Pattern Recognition 45 (2012) 2460–2477

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

# Evolution of heterogeneous ensembles through dynamic particle swarm optimization for video-based face recognition

Jean-François Connolly, Eric Granger*, 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*

ARTICLE INFO

*Article history:*
Received 24 January 2011
Received in revised form
2 December 2011
Accepted 15 December 2011
Available online 27 December 2011

*Keywords:*
Classification
Heterogeneous ensembles
Diversity
Incremental learning
Adaptive multiclassifier systems
ARTMAP neural networks
Biometrics
Face recognition in video

ABSTRACT

In many real-world applications, pattern recognition systems are designed a priori using limited and imbalanced data acquired from complex changing environments. Since new reference data often becomes available during operations, performance could be maintained or improved by adapting these systems through supervised incremental learning. To avoid knowledge corruption and sustain a high level of accuracy over time, an adaptive multiclassifier system (AMCS) may integrate information from diverse classifiers that are guided by a population-based evolutionary optimization algorithm. In this paper, an incremental learning strategy based on dynamic particle swarm optimization (DPSO) is proposed to evolve heterogeneous ensembles of classifiers (where each classifier corresponds to a particle) in response to new reference samples. This new strategy is applied to video-based face recognition, using an AMCS that consists of a pool of fuzzy ARTMAP (FAM) neural networks for classification of facial regions, and a niching version of DPSO that optimizes all FAM parameters such that the classification rate is maximized. Given that diversity within a dynamic particle swarm is correlated with diversity within a corresponding pool of base classifiers, DPSO properties are exploited to generate and evolve diversified pools of FAM classifiers, and to efficiently select ensembles among the pools based on accuracy and particle swarm diversity. Performance of the proposed strategy is assessed in terms of classification rate and resource requirements under different incremental learning scenarios, where new reference data is extracted from real-world video streams. Simulation results indicate the DPSO strategy provides an efficient way to evolve ensembles of FAM networks in an AMCS. Maintaining particle diversity in the optimization space yields a level of accuracy that is comparable to AMCS using reference ensemble-based and batch learning techniques, but requires significantly lower computational complexity than assessing diversity among classifiers in the feature or decision spaces.

## 1. Introduction

In pattern recognition systems, neural or statistical classifiers define class models (or hypotheses), using data samples defined in a $\mathbb{R}^I$ input feature space (also referred to as hypothesis space [5]), and map those models to a decision space to perform predictions. Exploiting several views of a same problem with classifier ensembles has been shown to improve the overall accuracy and reliability for a wide range of applications. However, generating an accurate pool of base classifiers and selecting an ensemble among that pool that maximizes prediction precision are challenging tasks. One key element in the success of classifiers ensembles that has attracted a great deal of interest in recent years is *classifier diversity measures* [6,24,27,37,45,50]. Since diversity is difficult to assess in the input feature space, these measures compute the disagreement between classifiers in the decision space, over several predictions. Through bias-variance error decomposition, it has been shown empirically that considering diversity for ensemble selection improves the generalization capabilities of multiple classifiers systems [5].

Diversity can be achieved via (1) different starting points in the input feature space, using a learning algorithm trained with different initial conditions, (2) different sets of accessible hypotheses using different training data sets (*e.g.*, boosting and bagging) or different learning algorithms, and (3) representation space traversal that optimizes parameters, using a penalty term or evolutionary method, to ensure that base classifiers occupy different areas in the feature space. In the latter case, the hyperparameters of a classifier (*e.g.*, learning rate) define an optimization space [20]. As described in Fig. 1, supervised learning strategies may then allow to optimize a classifier's hyperparameters such as accuracy is maximized. Since these hyperparameters govern the learning dynamics of a classifier, diversity among solutions in the *optimization environment* leads to ensemble diversity in the *classification environment*. Diversity can thus be maintained without relying on costly classifier diversity measures.

* Corresponding author. Tel.: +1 514 396 8650.
*E-mail addresses:* jfconnolly@livia.etsmtl.ca (J.-F. Connolly), Eric.Granger@etsmtl.ca (E. Granger), Robert.Sabourin@etsmtl.ca (R. Sabourin).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.12.016

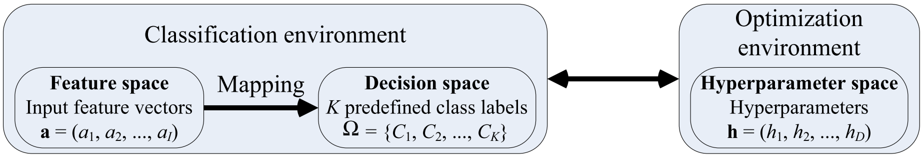
Classification environment
Feature space
Input feature vectors
$\mathbf{a} = (a_1, a_2, ..., a_I)$
Mapping
Decision space
$K$ predefined class labels
$\Omega = \{C_1, C_2, ..., C_K\}$
Optimization environment
Hyperparameter space
Hyperparameters
$\mathbf{h} = (h_1, h_2, ..., h_D)$

**Fig. 1.** Pattern classification systems may be defined according to two environments. A *classification environment* that maps a $\mathbb{R}^I$ input feature space to a decision space, respectively defined by feature vectors **a**, and a set of class labels $\Omega$. Interacting with the latter is an *optimization environment*, where each vector **h** indicates a position in the hyperparameter space defined according a classifier's learning algorithm. The representation space traversal seeks to maintaining diversity among classifiers by exploiting the interaction between these two environments. The basic assumption is that different positions in the hyperparameter space lead to different class models in the feature space, and thus different class label $C_k$ predictions in the decision space.

The recognition of individuals based on their biometric traits provides a powerful alternative to traditional authentication schemes that are presently applied in security and surveillance systems. In biometric applications, such as face recognition in video, the collection and analysis of labeled reference samples to design biometric systems (during an enrollment of re-enrollment process) is often expensive and time consuming. Samples acquired from video streams in *unconstrained* scenes are generally of poor quality with low resolution, resulting in classifiers applied to biometric matching that are often trained with limited and unbalanced data with much inter- and intra-class variability, Moreover, given that facial regions are often captured discreetly, without cooperation, they are subject to considerable variations due to limited control over operational conditions (*e.g.*, illumination, pose, facial expression, orientation and occlusion). In addition, operating conditions and individual physiology may even change over time, either temporary (*e.g.*, haircut, glasses, lighting, etc.) or permanently (*e.g.*, scars, aging). New informations, such as input features and output classes, may suddenly emerge and previously acquired data may eventually become obsolete in dynamically changing classification environments [21,49]. These factors contribute to a growing divergence between the biometric model of an individual and its underlying class distribution.[1]

It is common in many biometric applications to acquire additional data and knowledge from the environment or other sources over time, after the system has originally been deployed for operations. For accurate recognition of individuals, biometric systems should adapt their models over time in response to new or changing input features, data samples, priors, classes and environments. In this paper, it is assumed that new reference data becomes available to create new biometric models when individuals enroll to the system, and to update models of individuals previously enrolled to the system. Some adaptive biometric systems have been proposed in the literature to refine biometric models according to the intra-class variations in input samples [44]. Indeed, with self-adaptive or semi-supervised learning strategies, biometric models are initially designed during enrollment using labeled training data, and then updated with highly confident unlabeled data obtained during operations [40,43]. These strategies are however vulnerable to outliers, dispersion and overlap in class distributions. Stringent criteria are required for selection of highly confident data, to minimize the probability of introducing impostor data into updated biometric models. In this paper, supervised learning strategies are considered, and new data samples are assumed to be analyzed and labeled by an operator with expert knowledge of intra-class variations. Labeled data becomes available, for instance, over multiple (re)enrollment sessions, or when operational scenarios are analyzed off-line, and can allow an operator to gradually build the biometric models of a system over time. Adaptive biometric systems in literature have used newly acquired reference samples to update the selection of a user's template from a gallery via clustering and editing techniques [51]. Others have performed on-line learning of genuine samples over time to update each user's single super template [26]. It is however difficult to represent intra-class variations with a single template [44].

In previous work, the authors proposed an adaptive classification system (ACS) to update biometric models of individuals in response to new labeled reference data from the operational environment during video-based face recognition [13]. It uses the fuzzy ARTMAP (FAM) neural network for supervised incremental learning of limited data, as well as fast and efficient matching of facial regions detected in video streams against the model of individuals enrolled to a face recognition system. The authors have showed that (1) adaptation of a FAM network during supervised incremental learning is a dynamic optimization problem in the hyperparameter space, and (2) corruption of the biometric models resulting from incremental learning of new data can be reduced using an ensemble-based approach [14]. It exist several directions to address uncertain classification environments with ensembles, such as changing classifier combination rules, updating classifiers using the new reference data, and changing ensemble structure by replacing old or underperforming members [29]. However, few of these approaches explicitly exploit classifier diversity when adapting ensembles over time in a context where to few data are available.

Given the limited amount of data available to design biometric systems, creating diversity among classifiers through *representation space traversal* is an efficient way to exploit those data to provide reliable classifier ensembles. For instance, with a cooperative neural network co-evolution paradigm [41], evolutionary algorithms have been used to create *heterogeneous* ensembles [52].[2] It allows, exploring a hyperparameter space to train classifiers of the same type, on the same data, but with different learning dynamics. These approaches, where classifiers cooperate, exchange information, but yet have their design and training be independent, have been shown to provide more accurate ensembles [2,18,32,58].

In this paper, the relationship between diversity in the classification and optimization environments is exploited for efficient design of heterogeneous ensembles of classifiers in video-based face recognition. Under the hypothesis that diversity in the hyperparameter space is correlated with diversity among a corresponding pool of classifiers in the feature and decision spaces, a specialized learning strategy based on dynamic particle swarm optimization (DPSO) is proposed for supervised incremental learning of new data. This incremental DPSO-based learning strategy is applied to an adaptive multiclassifier system (AMCS) that consists of a pool FAM networks [8] for classification that interacts with a niching version of DPSO [36]. The DPSO-based

[1] Typically designed during an a priori enrollment phase, the biometric model of an individual for matching consists of one or more templates (reference samples) or the parameters statistical or structural model of reference samples.

[2] This definition differs with respect to certain other definitions of heterogeneous ensembles found in literature [37,42].

learning strategy incrementally evolves a heterogeneous ensemble of FAM networks in response to new reference data. Each particle in the optimization environment corresponds to a FAM network in the classification environment, and the DPSO strategy cojointly optimizes *all classifier parameters* – hyperparameters, weights, and architecture – of a pool of classifiers such as classification rate is maximized. The ability of DPSO algorithms to find and track several changing local optima in the hyperparameter space is exploited by the AMCS to create a diversified pool (or swarm) of *heterogeneous* classifiers. DPSO properties are applied in a novel greedy search process to efficiently select an ensemble among the pool of FAM classifiers, based on accuracy and particle swarm diversity.

This study focuses on video-based face recognition applications in which two incremental learning scenarios may occur—enrollment and update of facial models. Performance of this system is assessed in terms of classification rate and resource requirements for incremental learning of new data blocks from two real-world video data sets—Institute of Information Technology of the Canadian National Research Council (IIT-NRC) [19] and Motion of Body (MoBo) [22]. In proof-of-concept experiments, the AMCS performs biometric matching of facial regions against the facial model of individuals enrolled to a system for closed-set identification. Finally, the relationship between diversity in a classifier's hyperparameter space and diversity in its feature and decision space is analyzed for both batch and incremental learning cases.

In Section 2, the AMCS is described along with the FAM network used for classification, and the DPSO algorithm used to optimize system parameters. In Section 3, the new DPSO-based incremental learning strategy used to evolve heterogeneous ensembles is described. Application, data bases, incremental learning scenarios, protocol, and performance measures used for proof-of-concept simulations are described in Section 4. Finally, experimental results are presented and discussed in Section 5.

## 2. An adaptive multiclassifier system

Fig. 2 depicts the evolution of an adaptive multiclassifier system (AMCS) for supervised incremental learning of new reference labeled samples. It is composed of a pool of base classifiers, each one suitable for supervised incremental learning, a dynamic evolutionary optimization module that tunes the user-defined hyperparameters of each classifier, and a long term memory (LTM) that stores and manages incoming data for validation. This system differs from the system originally proposed in that a new DPSO incremental learning strategy allow to efficiently form a heterogeneous ensemble of classifiers [13]. It evolves a pool of classifiers, and is now composed of a selection and fusion module for efficient combination of heterogeneous ensembles.

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 the pool, or swarm, of incremental classifiers. Each classifier is associated to a particle in the hyperparameter space, and a dynamic optimization module using a DPSO-based learning strategy cojointly determines the classifiers hyperparameters, architecture, and parameters such that classification rate is maximized [14]. Once the optimization process is complete, the selection and fusion module produces a heterogeneous ensemble by selecting and combining classifiers from the swarm, based on their accuracy and diversity. The LTM stores data samples from each individual class for validation during incremental learning and fitness estimation of particles on the objective function [13]. The data from $D_t$ is partitioned and combined with that of the LTM to create three subsets: a training data set $D_t^{\mathrm{t}}$, a validation data set $D_t^{\mathrm{v}}$, and a fitness estimation data set $D_t^{\mathrm{f}}$.

In this paper, a particular realization of this AMCS is considered. The fuzzy ARTMAP (FAM) neural network [8] is employed for incremental learning classification and a dynamical niching particle swarm optimization (DNPSO) algorithm [36] is used for dynamic optimization. The rest of this section provides additional details on the FAM classifier and the DNPSO algorithm used within the AMCS.

### 2.1. Fuzzy ARTMAP neural network classifiers

Fuzzy ARTMAP [9] is a versatile neural classifier that may provide a high level of prediction accuracy with moderate time and memory complexity [20]. As such, fuzzy ARTMAP has been successfully applied to a wide variety of pattern recognition problems. It is very promising for fast and efficient for biometric matching (of feature patterns against the model of individuals enrolled to a face recognition system) due to its ability to perform fast, stable, on-line, unsupervised or supervised, and incremental learning from limited amount of training data. A key feature of the ARTMAP networks is their unique solution to the stability-plasticity dilemma. The popular fuzzy ARTMAP integrates the fuzzy ART to process both analog and binary-valued input patterns to the original ARTMAP architecture [8]. Several other ARTMAP networks have been proposed to address this architecture to specific problems. Members of the ARTMAP family can be broadly divided according to their internal matching

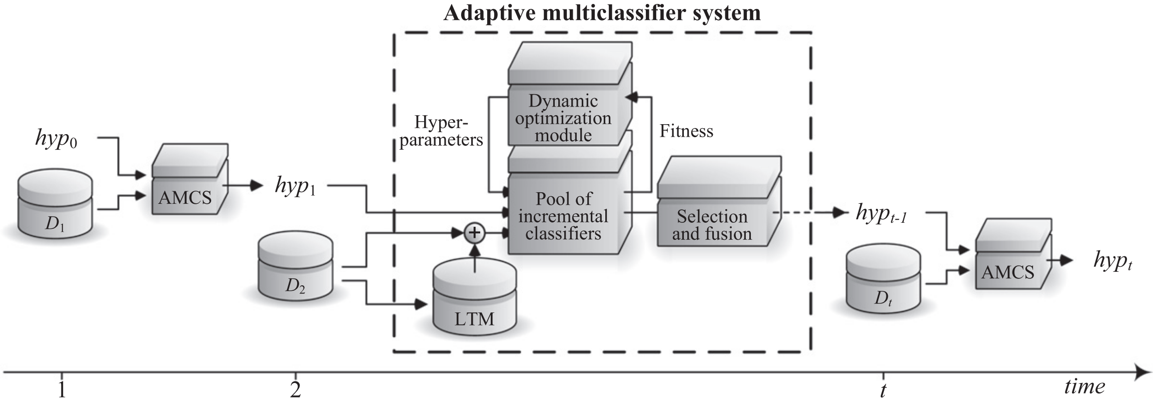

**Fig. 2.** Evolution over time of the adaptive multiclassifier system (AMCS) in a generic incremental learning scenario, where new blocks of data are used to update a swarm of classifiers. Let $D_1, D_2, \ldots$ be blocks of learning data that become available at different labeled instants in time $t = 1, 2, \ldots, T$. The AMCS starts with an initial hypothesis $hyp_0$ according to prior knowledge of the domain. Each hypothesis $hyp_{t-1}$ are updated to $hyp_t$ by the AMCS on the basis of a new data blocks $D_t$.

process, which depends on either deterministic or probabilistic category activation [12].

As shown in Fig. 3, the fuzzy ARTMAP (FAM) architecture consists of three layers: (1) an input layer $F_1$ of $2I$ neurons, with two neurons associated with each input feature (in $\mathbb{R}^I$), (2) a competitive layer $F_2$ of $J$ neurons, each one associated to a recognition category in the feature space, and (3) a map field $F^{ab}$ of $K$ output neurons, each one corresponding to a class [8].

In supervised training mode, FAM learns an arbitrary mapping between training set patterns $\mathbf{a}=(a_1,a_2,\ldots,a_I)$ and their corresponding binary supervision patterns $\mathbf{c}=(c_1,c_2,\ldots,c_K)$. These patterns are coded to have the value $c_k$=1 if $k^*$ is the target class label for $\mathbf{a}$, and zero elsewhere. Components of the vector $\mathbf{a}$ are scaled so that each $a_i \in [0,1]$, for $i=1\ldots I$. Complement coding doubles the number of components in the input vector, which becomes $\mathbf{A} \equiv (a_1,a_2,\ldots,a_I,1-a_1,1-a_2,\ldots,1-a_I)$. The prototype vector $\mathbf{w}_j=(w_{1j},\ldots,w_{2Ij})$, linking each $F_1$ input node to $F_2$ node $j$, may be visualized as a hyper-rectangle in the $\mathbb{R}^I$ feature space defined by all the input vectors $\mathbf{a}$ that selected node $j$ during training. Binary weight vectors $\mathbf{w}_j^{ab}=(w_{j1},\ldots,w_{jK})$ connects $F_2$ nodes to one of the $K$ classes of $F^{ab}$.

Initially, all the $F_2$ nodes are uncommitted, all weight values $w_{ij}$ are initialized to 1, and all weight values $w_{jk}^{ab}$ are set to 0. Prior processing each new training pattern $\mathbf{a}$, the vigilance parameter is set: $\rho=\overline{\rho}$. Given an input $\mathbf{a}$ and supervision output $\mathbf{c}$, the coding field $F_2$ is activated according to the Weber law choice function

$$T_j(\mathbf{A}) = |\mathbf{A} \wedge \mathbf{w}_j|/(\alpha+|\mathbf{w}_j|), \tag{1}$$

where $(\mathbf{p}\wedge\mathbf{q})_i \equiv \min(p_i,q_i)$, $|\mathbf{p}| \equiv \sum_{i=1}^{2I}|p_i|$, and $\alpha$ is the *choice parameter*. With winner-take-all coding, the $F_2$ node $j^*$ that receives the largest activation $T_{j^*}(\mathbf{A})$ is chosen, and undergoes the vigilance test defined by

$$|\mathbf{A}\wedge\mathbf{w}_{j^*}|/|\mathbf{A}| = |\mathbf{A}\wedge\mathbf{w}_{j^*}|/I > \rho, \tag{2}$$

where $\rho \in [0,1]$ is the dimensionless *vigilance parameter*. If node $j^*$ passes the vigilance test, FAM predicts the class corresponding to $j^*$. If the prediction is correct (*i.e.*, $k(j^*)=k^*$), weight vector $\mathbf{w}_j$ undergoes learning and is adjusted according to

$$\mathbf{w}'_{j^*} = \beta(\mathbf{A}\wedge\mathbf{w}_{j^*})+(1-\beta)\mathbf{w}_{j^*}, \tag{3}$$

where $\beta \in [0,1]$ is a fixed *learning rate parameter*.

If neuron $j^*$ does not pass the vigilance test, or makes an *incorrect* class prediction, it is deactivated for the rest of the search process with the current input $\mathbf{a}$. When an incorrect class prediction occurs, a *match tracking* signal also adjusts vigilance such as $\rho=|\mathbf{A}\wedge\mathbf{w}_{j^*}|/I+\epsilon$, where $\epsilon$ is the match tracking parameter.[3] The network then searches for another $F_2$ node that either satisfies both requirements, or commits a new $F_2$ node to encode $\mathbf{a}$ if no such node exist. When a new $F_2$ node is committed, network size is actualized ($J=J+1$), and the last committed node learns the correct output class by setting $\mathbf{w}_J=\mathbf{A}$ and $w_{Jk^*}^{ab}=1$.

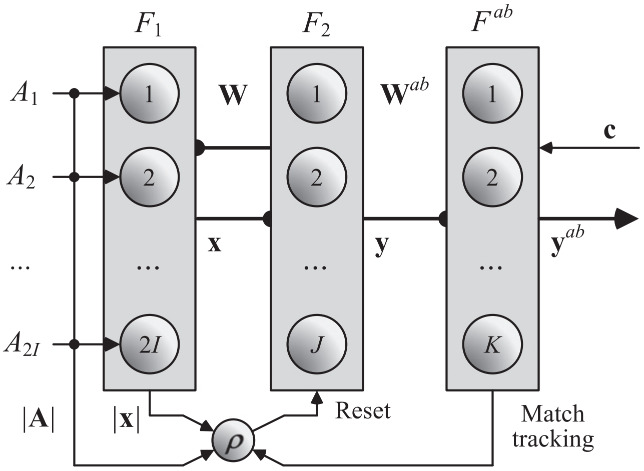

**Fig. 3.** Fuzzy ARTMAP neural network.

During training, FAM internal dynamics are governed by four hyperparameters: 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 $\overline{\rho} \in [0,1]$. Let $\mathbf{h}=(\alpha,\beta,\epsilon,\overline{\rho})$ be defined as the vector of FAM hyperparameters, these are inter-related and each have a distinct impact on network dynamics. While $\alpha$ and $\epsilon$ determine the depth of search attained before an uncommitted node is selected, and $\overline{\rho}$ limits the maximal size of the category hyper-rectangles in the $\mathbb{R}^I$ feature space. Although this is affected by the match tracking signal $\epsilon$, low baseline vigilance generally results in large hyper-rectangles and leads to broad generalization and abstract memories, while high vigilance yields small hyper-rectangles, leading to narrow generalization and detailed memories. During learning, $\beta$ determines the speed with which the recognition categories expand to fit $\mathbf{a}$. The algorithm can be set to slow learning with $0<\beta<1$, or to fast learning with $\beta=1$. With fast learning, each hyper-rectangles is just large enough to enclose the training set patterns $\mathbf{a}$ to which it has been assigned. Prototype vector $\mathbf{w}_j$ records the largest and smallest component values of training subset patterns $\mathbf{a}$ assigned to category $j$.

A standard vector of hyperparameters $\mathbf{h}_{\text{std}}=(\alpha=0.001,\beta=1,\epsilon=0.001,\overline{\rho}=0)$ is commonly fixed to minimize network complexity [8]. However, Fig. 4 illustrates with a synthetic 2D data base [52] that adjusting these hyperparameters allows to adapt FAM learning dynamics with regards to currently available training data. By using different hyperparameters settings to train FAM, each network learns different hyper-rectangles to fit the same data, leading to different decision boundaries and predictions. This diversity of opinion among classifiers may then be measured using several different indicators [6,24,27,37,45,50].

It is very well known that ensembles of classifiers can be used to improve the generalization capabilities of pattern recognition systems applied in different domains, including face recognition in video [17,33,46]. But as Fig. 4 shows, varying the hyperparameter values of several FAM neural networks provides an easy mean to model the same data with different perspective and generate a diversified pool of heterogeneous classifiers when few learning reference data is available. Given this correlation between diversity of hyperparameter values and decision boundaries, classifier diversity can also be easily exploited during ensemble selection form the pool to further improve accuracy of the face recognition system [5].

### 2.2. Dynamic particle swarm optimization

Particle swarm optimization (PSO) is a population-based stochastic optimization technique that was inspired by social behavior of bird flocking and fish schooling. With PSO, each particle corresponds to a single solution in the hyperparameter space, and the population of particles is called a swarm. Particles move through the hyperparameter 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 (like genetic algorithms), each particle always stores its best position and the best position of its surroundings in its memory.

Originally developed for static optimization problems, the PSO algorithm has been adapted for dynamic optimization problems

[3] In this paper, negative match tracking is employed [10].

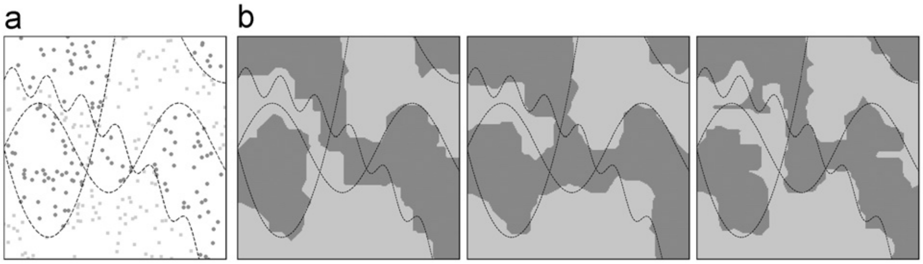

**Fig. 4.** Training data (a) from the P2 synthetic data base [52], and decision boundaries for FAM trained with different hyperparameters that are respectively (b): $\mathbf{h}=(70,0.70,0.80,0.85)$, $\mathbf{h}=(13,0.41,0.08,0.86)$, and $\mathbf{h}=(67,0.73,0.68,0.89)$. (a) Original data; (b) FAM decision boundaries.

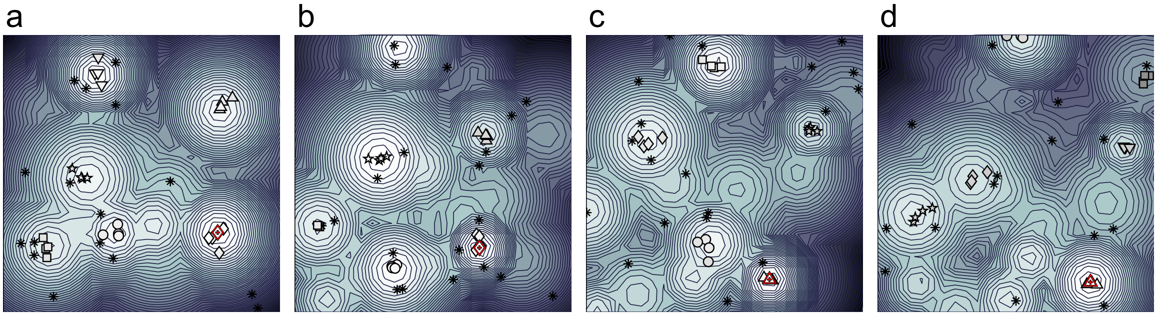

**Fig. 5.** Evolution of DNPSO particles for different changes in a type III optimization environment using the 2D multipeak benchmark problem [4]. In a video-based face recognition application for instance, this could be the classification rate landscape in a 2D hyperparameter space. Subswarms (shapes: circle, rectangle, etc.) are created dynamically around the *masters*—particles that detected local optima. Subswarms converge toward the local optima detected for the objective function. Free particles (stars), that are not associated to any subswarms, are free to explore the optimization space using only their cognitive influence. At different times $t$, the personal best of each particles is reevaluated to accommodate changes that may occur on the objective function. (a) $t=1$, (b) $t=2$, (c) $t=3$ and (d) $t=4$.

by adding mechanisms to (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 each particle, and (3) adapt the memory of its population if change occur in the optimization environment. The latest particle swarm optimization algorithms developed to insure diversity in the swarm are presented in [15,30,36,39]. Change detection and memory adjustment mechanisms for DPSO are presented in [3,7,25,54].

During supervised incremental learning of new data blocks $D_t$, the dynamic optimization module (see Fig. 2) iteratively updates the hyperparameter vector $\mathbf{h}=(\alpha,\beta,\epsilon,\overline{\rho})$ of each FAM classifier in the hyperparameter space, and determines the position $\mathbf{h}$ such that the FAM classification rat is maximized. In this paper, the hyperparameter space is bounded by $\alpha \in [0,100]$, $\beta \in [0,1]$, $\epsilon \in [-1,1]$, and $\overline{\rho} \in [0,1]$. Using PSO to evolve a swarm of FAM networks when data is learned incrementally over time, such adaptation has been shown to correspond to a dynamic optimization problem defined by

$$\text{maximize}\{f(\mathbf{h},t) \mid \mathbf{h} \in \mathbb{R}^4,\ t \in \mathbb{N}_1\}, \tag{4}$$

where the fitness, $f(\mathbf{h},t)$, is the FAM classification rate for a given vector of hyperparameters $\mathbf{h}$, and after learning data set $D_t$ at a discrete time $t$ [13]. There are three different types of dynamic optimization environment [16]: type I, where the location of the optimum changes over time; type II, where the location of the optimum remains fixed, but the value of the objective function optimum's position changes; and type III, where both the location and value of the optimum position change. In [13], it was shown that the optimization problem defined by Eq. (4) constitute a type III optimization environment.

In this paper, the adaptive multiclassifier system (AMCS) employs the Dynamical Niching PSO (DNPSO) algorithm [36] to maximize FAM classification rate as a function of its hyperparameters. As depicted in Fig. 5, this algorithm maintains diversity in the hyperparameter search space by (1) using a local neighborhood topology, where subswarms are dynamically created around *masters* (particles that are their own local best in their neighborhood), by (2) defining a minimal distance within which two masters that cannot co-exist, by (3) allowing free particles that do not belong to a subswarm, to move independently, and by (4) reinitializing those free particles that exhibit low velocities, indicating that they have converged on a non-optimal position. DNPSO has also been adapted for dynamic optimization problems by updating the fitness of the best position of each particle at each iteration. Using the moving peaks benchmark, the DNPSO algorithm has been shown to detect local optima and converge toward the global maximum in a multimodal type III optimization environment [36] (see Fig. 5).

When evolving FAM neural networks, updating the fitness of the best position of each particle at each iteration would double the number of time each network are trained, leading to a very costly process. However, for an AMCS, changes in the objective function may only occur when a new data block $D_t$ becomes available. Thus, the best position's fitness of each particle is only updated when a new $D_t$ is presented to the system, *before* the iterative DNPSO process.

## 3. Strategy for evolving heterogeneous ensemble of FAM networks

The DPSO-based incremental learning strategy proposed in this paper is based on the hypothesis that maintaining diversity among particles in the optimization environment implicitly generates diversity among classifiers in the classification environment. By associating each classifier of a pool to a particle in a swarm, properties of a DPSO algorithm (to maintain diversity in the hyperparameter space) may be exploited to evolve a diversified heterogeneous ensembles of FAM networks over time, as new data becomes available.

This section describes the DPSO-based incremental learning strategy used to evolve heterogeneous ensembles of classifiers in response to new labeled reference samples. First, a diversified pool of FAM networks is generated and evolved according to a DPSO learning algorithm (Section 3.1). This pool (or swarm) allows for efficient selection and fusion of ensembles of classifiers based on FAM accuracy and particle swarm diversity in the hyperparameter space (Section 3.2).

### 3.1. Generation and evolution of heterogeneous classifier pools

Algorithm 1 describes the DPSO algorithm proposed to generate and evolve a diversified pool (or swarm) of $N$ FAM networks. During incremental learning of a data block $D_t$, their hyperparameters, parameters and architecture are cojointly optimized such that the classification rate is maximized. For a PSO algorithm with $n=1,\ldots,N$ particles, each a hyperparameter vector (noted $\mathbf{h}_n$), a total of $2N+1$ FAM networks is required. The system stores $n=1,\ldots,N$ networks $FAM_n^{\mathrm{start}}$ in a short term memory to preserve networks associated with the best position of each particle (noted $\mathbf{h}_n^*$) at time $t-1$. It also stores $FAM_n$, the model associated with $\mathbf{h}_n^*$ during the optimization process at time $t$, and $FAM_{\mathrm{est}}$, a network employed for fitness estimation. To minimize the impact of pattern presentation order at a time $t$, FAM networks are trained using the training data set $D_t^{\mathrm{t}}$ under five different random pattern presentation orders. To determine the number of training epochs, cross-validation is performed with the validation data set $D_t^{\mathrm{v}}$, while fitness is estimated using the fitness estimation data set $D_t^{\mathrm{f}}$ [13]. Overall fitness is defined as the highest classification rate achieved over the five pattern presentation orders, and $FAM_{\mathrm{est}}$ is the network that yields this highest classification rate.

**Algorithm 1.** DPSO learning algorithm.

```
Inputs: New data sets $D_t$ for learning.
Outputs: Pool (or swarm) of $N$ FAM networks $FAM_n$.

      Initialization:
 1:   • Set the swarm's parameters,
      • Initialize all $N$ networks $FAM_n$ and $FAM_n^{\mathrm{start}}$,
      • Set PSO iteration counter at $\tau=0$, and
      • Randomly initialize particles positions and velocities.

      Upon reception of a new data block
      $D_t$, the following incremental process is initiated:

 2:   for each particle $n$, where $1 \le n \le N$ do
 3:     Train and validate $FAM_n$ with $D_t^{\mathrm{t}}$ and $D_t^{\mathrm{v}}$
        respectively, and estimate $f(\mathbf{h}_n^*,t)$ using $D_t^{\mathrm{f}}$.
 4:   end for
 5:   while PSO does not reach stopping condition do
 6:     Update particle positions according to the DNPSO
        algorithm.
 7:     for each particle $n$, where $1 \le n \le N$ do
 8:         $FAM_{\mathrm{est}} \leftarrow FAM_n^{\mathrm{start}}$
 9:         Train $FAM_{\mathrm{est}}$ with validation using $D_t^{\mathrm{t}}$ and $D_t^{\mathrm{v}}$, and
            estimate $f(\mathbf{h}_n(\tau),t)$ using $D_t^{\mathrm{f}}$.
10:         if $f(\mathbf{h}_n(\tau),t) > f(\mathbf{h}_n^*,t)$ then
11:           $\{\mathbf{h}_n^*,FAM_n,f(\mathbf{h}_n^*,t)\} \leftarrow \{\mathbf{h}_n(\tau),FAM_{\mathrm{est}},f(\mathbf{h}_n(\tau),t)\}$
12:         end if
13:     end for
14:     $\tau=\tau+1$
15:   end while
16:   for each particle $n$, where $1 \le n \le N$ do
17:     $FAM_n^{\mathrm{start}} \leftarrow FAM_n$
18:   end for
```

During the initialization process (line 1), all the FAM networks are initialized, and the swarm's parameters are set. Particle positions are then randomly initialized within their allowed range. When a new $D_t$ becomes available, the optimization process begins. Fitness associated with the best position of each particle, $f(\mathbf{h}_n^*,t)$, is updated according to the new data along with each network $FAM_n$ (lines 2–4). The optimization process continues were it previously ended until the DNPSO algorithm converges (lines 5–15). The DNPSO algorithm changes the position of subswarms and free particles in the hyperparameter space, and iteratively update each particle's new position along with their fitness (lines 6–13). If new personal best positions are found, the position ($\mathbf{h}_n^*$), fitness ($f(\mathbf{h}_n^*,t)$), and network associated with the personal best $FAM_n$ are updated (lines 10–12). At each iteration $\tau$, in the cases of equality between $f(\mathbf{h}_n(\tau),t)$ and $f(\mathbf{h}_n^*,t)$, the network that requires the least resources ($F_2$ nodes) is kept. Finally, the iteration counter $\tau$ is incremented (line 14).

Once the DNPSO algorithm converges, the $FAM_n$ networks associated to each personal best are stored as $FAM_n^{\mathrm{start}}$ (lines 16–18). These networks provide a short term memory of the swarm's state after learning data block $D_t$. When new data becomes available at a time $t+1$, the best network previously obtained at time $t$ ($FAM_n^{\mathrm{start}}$) serves as the initial condition, and is copied to $FAM_{\mathrm{est}}$ prior training on $D_{t+1}$ each time the fitness of particle $n$ is estimated. For the first learning block $D_1$, the $FAM_n^{\mathrm{start}}$ networks are in an initial state (see Section 2.1).

### 3.2. Selection of diversified ensembles

Once the pool of classifiers has evolved (Algorithm 1), members of this pool are selected to form a heterogeneous ensemble, where each network is trained on the same data, but with different hyperparameters [52]. In Algorithm 2, DNPSO capabilities to detect several local optima, while maintaining particle diversity in the hyperparameter space, are exploited for a selection of heterogeneous ensembles, driven by accuracy and diversity, that does not require computing costly classifier diversity indicators [6,24,27,37,45,50]. Indeed, those indicators involve computing the FAM choice functions $T_j(\mathbf{A})$ of all the networks over the fitness estimation data set $D_t^{\mathrm{f}}$. In a worse case scenario, the hyperparameter values are set to grow the largest possible FAM network such as $J_n = |D_1^{\mathrm{t}} \cup \cdots \cup D_t^{\mathrm{t}}|$. Since diversity indicators rely on classifiers disagreements in the decision space, this time complexity is $O(J_n \cdot |D_t^{\mathrm{f}}| \cdot I)$, where $|D_t^{\mathrm{f}}|$ is the size of the fitness estimation data set, and $I$ is the number of input features. In contrast, selecting ensembles in the hyperparameter space represent a less costly approach. With DPSO algorithms, diversity in the hyperparameter space involves computing the Euclidean distances between the personal best position of each particle. The time complexity of this operation is $O(N^2)$, where the size of the swarm $N$ is generally smaller than the number of nodes $J_n$, size of the fitness estimation data set $|D_t^{\mathrm{f}}|$, and the number of input features $I$.

Prior to selection, the ensemble of FAM networks (*EoFAM*) is empty (line 1). Selection is initially performed based on accuracy and diversity (lines 2–4). During this phase, the ensemble then consists of the networks corresponding to detected local optima in the optimization environment, *i.e.* personal best position of the DNPSO masters. Not only this ensures that the initial ensemble consist of networks that are locally the most accurate in the swarm, but since DNPSO forces a minimal Euclidean distance between masters, it ensures that this ensemble is also diverse.

**Algorithm 2.** Ensemble selection based on FAM accuracy and particle diversity.

```
Inputs: A swarm of $N$ networks associated with DPSO particles.
Outputs: A diverse heterogeneous ensemble of FAM networks (EoFAM).

    Initialization:
1:  $EoFAM \leftarrow \emptyset$
    Selection of the FAM networks associated to detected local optima:
2:  for $e=1$ to $N_{ss}$, the number of subswarms do
3:    $EoFAM \leftarrow FAM_n$ associated to master particle $e$
4:  end for
    Second selection aimed to maximize particle swarm diversity using greedy search:
5:  Compute initial swarm diversity $\overline{\delta_{e_1 e_2}}$ for the $N_{ss}$ networks in $EoFAM$ using Eq. (5).
6:  for $e=1$ to $N-N_{ss}$ do
7:    for all networks that are not part of ensemble do
8:      Find the one that maximizes swarm diversity $\overline{\delta_{e_1 e_2}}$ for the $N_{ss}+e$ networks in $EoFAM$.
9:    end for
10:   if there exist no networks such as $\overline{\delta_{e_1 e_2}}$ increases, then
11:     BREAK;
12:   else
13:     $EoFAM \leftarrow FAM_n$ associated to the particle that maximized $\overline{\delta_{e_1 e_2}}$.
14:   end if
15: end for
```

The second phase of selection seeks to further increase ensemble diversity by using a greedy search that maximizes particle diversity. For two classifiers $e_1$ and $e_2$, the pairwise diversity between their particles, $\delta_{e_1 e_2}$, is defined as the Euclidean distance in the hyperparameter space between those particles. For EoFAM, diversity in the hyperparameter space is then defined by the average value of all Euclidean distances

$$\overline{\delta_{e_1 e_2}} = \frac{2}{E(E-1)} \sum_{e_1=1}^{E-1} \sum_{e_2=e_1+1}^{E} \delta_{e_1 e_2}, \tag{5}$$

where $E$ is the number of networks in the ensemble. Although computing this particle swarm diversity has a time complexity of $O(N^2)$, it was revealed to be the most accurate [38]. Moreover, compared to training all the FAM network during fitness estimation (in Algorithm 1), the computation of $\overline{\delta_{e_1 e_2}}$ (Eq. (5)) is an insignificant component in the overall time complexity.

Ensemble diversity computed after the first selection process (lines 2–4) and the greedy search is performed (lines 6–15). Algorithm 2 iteratively scans through all particles that are not part of the ensemble to find those that maximizes swarm diversity $\overline{\delta_{e_1 e_2}}$ (lines 7–9). If no particle can raise diversity, Algorithm 2 stops (line 11). Otherwise, the network $FAM_n$ associated to the winning particle is added to the ensemble (line 13). Although a greedy search is not guaranteed to find the global best solution, it is a monotonic increasing search process that is efficient in practice [50]. Greedy search has a time complexity of $O(N^2)$, compared to an exhaustive search that has an exponential time complexity of $O(2^N)$.

Once the selection process is complete, the fusion of responses from selected classifiers is performed using a simple majority vote. In the case of a tie, simpler FAM networks are favored—the class is predicted by the networks that require the fewest overall number of $F_2$ nodes wins the vote.

## 4. Experimental methodology

The main problem addressed in this research is the design of accurate and efficient adaptive systems for the classification of faces in video streams. Biometric systems for the recognition of faces in video streams are relevant in different scenarios, ranging from to open-set video surveillance or screening applications, where criminals or terrorists enrolled to a watch list must be recognized within dense and moving crowds at major events and airports, to closed-set access control applications, where individuals enrolled to system must by identified prior to accessing secured resources. Other applications involve identification at access control points, verification of laptop or cell phone users, etc. In this section, a general system for face recognition in video is first described, followed by the data bases, incremental learning scenarios, and experimental protocol used to evaluated the performances of the proposed DPSO-based incremental learning strategy. Finally, the protocol employed to analyze the relationship between particle diversity in the hyperparameter space versus classifier diversity in the feature and decision spaces is described, followed by the performance indicators.

### 4.1. Application—face recognition in video

It is assumed that 2D images in the video streams of an external 3D scene are captured using one or more IP or network cameras with fast Ethernet interface, and that computer analysis is performed at a distance. Each camera captures a sequence of 2D images, or frames, from the external scene, and each frame provides the system with a particular view of individuals populating the scene. First, the system performs segmentation to locate and isolate regions of interest (ROIs) corresponding to the faces in a frame.

From the ROIs, features are extracted for 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 [21]. On the other hand, classifiers will require invariant and discriminant classification features extracted from the ROIs, and mapped to an $\mathbb{R}^I$ input feature space.

The tracking module generally follows the movement or expression of faces across video frames, while the classification module seeks to match input feature patterns to the face models of individuals enrolled to biometric the system. Biometric matching is typically implemented with a statistical or neural pattern classifier. With neural network classifiers, for instance, the biometric model of individuals is defined using the hyperparameters, synaptic weights, and architecture (determined in Algorithms 1). Finally, for each video frame, the decision module may combine and accumulate the responses from the tracking and classification modules. Given a video sequence threshold, and assuming that tracking is ideal, the frames are presented to the face recognition system and predictions for each ROIs are accumulated over time. With FAM networks, each prediction consists in a binary vector with one for the predicted class, and zero elsewhere. After a given number of video frames, prediction for the sequence is the class with the highest accumulated response. For identification and surveillance applications, the accumulated response is used as a classification score and the result is a list of the most likely or of all possible matching identities, respectively.

Several powerful techniques have been proposed to recognize faces in static 2D images [56]. A common approach to recognize faces in video consists in exploiting only spatial information

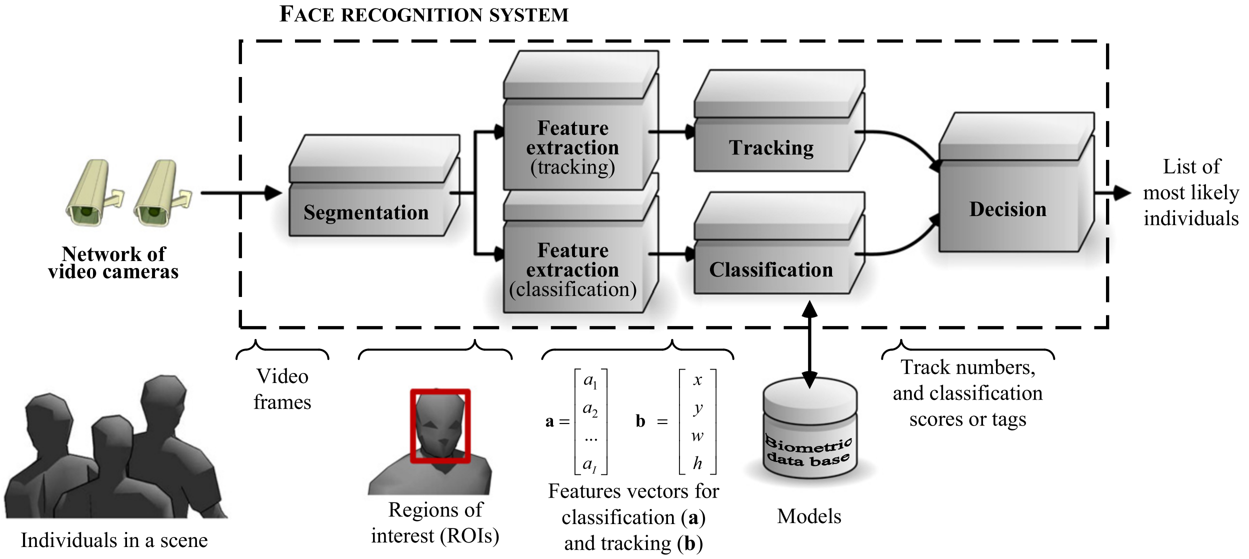
FACE RECOGNITION SYSTEM
Network of video cameras
Segmentation
Feature extraction (tracking)
Feature extraction (classification)
Tracking
Classification
Decision
List of most likely individuals
Individuals in a scene
Video frames
Regions of interest (ROIs)
$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_I \end{bmatrix}$ $\mathbf{b} = \begin{bmatrix} x \\ y \\ w \\ h \end{bmatrix}$
Features vectors for classification (a) and tracking (b)
Biometric data base
Models
Track numbers, and classification scores or tags

**Fig. 6.** A generic track-and-classify biometric system for video-based face recognition.

(*i.e.*, appearance), and applying extensions of static image-based techniques on high quality face images produced through segmentation. The predominant techniques are appearance-based methods like Eigenfaces, and feature-based methods like Elastic Bunch Graph Matching [56]. More recently, some authors have exploited temporal information contained in video sequences to improve performance of video-based face recognition. For example, track-and-classify systems (as the one shown in Fig. 6) combine spatial information with information on motion and appearance of faces in a scene [13]. Regardless, the performance of these techniques may degrade considerably when applied in real-world applications.

In addition to difficulties mentioned earlier, video-based face recognition remains a very challenging problem since faces captured in video frames are typically low quality and generally small. Moreover, there are limitations associated with the camera and techniques used for segmentation, scaling, filtering, feature extraction, and classification (*e.g.*, resolution and noise) [19,34,57].

### 4.2. Video data bases

In this paper, experiments are performed by applying AMCS to video-based face recognition in a closed-set access control (identification) applications. Proof-of-concept simulations are performed with two real-world video data bases for face recognition.

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 × 120 resolution that was mounted on a computer monitor. Of the two video sequences, one is dedicated to training and the other to testing. They are taken under approximately the same illumination conditions, the same setup, almost the same background, and each face occupies between 1/4 and 1/8 of the image. This data base contains a variety of challenging operational conditions such as motion blur, out of focus factor, facial orientation, facial expression, occlusion, and low resolution. The number of ROIs detected varies from class to class, ranging from 40 to 190 for one video sequences.

The second video data base is called Motion of Body (MoBo), and was collected at Carnegie Mellon University under the HumanID project [22]. Each video sequence shows one of 25 different individuals on a tread-mill so that they move their heads naturally to four different motion types when walking: slowly, fast, on an inclined surface, and while carrying an object. Six Sony DXC 9000 cameras, with a resolution of a 640 × 480 pixels, are positioned at different locations around the individuals. Only the video sequences with visible faces were kept: full frontal view and both sides with an angle of about 70° with the full frontal view.

In both cases, segmentation is performed using the Viola–Jones algorithm included in the OpenCV C/C++ computer vision library. For the IIT-NRC database, the small regions of interest (ROIs) produced are converted in gray scale and normalized to 24 × 24 images where the eyes are aligned horizontally, with a distance of 12 pixels between them. Principal Component Analysis is then performed to reduce the number of features. The 64 features with the greatest eigenvalues are extracted and vectorized into $\mathbf{a} = \{a_1, a_2, \dots, a_{64}\}$, where each feature $a_i \in [0,1]$ are normalized using the min–max technique. Learning is done with ROIs extracted from the first series of video sequences (1527 ROIs) while testing is done with ROIs extracted from the second series of video sequences (1585 ROIs). The ROIs obtained with the MoBo data base where processed with Local Binary Pattern and Principal Component Analysis to produce 32 features vectors, also normalized using the min–max technique. ROIs from sequences for each type of walk and view are divided in two; the first half is used for learning and the second half, for testing. This yields a total of 36374 learning patterns and 36227 test patterns. In both cases, the number of features was fixed after error convergence with a 1NN classifier trained on the learning data bases and tested on the test data base.

### 4.3. Incremental learning scenarios

Prior to computer simulations, each video data set is divided in blocks of data $D_t$, where $1 \le t \le T$, to emulate the availability of $T$ successive blocks of training data to the AMCS during a biometric identification application. 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 AMCS performance for $K$ classes, the first learning block $D_1$ is composed of two classes, and each successive block $D_t$, where $2 \le t \le K-1$, 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 the invariance of results to class presentation orders, 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 refine its internal models. To assess AMCS 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.

### 4.4. Experimental protocol

To illustrate (1) the performance of Algorithm 1 with different ensemble selection methods and (2) the impact of diversity in the optimization environment on the models in the classification environment, the results of two experiments are presented in this paper. The performance of the proposed DPSO-based learning strategy is first evaluated and compared with various techniques to generate and select classifiers during supervised incremental learning of data blocks $D_t$. Secondly, the correlation between particle diversity among particles in a swarm (in the hyperparameter) space and diversity among classifiers in an ensemble (in the feature and decision space) is shown empirically.

The DNPSO parameters used for both experiments are shown in Table 1. Weight values $\{w_1,w_2\}$ were defined as proposed in [28], and to detect a maximal number of local optima, no constraints were considered regarding the number of subswarms. Since Euclidean distances between particles are measured with the DNPSO algorithm, the swarm evolves in a *normalized* $\mathbb{R}^4$ space to avoid any bias due to the domain of each hyperparameter. Before being applied to FAM, particle positions are denormalized to fit the hyperparameters domain. For each new blocks of data $D_t$, the DPSO optimization process is set to either stop after 10 iterations without improving the classification rate of the best FAM network ($FAM_{n*,t}$) classification rate, or after maximum 100 iterations.

Learning is performed over ten trials using ten-fold cross-validation with the LTM used as specified in [13]. The proportion of $D_t$ assign to the LTM, and the maximal number of patterns for each class present in the LTM, are respectively set to $\lambda_D = 1/6$ and $|LTM|_k = 20$. Out of the 10 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 the LTM to estimate the fitness of each particle during the DPSO algorithm ($D_t^{\mathrm{f}}$). Between successive training epochs, the presentation order of training patterns is changed randomly. Within each trial, five different replications are performed using different class presentation order, for a total of 50 replications.

**Table 1**
DNPSO parameters.

| Parameter | Value |
|---|---|
| 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 |
| Neighborhood size | 5 |
| Minimal distance between two masters | 0.1 |
| Minimal velocity of free particles | 0.0001 |

The simulations evaluate the performance achieved during both incremental learning scenarios of new data blocks $D_t$, where AMCSs employ the DPSO-based strategy proposed in Section 3 (Algorithms 1 and 2) to evolve heterogeneous ensemble indicated by LBESTS$_{+\mathrm{d}}$ (local best particles combined with the diversity greedy search). This system is compared to AMCSs using the DPSO-based strategy but with different ensemble selection techniques, in particular:

- GREEDY$_{\mathrm{a}}$ ← the ensemble of FAM networks found using greedy search based on accuracy [50],
- SWARM ← the ensemble of FAM networks build with the entire swarm, and
- GBEST ← the FAM network corresponding to the DPSO global best solution.

For references, the performance is also given for an AMCS that uses the entire swarm of FAMs trained with a canonical PSO batch learning strategy [20] (PSO$_{\mathrm{B}}$), and a single *k*NN classifier that also performs batch learning. At a given time *t*, batch learning consist of initializing the system, and learning all the data blocks $D_t$ accumulated so far, $B_t = D_1 \cup \cdots \cup D_t$ [20].

Additional experiments (presented in Fig. 7) verify the hypothesis under which particle diversity in the optimization environment is correlated to that of ensemble classifier diversity, where each classifier is associated to a particle. Experiments are performed in two steps: (1) optimization during supervised batch learning of the whole IIT-NRC data base with the DPSO learning strategy, and (2) particles expansion.

Prior to optimization of hyperparameters, the normalized search space is bound by a constraint of 0.2. Once the global best hyperparameter values are found, an ensemble is formed with 17 FAM networks, each one associated with a particle organized into a hypercube centered around the global best in the normalized $R^4$ hyperparameter space. One particle is centered at the global best while the other 16 ($2^4$) are positioned as a 4 dimensions hypercube around the center. To change the diversity level, all particles are initially situated at the same position of the global best, and the size of the hypercube is gradually expanded up to the value of the constraint to form different swarms (each noted by a different color in Fig. 7b). The expansion of this hypercube will affect a change on diversity in the hyperparameter space.

### 4.5. Performance evaluation and diversity indicator

The average performance of AMCSs is assessed in terms of classification rate over a sequence of one or more ROIs, and resource requirements. The *classification rate for single facial images (ROIs)* is the ratio of correct predictions over all test set predictions, where each ROIs is tested independently. Note that classification decisions produced for a single image are considered to be the most conservative performance metric, and it is used for fitness estimation in Algorithms 1 and 2. However, for the video-based face recognition application, *classification rate for video sequences* (over two or more ROIs), the result of the fusion between the tracking and classification module, is used. Given video sequences, it is the ratio of correct predictions over all predictions made by the AMCS accumulated response over a fixed number of video frames. For unbalanced data bases (*i.e.*, video sequences of different length), classification rate for a number of frames exceeding the length of shorter sequences are computed with predictions obtained with all

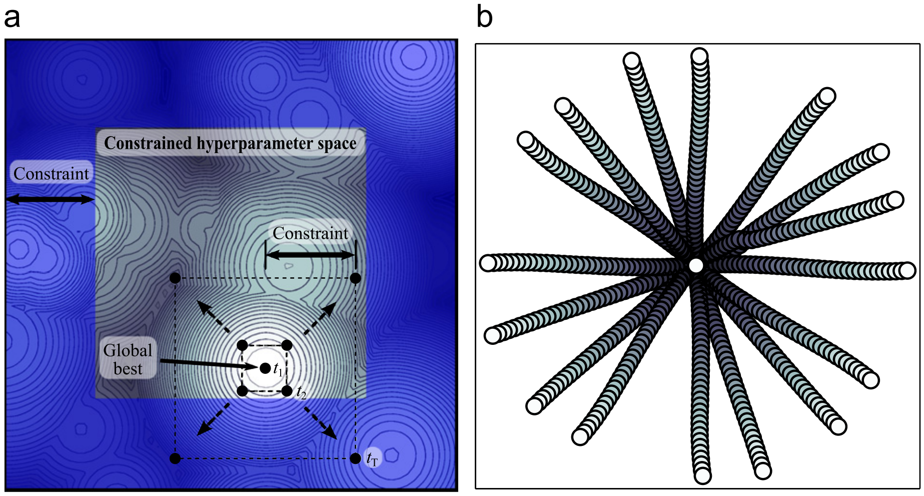

**Fig. 7.** Example of the particle positions for a 2D objective function (a) and 2D projection, obtained using Sammon's mapping of the particle positions in the $R^4$ hyperparameter space (b). The swarm is organized into a hypercube centered around the global best in the in the normalized $R^4$ hyperparameter space. The hypercube gradually expands, linearly changing particle diversity and affecting the corresponding ensemble of classifiers. (a) 2D example; (b) 2D projection.

ROIs of the latter. The accuracy of AMCSs is also evaluated with cumulative match curves (CMC) [35]. These curves estimate the ranking capabilities of a classification system for identification applications by providing a cumulative a posteriori probability estimation of having a correct prediction according to rank.

Resource requirements of AMCSs that employ the DPSO learning strategy is measures in terms of *compression*. That is, the average number of training patterns, contained in all $D_t^t$ presented to the AMCS, per category prototype in the classifier. For FAM networks, compression refers to the average number of training patterns per neuron in the $F_2$ layer. For ensembles, it is the total number of $F_2$ layer nodes for all classifiers in the ensemble. Since learning with $k$NN consist of memorizing the training data set $D_t^t$, compression in this case is always one.

While particle swarm diversity is computed using Eq. (5), three pairwise indicators are used to compute correlation, or diversity, between two ensemble's classifiers $e_1$ and $e_2$. As with most measures present in literature, the $Q$ statistic and the correlation coefficient [50] rely on classifier disagreement to compute correlation among classifiers. On the other hand, a specialized ambiguity indicator, inspired by margin theory [47], is used to compute FAM network diversity. For two ensemble classifiers $e_1$ and $e_2$, and a given data set (in our case the fitness estimation data set $D_t^f$), each pairwise indicator is computed as followed:

1. *The Q statistic*

$$Q_{e_1e_2} \in [0,1] = \frac{N_{11}N_{00} - N_{10}N_{01}}{N_{11}N_{00} + N_{10}N_{01}}, \tag{6}$$

where $N_{11}$, $N_{00}$, $N_{10}$, and $N_{01}$ are the number of patterns for each combination of correct and incorrect predictions by classifiers $e_1$ and $e_2$ on the given data set (see Table 2).

2. *Correlation coefficient*

$$\rho_{e_1e_2} \in [0,1] = \frac{N_{11}N_{00} - N_{10}N_{01}}{\sqrt{(N_{11}+N_{10})(N_{01}+N_{00})(N_{11}+N_{01})(N_{10}+N_{00})}}, \tag{7}$$

3. *Specialized ambiguity indicator for FAM networks* Given a pattern **a**, the FAM network selects the $F_2$ winning node $j^*$, corresponding to the highest choice function $T_j(\mathbf{a})$, and predicts class $k(j^*)$. FAM ambiguity is defined by

$$\tau \in [0,1] = T_{j^*}(\mathbf{a}) - \max(T_j(\mathbf{a}), k \neq k^*). \tag{8}$$

**Table 2**
Contingency table used to compute diversity among ensemble classifiers with the $Q$ statistic and correlation coefficient.

| | $FAM_{e_2}$ correct | $FAM_{e_2}$ incorrect |
|---|---|---|
| $FAM_{e_1}$ correct | $N_{11}$ | $N_{10}$ |
| $FAM_{e_1}$ incorrect | $N_{01}$ | $N_{00}$ |

Diversity between two FAM classifiers $e_1$ and $e_2$ is defined by the sum of the ambiguity differences for all patterns in the fitness estimation data set ($|D_t^f|$)

$$\Delta\theta_{e_1e_2} \in ]0, |D_t^f|] = \sum^{|D_t^f|} |\tau_{e_1} - \tau_{e_2}|. \tag{9}$$

Ensemble diversity is then defined as the average value deprived from all combination of the pairwise classifier diversity indicators, computed in the same manner as Eq. (5). Measures from Eqs. (6), (7), and (9) are noted: $\overline{Q_{e_1e_2}}$, $\overline{\Delta\rho_{e_1e_2}}$, and $\overline{\Delta\theta_{e_1e_2}}$. Higher ensemble diversity is observed for low correlation indicators values ($\overline{Q_{e_1e_2}}$ and $\overline{\Delta\rho_{e_1e_2}}$) and for high values of the FAM diversity indicator ($\overline{\Delta\theta_{e_1e_2}}$).

## 5. Results and discussion

### 5.1. Performance for single images (ROIs)

To assess the performance of ensembles evolved using the DPSO-based learning strategy, Figs. 8 and 9 present the average classification rate obtained with single facial regions of interest (ROIs), compression, and ensemble size achieved versus the number of data blocks $D_t$. Results obtained after learning all IIT-NRC and MoBo data bases with both learning scenarios are shown in Tables 3 and 4. For both incremental learning scenarios, results are shown for AMCSs that employs the DPSO-based strategy proposed in Section 3 (LBESTS$_{+\text{d}}$). It is compared to ensembles formed with the entire swarm of FAM networks (SWARM) and the global best network only (GBEST). In all cases, the accuracy-based greedy search [50] (GREEDY$_\text{a}$) is unable to improve the recognition capabilities of the single global best network (GBEST). Its performances are thus not shown. For reference, performance is also

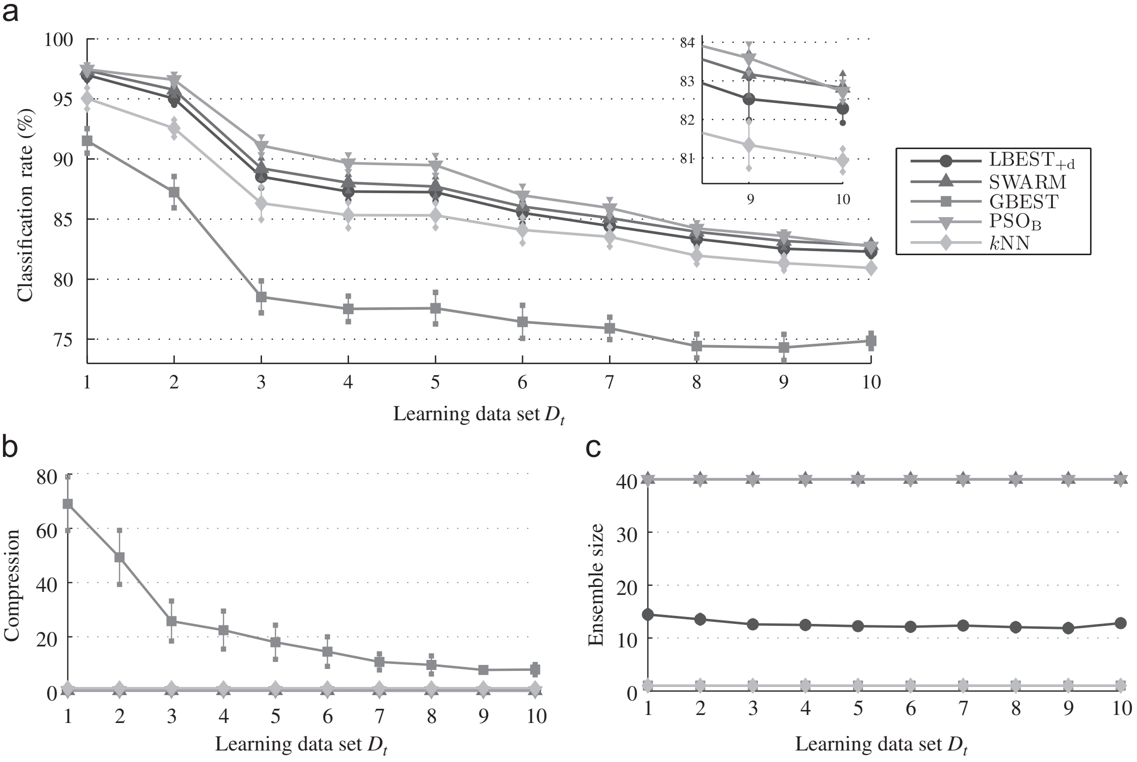

**Fig. 8.** Average classification rate, compression, and ensemble size of the AMCS versus blocks of IIT-NRC data learned during the enrollment scenario. Performance was evaluated during incremental learning for the AMCS with different ensemble selection techniques and the global best network alone (GBEST). The performance of the whole swarm optimized during batch learning (PSO$_B$) and $k$NN are shown for reference. Error bars correspond to the 90% confidence interval.

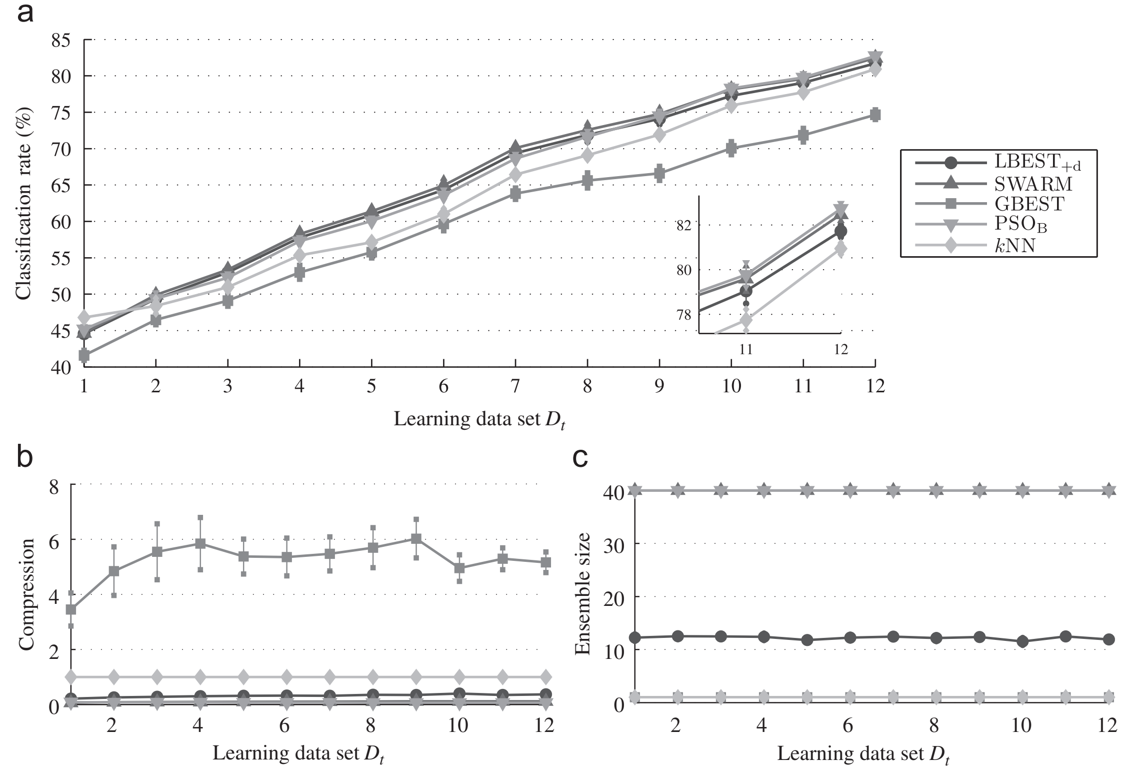

**Fig. 9.** Average classification rate, compression, and ensemble size of the AMCS versus blocks of IIT-NRC data learned during the update scenario. Performance was evaluated during incremental learning for the AMCS with different ensemble selection techniques and the global best network alone (GBEST). The performance of the whole swarm optimized during batch learning (PSO$_B$) and $k$NN are shown for reference. Error bars correspond to the 90% confidence interval.

**Table 3**
Average classification rate (in percentage), compression and ensemble size after incremental learning of all the IIT-NRC and MoBo data bases for the enrollment scenario. Each cell is presented with the 90% confidence interval.

| Type of learning method | Incremental | | | Batch | |
|---|---|---|---|---|---|
| | **LBESTS$_{+\mathrm{d}}$** | SWARM | GBEST | PSO$_{\mathrm{B}}$ | *k*NN |
| **IIT-NRC data base** | | | | | |
| Classification rate (%) | 82.3 ± 0.4 | 82.8 ± 0.4 | 74.9 ± 0.6 | 82.7 ± 0.2 | 80.9 ± 0.3 |
| Compression | 0.38 ± 0.03 | 0.13 ± 0.01 | 8 ± 2 | 0.062 ± 0.003 | 1 ± 0 |
| Ensemble size | 12.8 ± 0.6 | 40 ± 0 | 1 ± 0 | 40 ± 0 | 1 ± 0 |
| **MoBo data base** | | | | | |
| Classification rate (%) | 92 ± 2 | 91 ± 5 | 89.2 ± 0.7 | 94.9 ± 0.1 | 94.5 ± 0.1 |
| Compression | 1.3 ± 0.1 | 0.48 ± 0.02 | 9.0 ± 0.9 | 0.09 ± 0.02 | 1 ± 0 |
| Ensemble size | 12.1 ± 0.4 | 40 ± 0 | 1 ± 0 | 40 ± 0 | 1 ± 0 |

**Table 4**
Average classification rate (in percentage), compression and ensemble size after incremental learning of all the IIT-NRC and MoBo data bases for the update scenario. Each cell is presented with the 90% confidence interval.

| Type of learning method | Incremental | | | Batch | |
|---|---|---|---|---|---|
| | **LBESTS$_{+\mathrm{d}}$** | SWARM | GBEST | PSO$_{\mathrm{B}}$ | *k*NN |
| **IIT-NRC data base** | | | | | |
| Classification rate (%) | 81.7 ± 0.3 | 82.5 ± 0.3 | 74.7 ± 0.7 | 82.7 ± 0.3 | 80.9 ± 0.3 |
| Compression | 0.36 ± 0.02 | 0.11 ± 0.01 | 5.1 ± 0.4 | 0.062 ± 0.003 | 1 ± 0 |
| Ensemble size | 11.9 ± 0.5 | 40 ± 0 | 1 ± 0 | 40 ± 0 | 1 ± 0 |
| **MoBo data base** | | | | | |
| Classification rate (%) | 92.8 ± 0.3 | 95 ± 3 | 87 ± 2 | 94.9 ± 0.1 | 94.5 ± 0.1 |
| Compression | 1.1 ± 0.1 | 0.37 ± 0.02 | 12 ± 2 | 0.09 ± 0.01 | 1 ± 0 |
| Ensemble size | 13.0 ± 0.8 | 40 ± 0 | 1 ± 0 | 40 ± 0 | 1 ± 0 |

shown for batch learning with an ensemble formed with the entire swarm (PSO$_{\mathrm{B}}$) and the $k$ nearest neighbors algorithm ($k$NN). For face recognition on single ROIs from the IIT-NRC data base, Arandjelovic et al. [1] was able to achieve a classification rate of 91%, while Gorodnichy [19] obtained a classification of 80%. In both cases, batch learning was performed with settings in [19]; that is, the features are vectorized as unprocessed gray scale values of the $24 \times 24$ images and one class was used to verify the false acceptance rate rather than the classification rate. No such results are available for the MoBo data base.

For the enrollment scenario, only two classes are present at the beginning of the learning process. Class decision boundaries are initially simple and classification rates are high. As classes are added, these boundaries become more complex, leading to a decline in classification rate (see Fig. 8). When the AMCS is used with the global best only (GBEST), compression also diminishes considerably. While this is also true when the AMCS uses batch learning (PSO$_{\mathrm{B}}$), compression and ensemble size for the AMCS with both ensembles methods remains stable during incremental learning.

As expected, after training on all data, GBEST gives the lowest classification rate, while all other solutions give classification rates between 81% and 83% (see Fig. 8, Table 3). Using LBESTS$_{+\mathrm{d}}$ gives classification rate comparable to that of SWARM throughout all the enrollment process, except after learning new blocks at times $t=2$ and $t=10$. However, as compression and ensemble size show (in Table 3), LBESTS$_{+\mathrm{d}}$ is able to achieve this accuracy with a third of the resources.

Results with the MoBo data base are consistent with those obtained with the IIT-NRC data (see Table 3). However, since position of individuals and video cameras used in the MoBo protocol are fixed, data acquisition is more constrained than with the IIT-NRC data base. Class distributions are more compact and less likely to vary significantly from one block to the next. Classification rate and compression follow similar trends excepted that they are generally higher than with the IIT-NRC data base.

The overall phenomena observed during enrollment resemble the performance observed for the update scenario. The main difference is that all class distributions are defined from the outset, in $D_1$, and the AMCS initially has knowledge of the entire classification problem. Due to limited learning data, knowledge of the problem is however incomplete and decision boundaries in the input feature space are then poorly defined, leading to a low classification rate (see Fig. 9). As classes are updated incrementally, accuracy of the face recognition system tends to increase. The highest classification rate are again obtained with LBESTS$_{+\mathrm{d}}$ and SWARM (Table 4). Although the AMCS with LBESTS$_{+\mathrm{d}}$ uses about one third of resources used by SWARM, both selection techniques have comparable accuracy over all data blocks, except at times $t=\{10,12\}$.

Two differences are observed at the beginning of the learning process with batch learning methods. When using the AMCS with batch learning, the LTM is unnecessary, and all the cumulative data from successive blocks is directly assigned to the training, validation, and fitness estimation ($D_t^{\mathrm{t}}$, $D_t^{\mathrm{v}}$, and $D_t^{\mathrm{f}}$). Therefore, fewer data samples are used for validation during network training and fitness estimation on the objective function, leading to a lower classification rate than those obtained with the LTM. Secondly, in contrast with FAM, where the Webber Law choice function computes city block distances ($L^1$ norm), $k$NN instead computes Euclidean distances ($L^2$ norm). It only relies on validation data only to set the value of $k$, and does not perform sequential learning (*i.e.*, it is not sensitive to patterns order presentation). As such, it performs well if only few samples are available to define decision boundaries in a complex classification environment where all classes are defined. However, it must store all cumulative training data in memory, and requires a greater time complexity for matching input patterns to an output class.

Indeed, to perform predictions, FAM networks complement code the $I$ features, computes the choice function for the $J$ category prototypes in the ensemble, and find the best for each FAM, a time complexity of $O(2IJ)$. On the other hand, $k$NN computes the Euclidean distance for each $J$ category and ranks the solutions to find the best $k$, a time complexity of $O(kIJ\log(J))$. For equal compression values, matching an input pattern to a class is thus a simpler task with a FAM classifiers, and the difference between the two increases over time, has more category prototype are include in the AMCS.

Results with the IIT-NRC data are once again confirmed by those obtained with the MoBo data (see Table 4). As with the enrollment scenario, class distributions are more compact and both classification rate and compression are higher. However, updating classes through batch learning yields a higher classification rate at the expense of lower compression.

With DNPSO parameters presented in Table 1, the AMCS was able to find on average $6.4 \pm 0.1$ local maxima (with the 90% confidence interval) during the enrollment scenario, and $6.3 \pm 0.1$ for the update scenario. Using the greedy search to select classifiers that maximize particle diversity (LBESTS$_{+d}$) nearly doubles the average number of classifiers used in the ensembles to $12.7 \pm 0.2$ and $12.2 \pm 0.1$ FAM networks, respectively (see Figs. 8c, 9c). These ensembles yield a classification rate comparable to that of the AMCS with SWARM, and are efficiently obtained by maximizing particle diversity in the hyperparameter space. For instance, if the greedy search process where driven by classifier diversity in the classification environment, this would involve the costly computation of the Weber function (Eq. (1)) of every nodes of all FAM networks over each $D_t^f$ pattern every time a network is added to the ensemble by Algorithm 2.

Note that, to initially find more local optima, the ratio $|\text{swarm}|/|\text{neighborhood}|$ could be raise. But whereas large swarms would lead to a large number of fitness evaluation, unnecessarily slowing the DPSO optimization process, small neighborhoods sizes leads local optima detection that is very sensitive to noise on the objective function. The choice of those (DNPSO) parameters is problem dependent.

### 5.2. Performance for video-streams (multiple ROIs)

For video-based face recognition, classification is typically performed by accumulating the response of a classifier over several video frames. For both scenarios, Fig. 10 presents the evolution of the classification rate for video sequences achieved by the proposed system (LBESTS$_{+d}$) as a function of the number of ROIs used to perform identification, and Fig. 11 shows the cumulative match curves (CMC) for different number of ROIs used to perform identification. Table 5 presents the number of ROIs necessary to achieve an average classification rate statistically comparable to 100%, for all tested cases and both data bases. Comparison with other video-based face recognition systems from the literature is presented in Table 6 for both IIT-NRC and MoBo data bases.

As Fig. 10 shows, the video-based classification rate for both scenarios follow the same trends as when the system is tested with single ROIs (Figs. 8, 9). When classes are enrolled incrementally (Fig. 10a), class decision boundaries becomes more complex

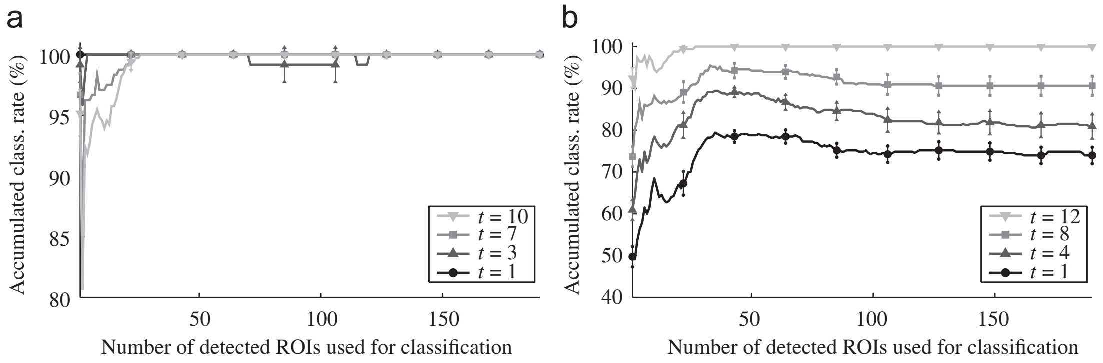

**Fig. 10.** Evolution of the average classification rate for video sequences of the AMCS's ensemble versus the number of ROIs used to identify individuals of the IIT-NRC data base. Performance is shown for incremental learning under both scenarios for the AMCS with LBESTS$_{+d}$. Error bars correspond to the 90% confidence interval. (a) Enrollment scenario, (b) update scenario.

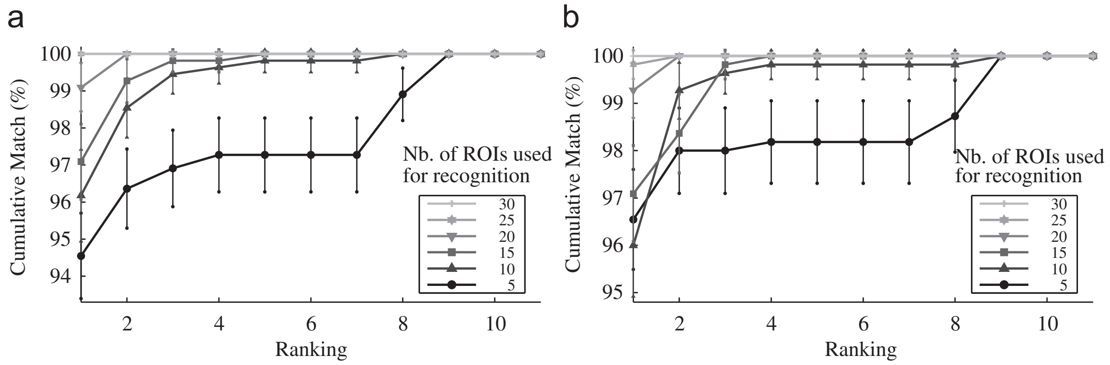

**Fig. 11.** Cumulative match curves the AMCS's ensemble for different number of ROIs used to perform face recognition. Performance is shown after incremental learning of all the IIT-NRC data base, under both scenarios for the AMCS with LBESTS$_{+d}$. Error bars correspond to the 90% confidence interval. (a) Enrollment scenario, (b) update scenario.

**Table 5**
Number of ROIs necessary to achieve a classification rate comparable to 100% for video-based face recognition after learning the entire IIT-NRC and MoBo data bases through both incremental learning scenarios with the AMCS.

| Type of learning method | Incremental | | | Batch | |
|---|---|---|---|---|---|
| | **LBESTS$_{+d}$** | SWARM | GBEST | PSO$_B$ | *k*NN |
| **IIT-NRC data base** | | | | | |
| Number of ROIs during enrollment | 24 | 23 | never | 19 | 24 |
| Number of ROIs during update | 25 | 20 | never | 19 | 24 |
| **MoBo data base** | | | | | |
| Number of ROIs during enrollment | 15 | 30 | 16 | 32 | 16 |
| Number of ROIs during update | 27 | 25 | 16 | 32 | 16 |

**Table 6**
Comparison of the DPSO-based learning strategy with other authors on the IIT-NRC and MoBo data bases. Classification rates where obtained for recognition on video sequences.

| | | | | | |
|---|---|---|---|---|---|
| **IIT-NRC data base** | | | | | |
| Proposed syst. | Arandjelovic et al. [1] | Gorodnichy [19] | Tangelder et al. [48] | Wang et al. [53] | |
| 100% | 100% | 95% | 95% | 93% | |
| **MoBo data base** | | | | | |
| Proposed syst. | Cevikalp et al. [11] | Hadid et al. [23] | Liu et al. [31] | Wang et al. [55] | Zhou et al. [57] |
| 100% | 98% | 94% | 99% | 94% | 100% |

in time. Accuracy obtained with few ROIs then decreases, while the number of ROIs necessary to achieve a video-based classification rate comparable to 100% increases. On the other hand, the video-based classification rate obtained after updating classes through incremental learning grows over time, as new blocks of data become available. When blocks are available, the AMCS needs fewer ROIs to achieve a higher video-based accuracy and it is eventually comparable to 100% with the same number of ROIs as during enrollment.

The effect on AMCS accuracy of video sequence length used to recognize individuals is also shown in Fig. 11. With each passing ROI, evidence in the form of class predictions is accumulated. As FAM networks outputs are binary vector, the number of ROIs that predicts a class is instead accumulated and used to establish a ranking through majority voting. The cumulative match curves in Fig. 11 show that as the length of the video sequences (and number of ROIs) increases, ambiguity regarding the predictions diminishes. The probability of the correct class being the first ranked prediction increases to eventually reach 100%, while the minimal ranking with a cumulative probability of 100% also decreases to eventually reach 1.

When both learning and test sequences of the IIT-NRC data base were recorded, the individuals were all initially facing the camera, giving a full frontal image of their face. The ROIs of the first frames are similar leading to classification rates obtained with the first pattern of each video sequences that are always higher than those obtained with a single ROI. As the individuals begin moving, changing his facial orientation and expression, different facial views, corresponding to data points in new regions of the feature space, are presented to the system. Since the first frames of each video sequence are initially present in $D_1$, the biometric face models are not well defined and these new regions in the feature space are then unexplored by the FAM networks. Recognizing an individual toward the end of a video sequence is thus more difficult. As the number of frames used to perform recognition increases, correct predictions for each ROIs accumulated at the beginning of the test sequences are surpassed by the wrong predictions accumulated with the subsequent ROIs. Until all classes are updated, this leads to a video-based classification rate that tends to decrease at the end of each sequence.

In the worst case (the update scenario), Table 5 shows that the AMCS with the DPSO-based strategy needs 5 additional ROIs than with SWARM to have an accuracy comparable to 100%. Assuming ideal tracking performances and a camera that acquires video sequences at a rate of 30 frames per second, this represents around a fifth of a second. This level of performance is also achieved with only a third of the resources (see Tables 3, 4). The number of additional ROIs needed to achieve a classification rate comparable to 100% grows to six with ensembles obtained through batch learning of all cumulative data. Results are similar with the MoBo data base, except for AMCSs with the proposed DPSO-based strategy which require fewer ROIs to achieve a 100% classification rate. The more controlled data acquisition conditions for MoBo also make it possible for a single FAM network to achieve a perfect video-based classification rate.

Compared to other methods proposed in literature for video-based face recognition, an AMCS with the proposed DPSO learning strategy outperforms other systems, except that of Arandjelovic et al. [1] with the IIT-NRC data base and Zhou et al. [57] with the MoBo data base. Regardless of the scenario, the AMCS with LBESTS$_{+d}$ must accumulate about 1 s of video stream to accumulate the ensemble responses and achieve a classification rates of 100% after incremental learning of the entire MoBo data base. In comparison, after performing *batch learning* of the MoBo data base Zhou et al. [57] achieved the same result by accumulating classifier responses for 0.5 s. While Arandjelovic et al. [1] also obtained a 100% video-based classification rate, the number of accumulated response to achieve this is not available.

### 5.3. Particle diversity versus classifier diversity

As mentioned, the DPSO-based incremental learning strategy is based on the hypothesis that particle diversity in the hyperparameter space implicitly generates diversity in the feature space, among classifiers associated with those particles. Based on the experiment introduced in Fig. 7, this hypothesis is verified. Fig. 12 presents the value of three classifiers correlation/diversity indicators – $Q$ statistic (Eq. (6)), correlation coefficient (Eq. (7)), and the specialized ambiguity indicator for FAM networks (Eq. (9)) – as a function of particle diversity in the hyperparameter space (Eq. (5)) when training on the IIT-NRC data base. Figs. 13 and 14 also show the classifier and particle diversity obtained during incremental learning for AMCS where ensembles are formed with LBESTS$_{+d}$ and SWARM.

FAM performs sequential learning of training patterns. Therefore, decision boundaries created during training depends heavily on patterns presentation order. Given that this order is typically determined randomly, prior each training epoch, an ensemble's classifiers will differ, even though they were trained with the same hyperparameters (see Fig. 12). When all particles are initially positioned at the global best position, this yields correlation indicators that are lower than one ($\overline{Q_{e_1e_2}} = 0.80 \pm 0.01$ and $\overline{\rho_{e_1e_2}} = 0.47 \pm 0.01$) and a diversity indicator higher than 0 ($\overline{\Delta\theta_{e_1e_2}} = 0.07 \pm 0.01$).

As the hypercube expands (see Fig. 7), particle swarm diversity increases linearly. No matter if diversity is computed in the decision space with the correlation indicators based on ensemble disagreement ($\overline{Q_{e_1e_2}}$ and $\overline{\rho_{e_1e_2}}$), or with ambiguity in the feature space ($\overline{\Delta\theta_{e_1e_2}}$), classifier diversity (correlation) follows the same trend by increasing (decreasing) constantly. Depending on the indicator, diversity in the classification environment changes significantly for

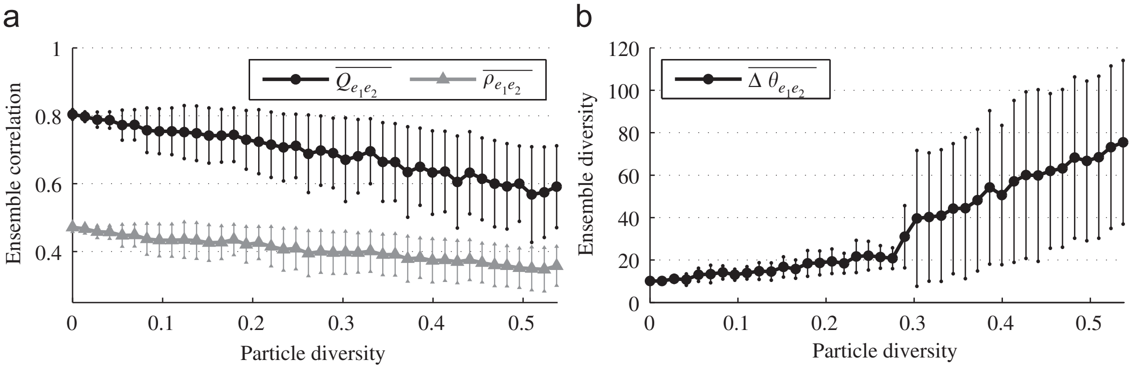

**Fig. 12.** Ensemble diversity in the classification environment as a function of particle diversity ($\overline{\delta_{e_1e_2}}$) in the optimization environment. Ensemble diversity is shown using two correlation indicators ($\overline{Q_{e_1e_2}}$ and $\overline{\rho_{e_1e_2}}$ in (a)), and a diversity indicator ($\overline{\Delta\theta_{e_1e_2}}$ in (b)). A decrease in correlation signifies an increase in diversity. Each indicator is shown with its 90% confidence interval.

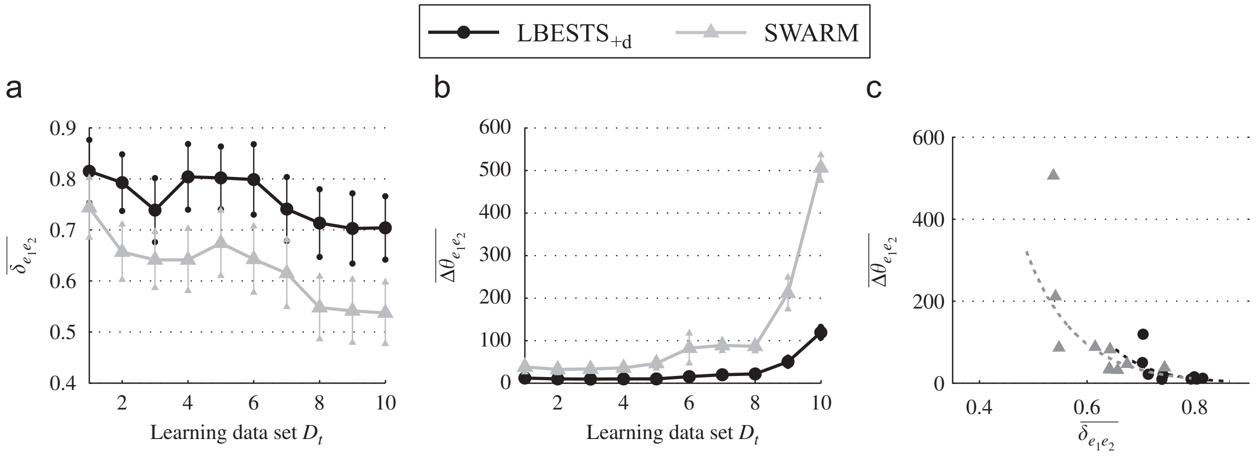

**Fig. 13.** Particle and classifier diversity of the AMCS's ensembles versus the number of learning blocks during the enrollment learning scenario (a, b). The FAM ambiguity indicator (Eq. (9)) was used for classifier diversity and all results are presented with their 90% confidence interval. Also shown is classifier diversity as a function of the particle diversity using all data points (c). (a) Particle diversity, (b) classifier diversity, (c) scatter plot.

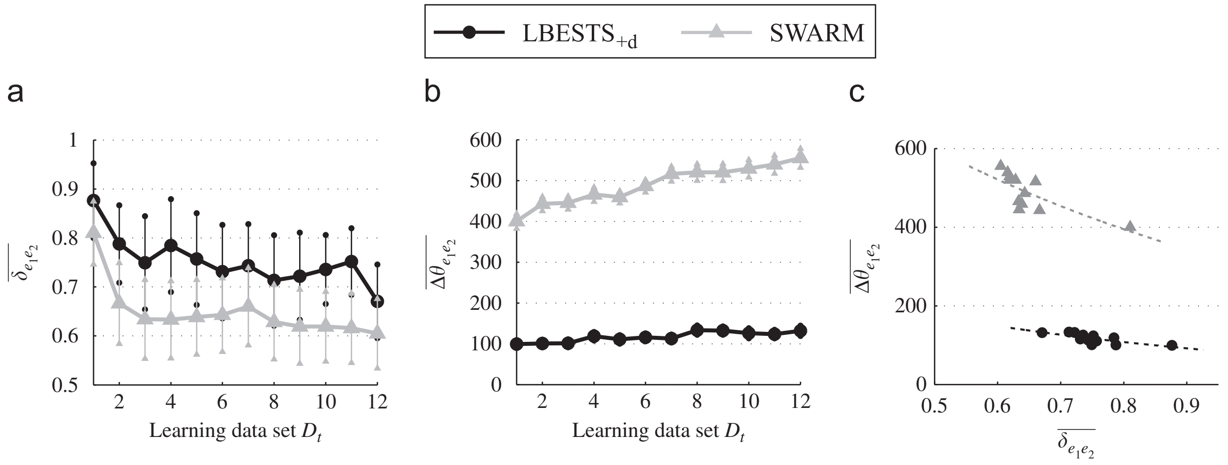

**Fig. 14.** Particle and classifier diversity of the AMCS's ensemble versus the number of learning block during the update learning scenario (a, b). The ambiguity indicator (Eq. (9)) was used for classifier diversity and all results are presented with their 90% confidence interval. Also shown is the classifier diversity as a function of the particle diversity using all data points (c). (a) Particle diversity, (b) classifier diversity, (c) scatter plot.

different levels of particle diversity: the $Q$ statistic differs for $\overline{\delta_{e_1e_2}} = 0.26$ ($\overline{Q_{e_1e_2}} = 0.7 \pm 0.1$), correlation differs for $\overline{\delta_{e_1e_2}} = 0.25$ ($\overline{\rho_{e_1e_2}} = 0.41 \pm 0.06$), and ambiguity-based diversity differs for $\overline{\delta_{e_1e_2}} = 0.09$ ($\overline{\Delta\tau_{e_1e_2}} = 0.10 \pm 0.02$). Overall, results confirm the initial hypothesis that diversity in the hyperparameter space does indeed translates to diversity among classifiers in the feature space.

It is important to note that FAM networks are very sensitive to the match tracking (MT) hyperparameter ($\epsilon$) when it is close to zero. Indeed, positive and negative values of $\epsilon$ have opposite effect on the dept of search performed among $F_2$ nodes when training on a pattern **a**. When a winning $F_2$ node $j^*$ leads to a prediction error, positive (negative) values of $\epsilon$ restricts (relaxes) the condition on which subsequent $F_2$ nodes passes the vigilance test.

With $\epsilon > 0$ ($\epsilon < 0$), FAM tends to create more (fewer), but smaller (larger), category hyper-rectangles, leading to narrow (broad) generalization. This explains the large confidence interval observed when $\overline{\delta_{e_1 e_2}} > 0.3$. For two out of 10 replications, the global best position found during DPSO optimization leads to global optimal values with $\epsilon \in [-0.03, 0.06]$. When particle diversity reaches $\delta = 0.3$, ensembles are then formed of classifiers with both positive and negative match tracking values, leading to a considerable classifier diversity ($\overline{\Delta\theta_{e_1 e_2}} > 200$).

However, as results shown in Figs. 13 and 14, the relation between particle and classifier diversity during incremental learning is not as simple as with batch learning. When data is learned incrementally over time, FAM decisions boundaries may be adjusted to accommodate new classes. In the hyperparameter space, the objective function changes over time and regions with potential optima become increasingly localized [13]. Diversity in the hyperparameter space then decreases gradually, and convergence of subswarms toward local optima reduces particle swarm diversity below that obtained with the greedy search process. Under the update scenario, all classes are represented at the beginning of the learning process, and FAM networks are more complex, which increases the impact of the hyperparameters used during training. Even if particle diversity values for $D_1$ are lower than those obtained during enrollment, classifier diversity is typically about ten times higher. This increased complexity leads to an increased variation across the different replications, resulting in larger confidence intervals for particle diversity. Even if results are comparable for AMCSs with LBESTS$_{+\mathrm{d}}$ and SWARM (Fig. 14a), LBESTS$_{+\mathrm{d}}$ still tends to provide the highest particle diversity.

Meanwhile, in the feature space, FAM networks are trained using different hyperparameters and different pattern presentation orders with each passing $D_t$. As shown in Fig. 13, as new data is learned by the AMCS, the ensemble of classifiers becomes increasingly diverse. Figs. 13c and 14c illustrate that, in a context of incremental learning, there is an inverse relationship between particle and classifier diversity. As mentioned, the higher ensemble diversity observed with SWARM does not translate to a significantly higher classification rate for the ensemble.

Note that this does not contradict results presented in Fig. 12, where a diversity analysis is performed with FAM networks that are initially all in the same state prior learning the whole IIT-NRC data base with batch learning. Instead, for each learning data set $D_t$, Figs. 13 and 14 present only one point in the particle—classifier diversity space of what is a diversity analysis when using the greedy search process (Algorithm 2) on a local time frame and with networks that are in different initial conditions.

## 6. Conclusion

In this paper, an incremental learning strategy based on DPSO is proposed to evolve heterogeneous ensembles of classifiers in response to new data. This strategy is applied to an AMCS for video-based face recognition consisting of a pool of FAM neural networks to classify face regions, the DNPSO algorithm to optimize classifier parameters such that classification rate is maximized. The dynamic swarm properties are then exploited to perform an ensemble selection process based on accuracy and diversity.

Overall results confirm that there is indeed a correlation between diversity in the optimization environment and diversity in the classification environment. The diversity of solutions can easily be controlled in the optimization environment with a DPSO algorithm, and allows for an efficient selection of diversified ensembles of classifiers. When the AMCS uses the DPSO learning strategy, the best results are thus obtained by combining the neural networks associated to the local best particles with a greedy search that aims to maximize particle diversity in the hyperparameter space. Although this approach does not ensure finding an ensemble with the global optimum particle diversity, this search algorithm allows to select ensembles that yield classification rates comparable to that of reference ensemble-based and batch learning techniques, but with only a fraction of the resources and without the need to assess diversity among classifiers in the feature or decision space.

While the classifiers evolved using the DPSO-based learning strategy are defined according to the personal best position of each particle, the mechanisms for DPSO subswarms creation proposed in the literature adjust instead the current position of the swarm. Moreover, as with most DPSO algorithms, DNPSO focuses on detecting and tracking the objective function's local optima. Even if this allows optimization in a dynamic environment, evolving a swarm of neural networks also requires maintaining diversity around each local optima to maximize the effect of the greedy search. Future work should involve devising a DPSO algorithm that circumvent those issues and is specifically aimed at the creation of diversified pool of classifiers within the AMCS framework. A majority vote is employed for classifier fusion, although further studies should be made to determine an optimal fusion function according to ensembles selected through the DPSO-based strategy. Moreover, one could define criteria to specify the face samples from each learning block $D_t$ that are selected for the long term memory (LTM) to provide the highest performance. Validation and fitness estimation (respectively done during the learning and optimization processes) would therefore be performed using the most representative data of each class distribution.

## Acknowledgments

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

## References

[1] O. Arandjelovic, R. Cipolla, A methodology for rapid illumination-invariant face recognition using image processing filters, Computer Vision and Image Understanding 113 (2009) 159–171.

[2] B. Bakker, T. Heskes, Clustering ensembles of neural network models, Neural Networks 16 (2003) 261–269.

[3] T. Blackwell, J. Branke, Multi-swarm optimization in dynamic environments, in: Applications of Evolutionary Computing, Coimbra, Portugal, 2004, pp. 489–500.

[4] J. Branke, Memory enhanced evolutionary algorithms for changing optimization problems, in: IEEE Congress on Evolutionary Computation, Washington, DC, USA, 1999, pp. 1875–1882.

[5] G. Brown, J. Wyatt, R. Harris, X. Yao, Diversity creation methods: a survey and categorization, Information Fusion 29 (2005) 5–20.

[6] A.M.P. Canuto, M.C.C. Abreu, L.d.M. Oliveira, J.C. Xavier Jr., A.d.M. Santos, Investigating 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 26 (2007) 472–486.

[7] A. Carlisle, G. Dozier, Tracking changing extrema with adaptive particle swarm optimizer, in: World Automation Congress, Orlando, Florida USA, 2002, pp. 265–270.

[8] G.A. Carpenter, S. Grossberg, N. Markuzon, J.H. Reynolds, D.B. Rosen, Fuzzy artmap: a neural network architecture for incremental supervised learning of analog multidimensional maps, IEEE Transactions on Neural Networks 3 (1992) 698–713.

[9] G.A. Carpenter, S. Grossberg, J.H. Reynolds, Artmap: supervised real-time learning and classification of nonstationary data by a self-organizing neural network, Neural Networks 4 (1991) 565–588.

[10] G.A. Carpenter, N. Markuzon, Artmap-IC and medical diagnosis: instance counting and inconsistent cases, Neural Networks 11 (1998) 323–336.

[11] H. Cevikalp, B. Triggs, Face recognition based on image sets, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 2567–2573.

[12] J.F. Connolly, E. Granger, R. Sabourin, Incremental adaptation of fuzzy artmap neural networks for video-based face classification, in: IEEE Symposium on Computational Intelligence for Security and Defence Applications (CISDA), Ottawa, Canada, 2009, pp. 1–8.

[13] J.F. Connolly, E. Granger, R. Sabourin, An adaptive classification system for video-based face recognition, Information Sciences, doi:10.1016/j.ins.2010.02.026, in press.

[14] J.F. Connolly, E. Granger, R. Sabourin, An adaptive ensemble of fuzzy artmap neural networks for video-based face classification, in: IEEE Congress on Evolutionary Computation, 2010, pp. 1–8.

[15] W. Du, B. Li, Multi-strategy ensemble particle swarm optimization for dynamic optimization, Information Science 178 (2008) 3096–3109.

[16] A.P. Engelbrecht, Fundamental of Computational Swarm Intelligence, John Wiley & Sons, 2005.

[17] M.J. Er, S. Wu, J. Lu, H.L. Toh, Face recognition with radial basis function neural networks, IEEE Transactions on Neural Networks 13 (2002) 697–710.

[18] N. Garcá-Pedrajas, C. Hervás-Martínez, D. Ortiz-Boyer, Cooperative coevolution of artificial neural network ensemble for pattern classification, IEEE Transactions on Evolutionary Computation 9 (2005) 271–302.

[19] D.O. Gorodnichy, Video-based framework for face recognition in video, in: Second Workshop on Face Processing in Video in Proceedings of the Conference on Computer and Robot Vision, Victoria, Canada, 2005, pp. 325–344.

[20] E. Granger, P. Henniges, L.S. Oliveira, R. Sabourin, Supervised learning of fuzzy artmap neural networks through particle swarm optimization, Journal of Pattern Recognition Research 2 (2007) 27–60.

[21] E. Granger, M.A. Rubin, S. Grossberg, P. Lavoie, A what-and-where fusion neural network for recognition and tracking of multiple radar emitters, Neural Networks 14 (2001) 325–344.

[22] R. Gross, J. Shi, The CMU Motion of Body (MoBo) Database, 2001, Technical Report CMU-RI-TR-01-18, Carnegie Mellon University, 2001.

[23] A. Hadid, M. Pietikinen, Selecting models from videos for appearance-based face recognition, in: In Proceedings of the 17th International Conference on Pattern Recognition, 2004, pp. 304–308.

[24] S.T. Hadjitodorov, L.I. Kuncheva, L.P. Todorova, Moderate diversity for better cluster ensembles, Information Fusion 7 (2006) 264–275.

[25] X. Hu, R.C. Eberhart, Adaptive particle swarm optimization: detection and response to dynamic systems, in: IEEE Congress on Evolutionary Computation, vol. 2, Honolulu, USA, 2002, pp. 1666–1670.

[26] X. Jiang, W. Ser, Online fingerprint template improvement, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (2002) 1121–1126.

[27] M.N. Kapp, R. Sabourin, P. Maupin, An empirical study on diversity measures and margin theory for ensembles of classifiers, in: International Conference on Information Fusion, Québec, Canada, 2007, pp. 1–8.

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

[29] L.I. Kuncheva, Classifier ensembles for changing environments, in: Proceedings of the 5th International Workshop on Multiple Classifier Systems (MSC), Cagliari, Italy, 2004, pp. 1–15.

[30] X. Li, J. Branke, T. Blackwell, Particle swarm with speciation and adaptation in a dynamic environment, in: Genetic And Evolutionary Computation Conference, Seattle, USA, 2006, pp. 51–58.

[31] X. Liu, T. Chen, Video-based face recognition using adaptive hidden Markov models, in: IEEE Conference on Computer Vision and Pattern Recognition, Los Alamitos, CA, USA, 2003, pp. 340–345.

[32] Y. Liu, X. Yao, Q. Zhao, T. Higuchi, Evolving a cooperative population of neural networks by minimizing mutual information, in: IEEE Congress on Evolutionary Computation, Seoul, Korea, 2001, pp. 384–389.

[33] J. Lu, K.N. Plataniotis, A.N. Venetsanopoulos, S.Z. Li, Ensemble-based discriminant learning with boosting for face recognition, IEEE Transactions on Neural Networks 17 (2006) 166–178.

[34] F. Matta, J.L. Dugelay, Person recognition using facial video information: a state of the art, Journal of Visual Language and Computing 20 (2009) 180–187.

[35] H. Moon, P.J. Phillips, Computational and performance aspect of pca-based face-recognition algorithms, Perception 30 (2001) 303–321.

[36] A. Nickabadi, M.M. Ebadzadeh, R. Safabakhsh, Evaluating the performance of DNPSO in dynamic environments, in: IEEE International Conference on Systems, Man, and Cybernetics, Singapore, 2008, pp. 12–15.

[37] D.F. Olivieira, A.M.P. Canuto, M.C.P. de Souto, Use of multi-objective genetic algorithms to investigate the diversity/accuracy dilemma in heterogeneous ensembles, in: IEEE International Joint Conference on Neural Networks, Atlanta, USA, 2009, pp. 1238–1245.

[38] O. Olorunda, A.P. Engelbrecht, Measuring exploration/exploitation in particle swarms using swarm diversity, in: IEEE Congress on Evolutionary Computation, Hong Kong, China, 2008, pp. 1128–1134.

[39] E. Özcan, M. Yýlmaz, Particle swarms for multimodal optimization, in: Adaptive and Natural Computing Algorithms, Warsaw, Poland, 2007, pp. 366–375.

[40] N. Poh, W. Rita, J. Kittler, F. Roli, Challenges and research directions for adaptive biometric recognition systems, in: M. Tistarelli, M. Nixon (Eds.), Advances in Biometrics, Lecture Notes in Computer Science, vol. 5558, Springer, Berlin, Heidelberg, 2009, pp. 753–764.

[41] M.A. Potter, K.A.D. Jong, Cooperative coevolution: an architecture for evolving coadapted subcomponents, Evolutionary Computation 8 (2000) 1–29.

[42] T. Rashid, A heterogeneous ensemble network using machine learning techniques, International Journal of Computer Science an Network Security 9 (2009) 335–339.

[43] A. Rattani, Adaptive biometric system based on template update procedures, Ph.D. Thesis, University of Cagliari, Cagliari, Italy, 2010.

[44] F. Roli, L. Didaci, G. Marcialis, Adaptive biometric systems that can improve with use, in: N.K. Ratha, V. Govindaraju (Eds.), Advances in Biometrics, Springer, London, 2008, pp. 447–471.

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

[46] Y. Su, S. Shan, X. Chen, W. Gao, Hierarchical ensemble of global and local classifiers for face recognition, in: IEEE International Conference on Computer Vision, Rio de Janeiro, Brazil, 2007, pp. 1–8.

[47] E.K. Tang, P.N. Suganthan, X. Yao, An analysis of diversity measure, Machine Learning 65 (2006) 247–271.

[48] J.W. Tangelder, B.A. Schouten, Learning a sparse representation from multiple still images for on-line face recognition in an unconstrained environment, in: 18th International Conference on Pattern Recognition, vol. 3, Los Alamitos, CA, USA, 2006, pp. 1087–1090.

[49] A. Tsymbla, M. Pechenizkiy, P. Cunningham, S. Puuronen, Dynamic integration of classifiers for handling concept drift, Information Fusion 9 (2008) 56–68.

[50] A. Ulaş, M. Semerci, O.T. Yildiz, E. Alpaydin, Incremental construction of classifier and discriminant ensembles, Information Sciences 179 (2009) 1298–1318.

[51] U. Uludag, A. Ross, A. Jain, Biometric template selection and update: a case study in fingerprints, Pattern Recognition 37 (2004) 1533–1542.

[52] G. Valentini, Ensemble methods based on bias-variance analysis, Ph.D. Thesis, University of Genova, Genova, Switzerland, 2003.

[53] C. Wang, Y. Li, X. Ao, Quality fusion rule for face recognition in video. in: Advanced Concepts for Intelligent Vision Systems of Lecture Notes in Computer Science, vol. 5807, Springer, Berlin, Heidelberg, 2009, pp. 333–342.

[54] H. Wang, D. Wang, S. Yang, Triggered memory-based swarm optimization in dynamic environments, in: Applications of Evolutionary Computing, Valencia, Spain, 2007, pp. 637–646.

[55] R. Wang, S. Shan, X. Chen, W. Gao, Manifold-manifold distance with application to face recognition based on image set, in: IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, USA, 2008, pp. 1–8.

[56] W.Y. Zhao, R. Chellappa, P.J. Phillips, A.P. Rosenfeld, Face recognition: a literature survey, ACM Computing Surveys 35 (2003) 399–458.

[57] S. Zhou, V. Krueger, R. Chellappa, Probabilistic recognition of human faces from video, Computer Vision and Image Understanding 91 (2003) 214–245.

[58] Z.H. Zhou, J. Wu, W. Tang, Ensembling neural networks: many could be better than all, Artificial Intelligence 137 (2002) 239–263.

**Jean-François Connolly** received his B.Ing. degree in Electrical Engineering in 2002, and M.Sc.A. degree in Applied Mathematics in 2005 from the École Polytechnique de Montréal. During his Master's degree, his work was focused on applying a multiscale scheme, inspired by wavelet multiresolution analysis, to approximate the discriminant function of the Quantron artificial neuron. He joined the École de Technologie Supérieure, Université du Québec, Montréal, P.Q. Canada in 2007, where he is currently completing his Ph.D. thesis in the Laboratoire d'imagerie, de vision et d'intelligence artificielle (LIVIA). His current research interests are machine learning, adaptive classification systems, incremental learning, evolutionary algorithms, multiclassifier systems, with applications in biometric face authentication.

**Eric Granger** obtained a Ph.D. in Electrical Engineering from the École Polytechnique de Montréal in 2001, and from 1999 to 2001, he was a Defence Scientist at Defence R&D Canada in Ottawa. Until then, his work was focused primarily on neural network signal processing for fast classification of radar signals in Electronic Surveillance systems. From 2001 to 2003, he worked in R&D with Mitel Networks Inc. on algorithms and ASICs to implement cryptographic functions in IP-based communication platforms. In 2004, Dr. Eric Granger joined the École de technologie supérieure (University of Quebec), where he presently holds the rank of Assistant Professor in the département de génie de la production automatisée. Since joining ÉTS, he has been a member of the Laboratoire d'imagerie, de vision et d'intelligence artificielle (LIVIA), and his main research interests are adaptive classification systems, incremental learning, ambiguity and novelty detection, neural and statistical classifiers, and multi-classifier systems, with applications in biometrics (authentication from signatures and faces), military surveillance (recognition of radar signals), and computer and network security (intrusion detection). To date, he has authored/co-authored over 50 open literature papers with peer reviews, and supervised/co-supervised 20 graduate students in these areas of research. In 2011, he has been named member of the Task Force on Biometric of IEEE CIS ISATC.

**Dr. Robert Sabourin** joined in 1977 the physics department of the Montreal University where he was responsible for the design, experimentation and development of scientific instrumentation for the Mont Mégantic Astronomical Observatory. His main contribution was the design and the implementation of a microprocessor-based fine tracking system combined with a low-light level CCD detector. In 1983, he joined the staff of the École de Technologie Supérieure, Université du Québec, in Montréal where he co-founded the Dept. of Automated Manufacturing Engineering where he is currently Full Professor and teaches Pattern Recognition, Evolutionary Algorithms, Neural Networks and Fuzzy Systems. In 1992, he also joined the Computer Science Department of the Pontifícia Universidade Católica do Paraná (Curitiba, Brazil) where he was co-responsible for the implementation in 1995 of a master program and in 1998 a Ph.D. program in applied computer science. Since 1996, he is a senior member of the Centre for Pattern Recognition and Machine Intelligence (CENPARMI, Concordia University).

Dr. Sabourin is the author (and co-author) of more than 300 scientific publications including journals and conference proceeding. He was co-chair of the program committee of CIFED'98 (Conférence Internationale Francophone sur l'Écrit et le Document, Québec, Canada) and IWFHR'04 (9th International Workshop on Frontiers in Handwriting Recognition, Tokyo, Japan). He was nominated as Conference co-chair of ICDAR'07 (9th International Conference on Document Analysis and Recognition) that has been held in Curitiba, Brazil in 2007.

His research interests are in the areas of adaptive biometric systems, intelligent watermarking systems, evolutionary computation and bio-cryptography.