# Comparing Dynamic PSO Algorithms for Adapting Classifier Ensembles in Video-Based Face Recognition

Jean-François Connolly, Éric Granger, and Robert Sabourin
Laboratoire d'imagerie, de vision et d'intelligence artificielle
École de technologie supérieure, Université du Québec
1100, rue Notre-Dame Ouest, Montréal, Canada, H3C 1K3
jfconnolly@livia.etsmtl.ca, Eric.Granger@etsmtl.ca, and Robert.Sabourin@etsmtl.ca.

***Abstract*—Biometric models are typically designed a priori using limited number of samples acquired from complex environments that change in time during operations. Therefore, these models are often poor representatives of the biometric trait to be recognized. To circumvent this problem, ensemble of classifiers can be used to integrate solutions obtained from multiple diverse classifiers. In this paper, two dynamic particle swarm optimization (DPSO) algorithms are compared for the evolution of classifier ensembles during supervised incremental learning of newly-acquired data samples in video-based face recognition. Using the properties of these population-based optimization algorithms, an incremental DPSO learning strategy for adaptive classification systems (ACSs) is employed to evolve a pool of fuzzy ARTMAP classifiers while an heterogeneous ensemble is selected through a greedy search process that seeks to maximize both performance and diversity. The performance of dynamic niching PSO (DNPSO) and speciation PSO (SPSO) algorithms is assessed in terms of classification rate, resource requirements and diversity for different incremental learning scenarios of new data blocks extracted from real-world video streams. Simulation results indicate that both DPSO algorithms can efficiently create accurate ensembles while reducing computational complexity. In addition, directly selecting representative subswarm particles to form diversified classifier ensembles significantly reduces the computational complexity.**

## I. Introduction

With the availability of affordable technology, and the wide range of commercial and law enforcement applications, face recognition has received considerable attention in the area of biometrics. Given that face images may be sampled discreetly, without cooperation, face recognition is employed in applications where the process used to acquire biometric samples is difficult to control. Face recognition can thus be used in video surveillance to detect the presence of individuals belonging to a watch list within dense and moving scenes (e.g., screening for criminals or terrorists in an airport setting), identification at access control points, and laptop or cell phone user verification. However, the rapid recognition of individuals from video cameras remains a challenging problem.

A critical function in these applications is the classification of face regions captured from video streams. To save resources during the design phase, face recognition systems typically estimate the biometric model of individuals with statistical or neural network classifiers. With neural networks, for instance, estimation is performed using a given architecture and a set of synaptic weights. In practice, these biometric models are created during an enrollment phase, using samples collected from unknown data distributions, a set of hyperparameters (e.g., learning rate parameter), and a limited amount of learning samples acquired from video streams in *unconstrained* scenes. Due to limited control over operational conditions (illumination, pose, etc.), and physiological changes over time, either temporary (haircut, glasses, etc.) or permanent (scars, aging, etc.), these distributions are complex and with much inter- and intra-class variability. In addition, since collection of data is expensive and time consuming, training samples are often sparse and unbalanced, leading to biometric models that are often poor representative of the biometric trait to be recognized [1].

It is however common to acquire new data at some point in time after the classifier has originally been trained and deployed for operations. Adaptation of video-based face recognition systems should occur during enrollment (new classes are added to the system) and update (pre-existing classes are refined using the new data). For accurate recognition of individuals over time, biometric systems should adapt their models in response to new or changing data samples, features, and classes through *supervised incremental learning*.

In previous work, the authors have proposed an adaptive classification system (ACS) that performs incremental learning of new data in a video-based face recognition application. This system consist of an evolving swarm of fuzzy ARTMAP neural networks (FAM), a dynamic particle swarm optimization (DPSO) module, and a long term memory (LTM) [2]. The authors have also shown that corruption of biometric models, that would occur during incremental learning of new data, can be avoided by using ensembles of classifiers [3].

One key element in the success of multiple classification systems is the use of *diversity measures* when generating a pool of classifiers [4], [5], [6], [7], [8]. When trained on the same data, but with different learning dynamics (i.e., sets of hyperparameters), it is possible to exploit the limited training samples available to create a pool of heterogeneous classifiers [9]. By cooperating and exchanging information, their training is *interdependent*, and each one contributes different decision boundaries, leading to better pools of classifiers [10], [11]. If diversity is also considered to select and combine ensembles from these pools, the accuracy of a face recognition system can be improved [12].

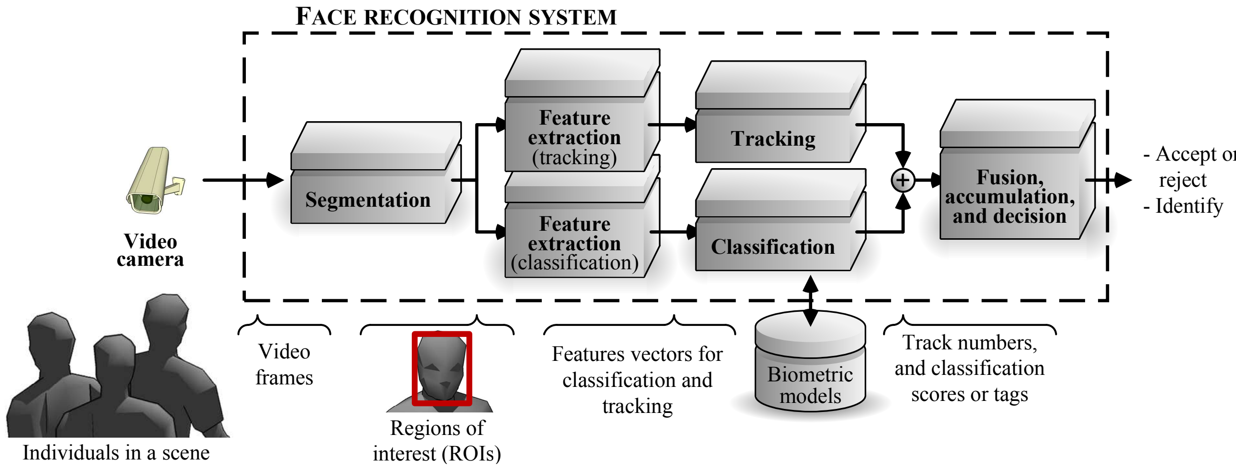



Fig. 1. A general biometric system for video-based face recognition.

In this paper, dynamic particle swarm optimization (DPSO) algorithms are used to generate pools of classifiers and select ensembles during supervised incremental learning of new data in video-based face recognition. A learning strategy based on dynamic particle swarm optimization is first applied to the adaptive classification system (ACS) [3] to evolve a diversified swarm of classifiers to be used as a pool, among which an heterogeneous ensemble is selected through a greedy search process. As each particle in the hyperparameter space corresponds to a classifier, the DPSO-based strategy cojointly optimizes all *classifier parameters* – hyperparameters, synaptic weights, and architecture – of a swarm of FAM networks such as classification rate is maximized. Under the hypothesis of a correlation between diversity in the hyperparameter space and diversity among classifiers in the swarm, the ACS can exploit the DPSO algorithms to find and track several changing local optima in the hyperparameter space to force diversity when generating pools of classifiers, and select ensembles on the basis of performance *and* diversity.

This study compares the performance of different DPSO algorithms used to guide pools of classifiers and ensemble selection. For different ensemble selection methods, performance of this system is assessed in terms of classification rate for video sequences, resource requirements, and diversity during incremental learning of new data blocks from two real-world video data sets. For reference, the performance of FAM, trained with the PSO learning strategy [13], and kNN are given for batch learning.

In the next section, a general biometric system for video-based face recognition is presented. Then, in Section III, a description of the ACS is presented, along with the FAM neural network used for classification, and the DPSO algorithms used for its hyperparameters adaptation. In Section IV, the greedy search process for ensemble selection is described. Then, the data bases, incremental learning scenarios, performance measures and experimental protocol used for proof-of-concept simulations are described in Section V. Finally, experimental results are presented and discussed in Section VI.

## II. Face Recognition in Video

A biometric system for video-based face recognition is depicted in Figure 1. It is assumed that the face recognition system captures a sequence of 2D images, or video frames, from an external 3D scene via one fixed camera. Each frame provides the system with a particular view of individuals occupying the scene, and segmentation is performed to locate and isolate regions of interest (ROIs) corresponding to the faces present on the filmed scene.

Discriminant features are then extracted from the ROIs for both tracking and classification. The tracking features can be the position in the 2D images, speed, acceleration, and track number assigned to each ROI on the scene. On the other hand, the classification module will require invariant and discriminant classification features extracted from the ROIs using several techniques available [14], and mapped to an $\mathbb{R}^I$ input feature space. Classification is done with either statistical or neural classifier that maps the $\mathbb{R}^I$ input feature space to a set of $K$ predefined class labels $\Omega = \{C_1, C_2, ..., C_K\}$, where each class $k$ ($k = 1, ..., K$) corresponds to the face model of an individual enrolled in the biometric system.

While the goal of the tracking module is only to follow face(s) across video frames, the classification module has the most complex task of identifying face(s) on a given frame. In this case, classification is done by the adaptive classification system presented in Section III. For each frame, the fusion, accumulation, and decision module combines and accumulates responses from the tracking and classification modules to provide a video-based classification score. With FAM networks, each ROI prediction consists in a binary vector with one for the predicted class, and zero elsewhere. After a given number of video frames, prediction is the class with the highest accumulated response. A decision is then made according the application itself. For verification applications, authenticity is accepted or rejected, while for identification and surveillance applications, a list of the most likely or of all possible matching identities is provided. A review of video-based face recognition systems is given in [2].

## III. Adaptive Classification System

Figure 2 depicts the adaptive classification system (ACS) for incremental learning of new data that replaces the classification module of Figure 1 [2]. This system is composed of a swarm (pool) of classifiers, each one suitable for incremental learning, a dynamic optimization module that adjusts each classifier's hyperparameters, and a long term memory (LTM) that stores and manages incoming data.

When a new block of learning data $D_t$ becomes available to the system at a discrete time $t$, it is employed to update the LTM and evolve a swarm of incremental classifiers. As each classifier is associated to a particle in the hyperparameter space, a dynamic optimization module using a DPSO-based learning strategy cojointly determines the *classifier parameters* – the vector of hyperparameters $\mathbf{h}$,[1] synaptic weights, and architecture – such that classification rate is maximized [3]. At this stage, distributed computer architectures can be used to exploit the parallel nature of evolving a swarm of classifiers and perform high level parallelization. Once the optimization process is over, the fusion module acts according to the method presented in Section IV and creates an heterogeneous ensemble by selecting and combining classifiers from the swarm. It then fuses the ensemble's classifiers to provide an hypothesis that can be updated at time $t+1$.

During evolution of the networks, the LTM ensures that a representative set of all class distributions is always used for the validation process, when training FAM networks, and fitness estimation on the objective function [2]. It manages the learning data from $D_t$ to create a training data set $D_t^{\text{t}}$, validation data set $D_t^{\text{v}}$, and fitness estimation data set $D_t^{\text{f}}$.

In this paper, the fuzzy ARTMAP neural network (FAM) [15] is employed as an incremental learning classifier while optimization is done with two dynamic particle swarm optimization (DPSO) algorithms. The rest of this section provides additional details on the FAM and DPSO algorithms used to evolve a swarm of networks and select ensembles.

### A. Fuzzy ARTMAP Neural Networks

ARTMAP refers to a family of self-organizing neural network architectures that is capable of fast, stable, on-line, unsupervised or supervised, incremental learning, classification, and prediction. The fuzzy ARTMAP (FAM) consists of three layers: (1) an input layer $F_1$ of $2I$ neurons (for a $\mathbb{R}^I$ features space), (2) a hidden competitive layer $F_2$, where each node is associated to an hyperbox in the features space, and (3) a map field of $K$ neurons (the number of classes) [15]. In supervised training mode, ARTMAP classifiers learn an arbitrary mapping between training set patterns $\mathbf{a} = (a_1, a_2, ..., a_I)$ and their corresponding binary supervision patterns $\mathbf{c} = (c_1, c_2, ..., c_K)$. It does so by adjusting its synaptic weights according to the training vectors to (1) create and grow hyper-rectangles in the $\mathbb{R}^I$ features space, and (2) associate those hyper-rectangles to the respective class of each training sample.

[1] Let $\mathbf{h}$ be a vector of hyperparameters that set classifier dynamics. For fuzzy ARTMAP, it is composed of the four hyperparameters $\mathbf{h} = (\alpha, \beta, \epsilon, \bar{\rho})$ described in Section III-A.

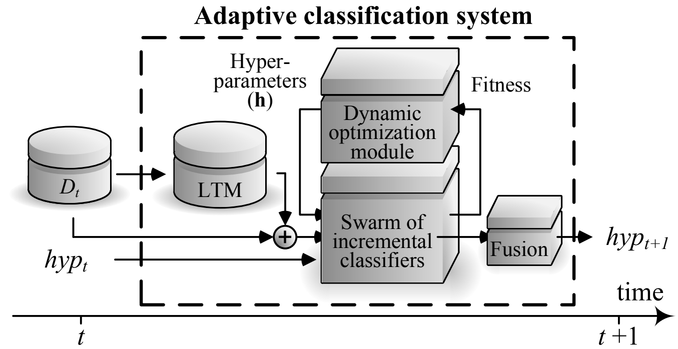


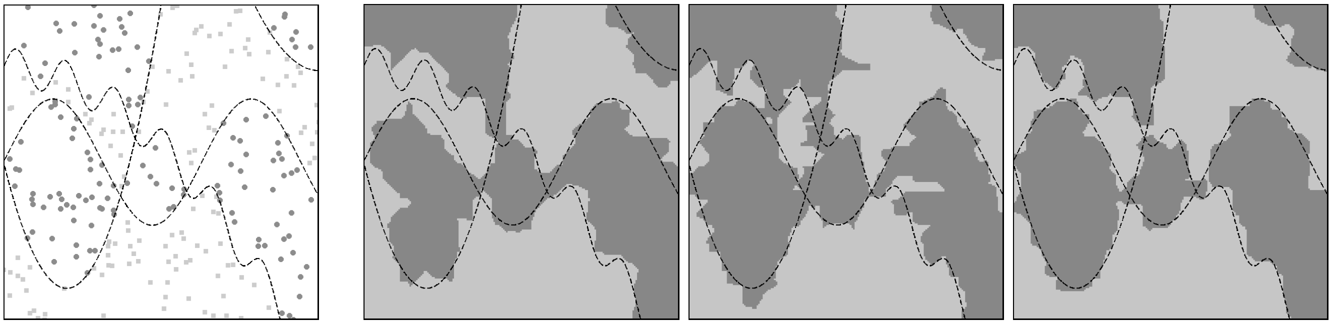

Fig. 2. Evolution of the adaptive classification system (ACS) during a generic incremental learning scenario where new blocks of data are used to update a swarm of classifiers. Let $D_t$, $t \in \mathbb{N}_1$, be blocks of training data available at different instants in time. The ACS starts with an initial hypothesis $hyp_0$ according to the prior knowledge of the domain. Each hypothesis $hyp_t$ are updated to $hyp_{t+1}$ by the ACS on the basis of a new data blocks $D_t$.

During training and testing, FAM internal dynamics are governed by the vector of four hyperparameters $\mathbf{h} = (\alpha, \beta, \epsilon, \bar{\rho})$: the choice parameter $\alpha \geq 0$, the learning parameter $\beta \in [0, 1]$, the match tracking parameter $\epsilon \in [-1, 1]$, and the baseline vigilance parameter $\bar{\rho} \in [0, 1]$. These hyperparameters are inter-related and each has a distinct impact on network dynamics. As shown in Figure 3 on the P2 synthetic 2D data base [9], using different hyperparameters to train a FAM network on the same data results in different decision boundaries. By adjusting these hyperparameters, it is then possible to adapt learning dynamics of the FAM networks over time. That is, to adapt creation and growth of the hyper-rectangles so that modelization of the training patterns available in the current data block $D_t$ is optimal. With the DPSO-based learning strategy, it is possible to exploit diversity in the hyperparameter space to obtain a pool of diversified solutions using classifiers such as FAM. A standard vector of hyperparameters ($\alpha = 0.001$, $\beta = 1$, $\epsilon = 0.001$, $\bar{\rho} = 0$) is commonly used to minimize network complexity [15].

### B. Dynamic Particle Swarm Optimization

Particle swarm optimization (PSO) is a population-based stochastic optimization technique that is inspired by social behavior of bird flocking or fish schooling. With PSO, each particle corresponds to a single solution in the optimization space, and the population of particles is called a swarm. Particles move through the optimization space and change their course under the guidance of a cognitive influence (i.e., their own previous search experience) and a social influence (i.e., their neighborhood previous search experience). Unlike evolutionary algorithms (such as genetic algorithms), each particle always keeps in memory its best position and the best position of its surrounding.

During supervised incremental learning of newly available data blocks $D_t$, particles move in an optimization space defined by the four FAM hyperparameters. The ACS optimization module (Figure 2) will seek to find the vector $\mathbf{h} = (\alpha, \beta, \epsilon, \bar{\rho})$ that maximizes FAM classification rate.



(a) Original data

(b) FAM decision boundaries

Fig. 3. Training data (3a) from the P2 synthetic data base [9], and decision boundaries for FAM trained with different hyperparameters (3b).

As data is added in time, it has been shown that this adaptation corresponds to a dynamic optimization problem such as *maximize* $\left\{ f(\mathbf{h}, t) \mid \mathbf{h} \in \mathbb{R}^4, t \in \mathbb{N}_1 \right\}$ where the fitness, $f(\mathbf{h}, t)$, is the classification rate of FAM trained with the vector of hyperparameters $\mathbf{h}$ over the data set $D_t$ at a discreet time $t$ [2]. Hence, a dynamic version of the original PSO algorithm is needed to properly track optimal hyperparameters through time.

Originally developed for static optimization problems, PSO algorithms have been adapted for dynamic optimization problems by adding mechanisms that (1) modify the social influence to maintain diversity in the optimization space and detect several optima, (2) detect changes in the objective function by using the memory of certain particles, and (3) adapt the memory of the population if change is indeed detected. A review of the common dynamic particle swarm optimization (DPSO) algorithms is presented [16].

In this paper, two DPSO algorithms are compared: a Dynamical Niching PSO (DNPSO) [17], and modified Speciation PSO (SPSO) [16]. With the moving peaks benchmark, both algorithms have been shown to detect local optima and converge toward the global maximum in a multimodal type III optimization environment, where both location and value of optimal points change in time [16], [17].

DNPSO maintains diversity in the hyperparameter space by (1) using a local neighborhood topology, in which subswarms are dynamically created around *local best* particles, that are their own best position among their neighborhood, (2) allowing free particles that do not belong to any subswarms, to move independently, and (3) reinitializing those free particles when they exhibit low velocities (meaning that they have converged on a non-optimal position). The advantages of niching is that subswarms can only be located around local optima, and the main parameter that needs adjustment, the size of the neighborhood, is not problem dependent.

On the other hand, SPSO also uses a local neighborhood topology to group subswarms around local best particles. It first sorts particles by fitness and, in descending order from the global best particle, it defines the local bests by the particles that have the best fitness and are outside the minimal range of other local bests. While this method is heavily dependent on the minimal distance between local best particles, it enable subswarms to be created elsewhere than around local optima. The modified SPSO also uses an anticonvergence mechanism that reinitializes particles from the least fitness subswarm, and a quantum cloud resampling procedure that randomly reposition particles around the center of their respective subswarms, thus maintaining particle diversity in each subswarms.

Each algorithm have their own mechanism to track local optima of dynamic objective functions $f(\mathbf{h}, t)$. For an ACS, changes in the objective function may only occur when new data blocks $D_t$ become available. Out of simplicity, the fitness of the best particle positions is only updated when a new $D_t$ is presented to the system, *before* the iterative DPSO process.

## IV. Heterogeneous Ensemble of FAM Networks

After convergence of the DPSO learning strategy [2], the proposed Algorithm 1 is applied to the fusion module presented in Figure 2. It selects networks among the swarm to form diversified and accurate heterogeneous ensembles, where each network is trained on the same data, but with different hyperparameters [9]. Again, as Figure 3 illustrates, this method is based on the hypothesis that particle diversity in the hyperparameter space is reflected on diversity of the FAM networks associated with those particles.

The proposed method uses information about the particle swarm to properly select good heterogeneous ensembles based on the networks accuracy and diversity. However, it does so by avoiding computing costly classifier diversity indicators [4], [5], [6], [7], [8]. With FAM networks, those indicators involve evaluating predictions of all the networks over the fitness data sets ($D_t^{\mathrm{f}}$). Instead, it uses a much less complex approach: it computes diversity in the hyperparameter space with the matrix of Euclidean distances between each particle personal best position.

Before any selection occurs, the ensemble *EoFAM* is empty (line 1). The first selection occurs on lines 2–4, where the ensemble is initially made of the networks corresponding to detected local optima (i.e. personal best position of the DPSO local best particles). This ensures that the ensemble is made of the most accurate networks in the swarm. Since most DPSO algorithms force a minimal distance between local bests, this initial selection also insures ensemble diversity.

**Algorithm 1** Heterogeneous ensemble selection technique.

**Inputs:** A swarm of $N$ fuzzy ARTMAP networks associated with $N$ DPSO particles.

**Outputs:** An heterogeneous ensemble *EoFAM* based on accuracy and diversity.

**Initialization:**

1: *EoFAM* $\leftarrow \emptyset$

**Selection of FAM networks associated to local bests:**

2: **for** $e = 1$ to $N_{ss}$, the number of subswarms **do**

3: Add the network associated to the local best particle $e$

4: **end for**

**Second selection aimed to maximize diversity using greedy search:**

5: Compute initial swarm diversity $\overline{\delta_{e_1 e_2}}(N_{ss})$ for the $N_{ss}$ networks in ensemble $E$.

6: **for** $e = 1$ to $(N - N_{ss} + 1)$ **do**

7: **for** all networks that are not part of the ensemble **do**

8: Find the one that maximizes current swarm diversity $\overline{\delta_{e_1 e_2}}(N_{ss} + e)$.

9: **end for**

10: **if** there exist no network such as $\overline{\delta_{e_1 e_2}}(N_{ss} + e) > \overline{\delta_{e_1 e_2}}(N_{ss} + e - 1)$ **then**

11: BREAK;

12: **else**

13: Add the network that maximized the most particle diversity to the ensemble.

14: **end if**

15: **end for**

The second selection then seeks to maximize particle diversity. For two classifiers $e_1$ and $e_2$, the pairwise diversity between their particles, $\delta_{e_1 e_2}$, is the Euclidean distance in the hyperparameter space between those particles. Ensemble diversity in the hyperparameter space, $\overline{\delta_{e_1 e_2}}$, is then defined by the average value of all Euclidean distances between particles. Although this particle diversity measure is the most complex, it was revealed to be the most accurate [18]. Still, compared to all neural network training needed for fitness estimation on the objective function, computing $\overline{\delta_{e_1 e_2}}$ remains a small part of the overall learning process.

Ensemble diversity obtained after the first selection is computed (line 5) and the greedy search takes place (lines 6–15). Algorithm 1 iteratively scans through all particles associated to fuzzy ARTMAP networks that are not part of the ensemble to find which one maximizes swarm diversity (lines 7–9). If there exist no particle that can raise diversity, Algorithm 1 stops (line 11). Otherwise, the network associated to the winning particle is added to the ensemble (13).

Once ensemble selection is completed, prediction is done using a simple majority vote. In the case of a tie, the class predicted by the smallest networks (the ones that yielded the fewer $F_2$ nodes) is declared winner of the vote.

## V. Experimental Methodology

### A. Video Databases

In order to illustrate the performance of generating pools of classifiers and selecting ensembles with different DPSO algorithms during supervised incremental learning in video-based face recognition, proof-of-concept simulations are performed with two real-world video data bases. In both cases, segmentation is performed using the Viola-Jones algorithm included in the OpenCV computer vision library. It produces regions of interest (ROIs) that are converted in gray scale and normalized to $24 \times 24$ images, where the eyes are aligned horizontally with a distance of 12 pixels between them. Principal Component Analysis is then used to extract the least 64 uncorrelated features that are normalized using the min-max technique to accommodate the FAM neural network.

The first data base was collected by the Institute for Information Technology of the Canadian National Research Council (IIT-NRC) [19]. It is composed of 22 video sequences captured from eleven individuals positioned in front of a computer. For each individual, two color video sequences of about fifteen seconds are captured at a rate of 20 frames per seconds with an Intel web cam of a $160 \times 120$ resolution that was mounted on a computer monitor. Of the two video sequences, one is dedicated to training (1527 ROIs) and the other to testing (1585 ROIs). The number of ROIs detected varies from class to class, ranging from 40 to 190 for each video sequences. This data base contains a variety of challenging operational conditions such as different facial orientations and expressions, occlusion, and low resolution.

The second video data base is called Motion of Body (MoBo), and was collected at Carnegie Mellon University under the HumanID project [20]. Each video sequence shows one of 25 different individuals under six viewpoints, and walking on a tread-mill so that they move their heads naturally to four different motion types. Only the video sequences with visible faces were kept: full frontal view and both sides with an angle of about $45°$ from the full frontal view. This data base was reduced in order to be roughly the same size of the IIT-NRC data base, while having, for each individual, the same number of ROIs from each motion types and camera angle. Data from 10 individuals was employed, with 288 ROIs per class (the first 24 ROIs detected for each type of walk and camera angle) for a total of 2880 patterns. The data base was divided into a learning and test data sets of 1440 patterns each. For each type of walk and camera angle, the first 12 of the 24 RIOs were assign to the learning data set, while the last 12 were assign to the test data set.

### B. Incremental Learning Scenarios

Prior computer simulations, each video data set is divided in blocks of data $D_t$ to emulate the availability of $T$ successive blocks of learning data to the ACS. Supervised incremental learning is performed according to two different scenarios.

**Enrollment –** In this scenario, each block contains ROIs of individuals that are not enrolled to the system. Classes are added incrementally to the system, one at a time. To assess ACS performance for $K$ classes, the first learning block $D_1$ is composed of two classes, and each successive block $D_t$, where $2 \leq t \leq K-1$, contains the ROIs captured in a video sequence corresponding to an individual that has not previously been enrolled to the system. For each $D_t$, performance is only evaluated for existing classes. To insure invariance of results to class presentation order, this experiment is performed using five different random *class* presentation orders.

**Update –** In this scenario, each block contains ROIs of individuals that have previously been enrolled to the system. It is assumed that at a given time, the ROIs of an individual is captured in a video sequence, and then learned by the system to refined its internal models. To assess ACS performance, all classes are initially learned with the first data block $D_1$ and are updated one class at a time with blocks $D_2$ through $D_{K+1}$. In order to better observe cases where classes are not initially well defined, block $D_1$ is composed of 10% of the data for each class, and each subsequent block $D_t$, where $2 \leq t \leq K+1$, is composed of the remaining 90% of one specific class. Here again, invariance to class order presentation is insured by repeating this experimentation with five different *class* presentation orders.

### C. Experimental Protocol

The DNPSO and SPSO parameters used for both experiments are shown in Table I. Weight values $\{w_1, w_2\}$ are defined as proposed in [21]. To detect a maximal number of local optima, all constraints regarding the number of the subswarms are lifted. Since distances between particles are measured during both DPSO algorithms, the swarm evolves in a *normalized* $\mathbb{R}^4$ hyperparameter space to avoid any bias due to the domain of each hyperparameter. Before being applied to FAM, particle positions are denormalized to fit each hyperparameter's domain. For each learning data block $D_t$, the DPSO optimization process is set to either stop after 10 iterations without improvement of the best FAM network ($FAM_{n^*,t}$) classification rate, or after 100 iterations.

Learning is performed over ten trials using ten folds cross-validation with the LTM used as specied in [2]. Within each replication, there are five different trials using different class presentation order, for a total of fifty replications. Out of the ten folds, eight are dedicated to training ($D_t^{\mathrm{t}}$), one fold is combined with half of LTM to validate and determine the number of FAM training epochs ($D_t^{\mathrm{v}}$), and the remaining fold is combined with the other half of LTM to estimate the fitness of each particle during the DPSO algorithm ($D_t^{\mathrm{f}}$). In this experiment, initialization and update of the LTM is performed with $p_D = 1/6$ and $|LTM|_k = 20$.

Performance of the proposed heterogeneous ensemble selection algorithm for incremental learning, that is, the ACS with (1) an ensemble of FAM networks selected with Algorithm 1 ($EoLB_t^{\mathrm{+d}}$), is evaluated using both DNPSO and SPSO. This system is also compared to the ACS with different ensemble selection techniques during supervised incremental learning of data blocks $D_t$ using: (2) an ensemble of FAM networks determined by an accuracy-based greedy search [8] ($EoFAM_t^{\mathrm{+a}}$), (3) an ensemble of FAM networks build with the whole swarm ($EoFAM_t^{\mathrm{all}}$), and (4) the FAM network corresponding to the global best solution ($FAM_{g,t}$).

For reference, the performances of FAM, trained with the PSO learning strategy ($EoFAM_t^{\mathrm{all\text{-}B}}$), and kNN (*kNN*) are given for batch learning. At a time $t$, batch learning consist of initializing the system, and learning all the data blocks $D_t$ accumulated thus far, $B_t = D_1 \cup ... \cup D_t$ [13].

TABLE I
DNPSO AND SPSO PARAMETERS

| Parameter | Value |
|---|---|
| **DNPSO** | |
| Swarm's size $N$ | 40 |
| Weights $\{w_1, w_2\}$ | $\{0.73, 2.9\}$ |
| Maximal number of subswarms | $\infty$ |
| Maximal size of each subswarm | 4 |
| Minimal distance between two local bests | 0.1 |
| Neighborhood size | 5 |
| Minimal velocities of free particles | 0.01 |
| **SPSO** | |
| Swarm's size $N$ | 40 |
| Weights $\{w_1, w_2\}$ | $\{0.73, 2.9\}$ |
| Maximal number of subswarms | $\infty$ |
| Maximal size of each subswarm | 4 |
| Minimal distance between two local bests | 0.6 |

### D. Performance Evaluation and Diversity Indicator

The average performance of FAM is assessed in terms of classification rate and resources requirements, measured by compression. In video-based face recognition applications, *video-based classification rate* is the result of the fusion between the tracking and classification module. Given a video sequence, and assuming that tracking is perfect, it is the estimated ratio of correct predictions over all predictions made by the ACS accumulated response over a given number of video frames. *Compression* refers to the average number of training patterns per category created in the $F_2$ layer.

While particle diversity is computed as described in Section IV, the $Q$ statistic pairwise indicators is used to compute correlation between two ensemble classifiers [8]. Ensemble correlation is defined as the average $Q$ statistic for all possible pairwise classifiers combination.

## VI. Results and Discussion

For both IIT-NRC and MoBo databases, overall performance and diversity of the heterogeneous ensembles defined with both DPSO algorithms are presented in Tables II and III. As all methods using ensembles eventually achieve a classification rate of 100%, the number of frames necessary to reach that level of accuracy is instead showed alongside compression. An example of incremental learning's impact on recognition capabilities of the proposed heterogeneous ensemble selected with the DNPSO algorithm is shown in Figure 4. Note that, for both learning scenarios, the addition of only one network at a time with the accuracy-based greedy

TABLE II

NUMBER OF ROIS NECESSARY TO ACHIEVE A VIDEO-BASED CLASSIFICATION RATE OF $100 \pm 0\%$ AFTER LEARNING BOTH DATABASES USING THE TWO SCENARIOS. COMPRESSION VALUES ARE PRESENTED WITH THEIR 90% CONFIDENCE INTERVAL.

| Method | | DNPSO and $EoLB_t^{+d}$ | DNPSO and $EoLB_t^{all}$ | DNPSO and $FAM_{g,t}$ | SPSO and $EoLB_t^{+d}$ | SPSO and $EoLB_t^{all}$ | SPSO and $FAM_{g,t}$ | PSO and $EoFAM_t^{all\text{-}B}$ | *kNN* |
|---|---|---|---|---|---|---|---|---|---|
| **IIT-NRC data base** | | | | | | | | | |
| **N. of ROIs** | Enroll. | 27 | 27 | never | 27 | 26 | never | 20 | 23 |
| | Update | 26 | 22 | never | 24 | 23 | never | 20 | 23 |
| **Compres.** | Enroll. | $0.46 \pm 0.04$ | $0.14 \pm 0.01$ | $8.1 \pm 0.9$ | $0.18 \pm 0.09$ | $0.13 \pm 0.01$ | $6.2 \pm 0.5$ | $0.06 \pm 0.01$ | 1 |
| | Update | $0.41 \pm 0.04$ | $0.11 \pm 0.01$ | $5.0 \pm 0.4$ | $0.15 \pm 0.01$ | $0.11 \pm 0.01$ | $4.9 \pm 0.4$ | $0.06 \pm 0.01$ | 1 |
| **MoBo data base** | | | | | | | | | |
| **N. of ROIs** | Enroll. | 41 | 47 | never | 49 | 49 | never | 6 | 5 |
| | Update | 3 | 1 | never | 5 | 3 | never | 6 | 5 |
| **Compres.** | Enroll. | $0.70 \pm 0.09$ | $0.17 \pm 0.01$ | $10.8 \pm 0.8$ | $0.26 \pm 0.09$ | $0.17 \pm 0.01$ | $8.1 \pm 0.9$ | $0.11 \pm 0.01$ | 1 |
| | Update | $0.55 \pm 0.05$ | $0.16 \pm 0.01$ | $9.6 \pm 0.6$ | $0.26 \pm 0.01$ | $0.16 \pm 0.01$ | $7.8 \pm 0.7$ | $0.11 \pm 0.01$ | 1 |

TABLE III

CLASSIFIER CORRELATION ($Q$ STATISTIC) AND PARTICLE DIVERSITY ($\overline{\delta_{e_1e_2}}$) FOR ENSEMBLES CREATED WITH THE GREEDY SEARCH PROCESS, AND THE ENTIRE POOL. RESULTS ARE PRESENTED, WITH THEIR 90% CONFIDENCE INTERVAL, AFTER LEARNING THE DATABASES USING THE TWO SCENARIOS.

| Method | | | DNPSO and $EoLB_t^{+d}$ | DNPSO and $EoLB_t^{all}$ | SPSO and $EoLB_t^{+d}$ | SPSO and $EoLB_t^{all}$ | PSO and $EoFAM_t^{all\text{-}B}$ |
|---|---|---|---|---|---|---|---|
| **IIT-NRC data base** | **Q statistic** | Enrollment | $0.79 \pm 0.02$ | $0.70 \pm 0.03$ | $0.77 \pm 0.02$ | $0.37 \pm 0.06$ | $0.40 \pm 0.04$ |
| | | Update | $0.83 \pm 0.01$ | $0.82 \pm 0.01$ | $0.77 \pm 0.02$ | $0.40 \pm 0.04$ | $0.40 \pm 0.04$ |
| | $\overline{\delta_{e_1e_2}}$ | Enrollment | $0.79 \pm 0.06$ | $0.77 \pm 0.06$ | $0.62 \pm 0.06$ | $0.75 \pm 0.06$ | $0.75 \pm 0.06$ |
| | | Update | $0.67 \pm 0.06$ | $0.78 \pm 0.06$ | $0.65 \pm 0.06$ | $0.73 \pm 0.06$ | $0.75 \pm 0.06$ |
| **MoBo data base** | **Q statistic** | Enrollment | $0.71 \pm 0.03$ | $0.67 \pm 0.04$ | $0.68 \pm 0.04$ | $0.38 \pm 0.06$ | $0.89 \pm 0.04$ |
| | | Update | $0.76 \pm 0.03$ | $0.67 \pm 0.04$ | $0.77 \pm 0.03$ | $0.47 \pm 0.06$ | $0.89 \pm 0.04$ |
| | $\overline{\delta_{e_1e_2}}$ | Enrollment | $0.75 \pm 0.06$ | $0.81 \pm 0.06$ | $0.69 \pm 0.06$ | $0.77 \pm 0.06$ | $0.55 \pm 0.05$ |
| | | Update | $0.77 \pm 0.06$ | $0.73 \pm 0.06$ | $0.70 \pm 0.05$ | $0.80 \pm 0.06$ | $0.55 \pm 0.05$ |

search ($EoFAM_t^{+a}$) is not sufficient to change the recognition capabilities of the ensemble. The performance of this method being exactly the same as that of the global best network only ($FAM_{g,t}$), it is thus not shown.

Results indicate that even if ensembles selected using the proposed greedy search ($EoLB_t^{+d}$) necessitate a few more frames to yield a perfect classification rate than ensembles made of all the classifiers ($EoFAM_t^{all}$), it is able to do so with as much as a fourth of the resources (see Table II). Since one of the criteria used in Algorithm 1 is performance, ensembles selected through diversity-based greedy search account for most of the correct classifications, while the rest of the classifiers, when all the swarm is used, represent an excess of resources. They consist in suboptimal classifiers that are different, but still overshadowed by those who perform better (i.e. classifiers in $EoLB_t^{+d}$). For the entire swarm ($EoFAM_t^{all}$), this is reflected by a lower correlation rate ($Q$ statistic), and a higher or comparable particle diversity ($\overline{\delta_{e_1e_2}}$).

When comparing the two DPSO algorithms, subswarms formation through dynamical niching gives an advantage compared to speciation. While the neighborhood topology used with DNPSO almost always depends on the neighborhood size for the subswarms creation, speciation relies entirely on the minimal distance between local best particles. Unlike DNPSO, which contains additional mechanisms that redefines each particle's local best, SPSO will almost always create a few subswarms located far from any local optima. For $EoLB_t^{+d}$ and $EoLB_t^{all}$, both DPSO algorithms achieve comparable classification rates, but DNPSO does so with less resources. Moreover, particle diversity of $EoLB_t^{+d}$ obtained with DNPSO are either, higher, or comparable to that of SPSO.

An example of incremental learning's impact on classification rate, during the update scenario of a video-based face recognition application, is illustrated in Figure 4. After learning the first data block ($D_1$), all classes are defined with few data, and the face recognition system is not able to reach a classification rate of 100% and, as classes are updated, the recognition capability increases.

Still, unless all the learning data are presented, Figure 4 shows that classification rate decreases when samples obtained at the end of the sequences are used. The first block consist of images taken at the beginning of the video sequences, in which the filmed individuals, positioned facing the camera, are barely moving. However, test samples obtained in the second part of the sequences, where the individuals are moving, are wrongly predicted as they represent perspectives of each face that were not seen in $D_1$. When data is added, this drop in classification rate decreases, and once all classes are updated, the ACS achieves a perfect classification rate for all sequences. Unlike with incremental learning, batch learning is able to account

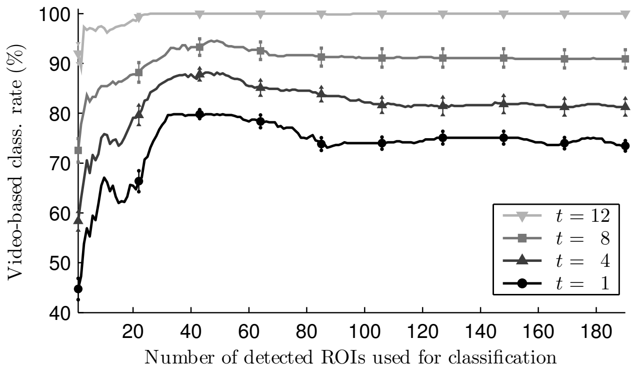



Fig. 4. Example of the average ensemble video-based classification rate's evolution versus the number of ROIs used to identify classes of the IIT-NRC data base. The classification rate shown was evaluated during the update incremental learning scenario with the proposed ensemble selection method ($EoLB_t^{+d}$) and DNPSO. Error bars correspond to the 90% confidence interval.

for the changing classification environment. However, while ensembles obtained with the original PSO algorithms achieve perfect recognition with less ROIs and yield higher classifier diversity, it necessitates nearly twice as much resources than using all the swarm during incremental learning.

## VII. Conclusion

In this paper, two DPSO algorithms are compared for the evolution of classifier ensembles during supervised incremental learning in video-based face recognition. With a DPSO-based learning strategy applied to an adaptive classification system, and an greedy search process, the DPSO algorithms abilities to create ensembles that improves recognition capabilities are assessed for classification rate, resources requirements, and diversity. Overall results indicate that using a DPSO algorithm is an efficient way to create diversified heterogeneous ensembles as their recognition capabilities are comparable to those of using the entire pool as an ensemble, but with a fraction of the resources.

However, while classifiers are defined according to the particles personal best position, both DPSO algorithms create subswarms with the current positions instead. In consequences, the SPSO mechanism used to maintain diversity within each subswarm does not significantly affect classifier diversity. Moreover, using dynamical niching (DNPSO) rather than speciation (SPSO) to create subswarms results in comparable classification rates, although niching requires fewer resources. Future work should then involve devising a DPSO algorithm specifically aimed at the creation of diversified ensembles when using the ACS. While maintaining the mechanisms and properties of DPSO algorithms, knowledge about the personal best positions, rather than the current positions, should be used to create subswarms and maintain diversity.


## Acknowledgment

This research was supported by the Natural Sciences and Engineering Research Council of Canada.


## References


[1] A. Rattani, B. Freni, G. L. Marcialis, and F. Roli, "Template update methods in adaptive biometric systems: A critical review," in *ICB*, 2009, pp. 847–856.

[2] J.-F. Connolly, E. Granger, and R. Sabourin, "An adaptive classification system for video-based face recognition," *Information Sciences (in press)*, 2010.

[3] ——, "An adaptive ensemble of fuzzy artmap neural networks for video-based face classification," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*, Jul. 2010, pp. 1–8.

[4] A. M. Canuto, J. a. C. X. J. Marjory C.C. Abreu, Lucas de Melo Oliveira, and A. de M. Santos, "Investingating the influence of the choice of the ensemble members in accuracy and diversity of selection-based and fusion-based methods for ensembles," *Pattern Recognition Letters*, vol. 26, pp. 472–486, 2007.

[5] S. T. Hadjitodorov, L. I. Kuncheva, and L. P. Todorova, "Moderate diversity for better cluster ensembles," *Information Fusion*, vol. 7, no. 3, pp. 264–275, 2006.

[6] D. F. de Olivieira, A. M. P. Canuto, and M. C. P. de Souto, "Use of multi-objective genetic algorithms to investigate the diversity/accuracy dilemma in heterogeneous ensembles," in *IEEE Int'l Joint Conf. on Neural Networks*, Atlanta, USA, Jun. 2009, pp. 1238–1245.

[7] K. Sirlantzis, S. Hoque, and M. Fairhurst, "Diversity in multiple classifier ensembles based on binary feature quantisation with application to face recognition," *Journal of Visual Language and Computing*, vol. 8, pp. 437–445, 2008.

[8] A. Ulaş, M. Semerci, O. T. Yildiz, and E. Alpaydin, "Incremental construction of classifier and discriminant ensembles," *Information Sciences*, no. 179, pp. 1298–1318, 2009.

[9] G. Valentini, "Ensemble methods based on bias-variance analysis," Ph.D. dissertation, PhD thesis, University of Genova, Genova, Switzerland, 2003.

[10] Y. Liu, X. Yao, Q. Zhao, and T. Higuchi, "Evolving a cooperative population of neural networks by minimizing mutual information," in *Proceeding of the IEEE Congress on Evolutionary Computation (CEC)*, Seoul, Korea, May 2001, pp. 384–389.

[11] Z.-H. Zhou and W. Tang, "Ensembling neural networks: Many could be better than all," *Artificial Intelligence*, vol. 137, no. 1-2, pp. 293–253, 2002.

[12] G. Brown, J. Wyatt, R. Harris, and X. Yao, "Diversity creation methods: a survey and categorisation," *Information Fusion*, vol. 29, no. 6, pp. 5–20, 2005.

[13] E. Granger, P. Henniges, L. S. Oliveira, and R. Sabourin, "Supervised learning of fuzzy artmap neural networks through particle swarm optimization," *Journal of Pattern Recognition Research*, vol. 2, no. 1, pp. 27–60, 2007.

[14] W. Zhao, R. Chellappa, P. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Computing Surveys*, vol. 35, pp. 399–458, 2003.

[15] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen, "Fuzzy artmap: A neural network architecture for incremental supervised learning of analog multidimensional maps," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 698–713, 1992.

[16] T. Blackwell, J. Branke, and X. Li, "Particle swarms for dynamic optimization problems," in *Swarm Intelligence*, ser. Natural Computing Series. Springer Berlin Heidelberg, 2008, pp. 193–217.

[17] A. Nickabadi, M. M. Ebadzadeh, and R. Safabakhsh, "Evaluating the performance of DNPSO in dynamic environments," in *IEEE Int'l Conference on Systems, Man, and Cybernetics*, Singapore, Oct. 2008, pp. 12–15.

[18] O. Olorunda and A. P. Engelbrecht, "Measuring exploration/exploitation in particle swarms using swarm diversity," in *IEEE Congress on Evolutionary Computation (CEC)*, Hong Kong, China, May 2008, pp. 1128–1134.

[19] D. O. Gorodnichy, "Video-based framework for face recognition in video," in *Second Workshop on Face Processing in Video in Proc. on Conf. on Computer and Robot Vision*, Victoria, Canada, May 2005, pp. 325–344.

[20] R. Gross and J. Shi, "The cmu motion of body (mobo) database, 2001," Carnegie Mellon University, Tech. Rep. CMU-RI-TR-01-18, 2001.

[21] J. Kennedy, "Some issues and practices for particle swarms," in *IEEE Swarm Intelligence*, Honolulu, USA, Apr. 2007, pp. 162–169.