552, doi:10.1109/TIFS.2011.2176118.
3. Wan, C.; Wang, L.; Phoha, V.V. A survey on gait recognition. *ACM Comput. Surv.* **2018**, *51*, doi:10.1145/3230633.
4. Little, J.J.; Boyd, J.E. Recognizing People by Their Gait : The Shape of Motion. *J. Comput. Vis. Res.* **1998**, *1*, 1–33.
5. Han, J.; Bhanu, B. Individual recognition using gait energy image. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 316–322.
6. Kusakunniran, W.; Wu, Q.; Zhang, J.; Li, H. Support vector regression for multi-view gait recognition based on local motion feature selection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 974–981, doi:10.1109/CVPR.2010.5540113.
7. Li, X.; Chen, Y. Gait recognition based on structural Gait energy image. *J. Comput. Inf. Syst.* **2013**, *9*, 121–126.
8. Mohan Kumar, H.P.; Nagendraswamy, H.S. LBP for gait recognition: A symbolic approach based on GEI plus RBL of GEI. In Proceedings of the 2014 International Conference on Electronics and Communication Systems (ICECS), Prague, Czech Republic, 2–4 April 2014; pp. 1–5.
9. Wang, H.; Fan, Y.; Fang, B.; Dai, S. Generalized linear discriminant analysis based on euclidean norm for gait recognition. *Int. J. Mach. Learn. Cybern.* **2018**, doi:10.1007/s13042-016-0540-0.
10. Lishani, A.O.; Boubchir, L.; Khalifa, E.; Bouridane, A. Human gait recognition using GEI-based local multi-scale feature descriptors. *Multimed. Tools Appl.* **2019**, *78*, 5715–5730, doi:10.1007/s11042-018-5752-8.

11. Chao, H.; He, Y.; Zhang, J.; Feng, J. GaitSet: Regarding gait as a set for cross-view gait recognition. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Hilton Hawaiian Village, Honolulu, Hawaii, USA, 27 January–1 February 2019; pp. 8126–8133, doi:10.1609/aaai.v33i01.33018126.

12. Guo, H.; Li, B.; Zhang, Y.; Zhang, Y.; Li, W.; Qiao, F.; Rong, X.; Zhou, S. Gait Recognition Based on the Feature Extraction of Gabor Filter and Linear Discriminant Analysis and Improved Local Coupled Extreme Learning Machine. *Math. Probl. Eng.* **2020**, *2020*, doi:10.1155/2020/5393058.

13. BenAbdelkader, C.; Cutler, R.; Davis, L. Stride and cadence as a biometric in automatic person identification and verification. In Proceedings of the 5th IEEE International Conference on Automatic Face Gesture Recognition, FGR 2002, Washington, DC, USA, 21–21 May 2002; pp. 372–377, doi:10.1109/AFGR.2002.1004182.

14. Yam, C.Y.Y.; Nixon, M.S.; Carter, J.N. Automated person recognition by walking and running via model-based approaches. *Pattern Recognit.* **2004**, *37*, 1057–1072, doi:10.1016/j.patcog.2003.09.012.

15. Bouchrika, I.; Nixon, M.S. Model-based feature extraction for gait analysis and recognition. *Lect. Notes Comput. Sci.* **2007**, *4418 LNCS*, 150–160, doi:10.1007/978-3-540-71457-6_14.

16. Ng, H.; Ton, H.L.; Tan, W.H.; Yap, T.T.V.; Chong, P.F.; Abdullah, J. Human Identification Based on Extracted Gait Features. *Int. J. New Comput. Archit. Their Appl. (IJNCAA)* **2011**, *1*, 358–370.

17. Ariyanto, G.; Nixon, M.S. Model-based 3D gait biometrics. In Proceedings of the International Joint Conference on Biometrics (IJCB), Washington, DC, USA, 11–13 October 2011; pp. 1–7, doi:10.1109/IJCB.2011.6117582.

18. Deng, M.; Wang, C.; Cheng, F.; Zeng, W. Fusion of spatial-temporal and kinematic features for gait recognition with deterministic learning. *Pattern Recognit.* **2017**, *67*, 186–200, doi:10.1016/j.patcog.2017.02.014.

19. Switonski, A.; Krzeszowski, T.; Josinski, H.; Kwolek, B.; Wojciechowski, K. Gait recognition on the basis of markerless motion tracking and DTW transform. *IET Biom.* **2018**, *7*, 415–422, doi:10.1049/iet-bmt.2017.0134.

20. Kumar, P.; Mukherjee, S.; Saini, R.; Kaushik, P.; Roy, P.P.; Dogra, D.P. Multimodal Gait Recognition with Inertial Sensor Data and Video Using Evolutionary Algorithm. *IEEE Trans. Fuzzy Syst.* **2019**, *27*, 956–965, doi:10.1109/TFUZZ.2018.2870590.

21. Kwolek, B.; Michalczuk, A.; Krzeszowski, T.; Switonski, A.; Josinski, H.; Wojciechowski, K. Calibrated and synchronized multi-view video and motion capture dataset for evaluation of gait recognition. *Multimed. Tools Appl.* **2019**, *78*, 32437–32465, doi:10.1007/s11042-019-07945-y.

22. Liao, R.; Yu, S.; An, W.; Huang, Y. A model-based gait recognition method with body pose and human prior knowledge. *Pattern Recognit.* **2020**, *98*, 107069, doi:10.1016/j.patcog.2019.107069.

23. Balazia, M.; Sojka, P. Gait Recognition from Motion Capture Data. *ACM Trans. Multimed. Comput. Commun. Appl.* **2018**, *14*, doi:10.1145/3152124.

24. Castro, F.; Marín-Jiménez, M.; Mata, N.; Muñoz-Salinas, R. Fisher Motion Descriptor for Multiview Gait Recognition. *Int. J. Pattern Recognit. Artif. Intell.* **2017**, *31*, 1756002, doi:10.1142/S021800141756002X.

25. Kwolek, B.; Krzeszowski, T.; Gagalowicz, A.; Wojciechowski, K.; Josinski, H. Real-Time Multi-view Human Motion Tracking Using Particle Swarm Optimization with Resampling. In *Articulated Motion and Deformable Objects*; Perales, F.J.; Fisher, R.B.; Moeslund, T.B., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 92–101.

26. Lu, H.; Plataniotis, K.N.; Venetsanopoulos, A.N. MPCA: Multilinear Principal Component Analysis of Tensor Objects. *IEEE Trans. Neural Networks* **2008**, *19*, 18–39.

27. Fisher, R.A. The use of multiple measurements in taxonomic problems. *Ann. Eugen.* **1936**, *7*, 179–188, doi:10.1111/j.1469-1809.1936.tb02137.x.

28. Rao, C.R. The Utilization of Multiple Measurements in Problems of Biological Classification. *J. R. Stat. Soc. Ser. (Methodological)* **1948**, *10*, 159–203.

29. Guo, Y.; Hastie, T.; Tibshirani, R. Regularized linear discriminant analysis and its application in microarrays. *Biostatistics* **2006**, *8*, 86–100, doi:10.1093/biostatistics/kxj035.

30. MathWorks. *Statistics and Machine Learning Toolbox: User's Guide*; The MathWorks, Inc., Natick, MA, USA, 2020.

31. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995;Volume 4, pp. 1942–1948.

32. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61, doi:10.1016/j.advengsoft.2013.12.007.

33. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67, doi:10.1016/j.advengsoft.2016.01.008.

34. MathWorks. *Global Optimization Toolbox: User's Guide*; The MathWorks, Inc., Natick, MA, USA, 2020.

35. Bishop, C.M. *Pattern Recognition and Machine Learning*; Information Science and Statistics; Springer: Secaucus, NJ, USA, 2006.

36. Platt, J. *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines*; Technical Report MSR-TR-98-14; Microsoft: Albuquerque, NM, USA, 1998.

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.