Adaptive ensembles for face recognition in changing video surveillance environments

Article *in* Information Sciences · December 2014

DOI: 10.1016/j.ins.2014.07.005

Eric Granger
École de Technologie Supérieure (Université …
190 PUBLICATIONS **1,450** CITATIONS

Robert Sabourin
École de Technologie Supérieure
416 PUBLICATIONS **6,467** CITATIONS

Gian Luca Marcialis
Università degli studi di Cagliari
119 PUBLICATIONS **1,953** CITATIONS

Information Sciences 286 (2014) 75–101

Contents lists available at ScienceDirect

Information Sciences

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

Adaptive ensembles for face recognition in changing video surveillance environments

C. Pagano [a,*], E. Granger [a], R. Sabourin [a], G.L. Marcialis [b], F. Roli [b]

[a] *Lab. d'imagerie, de vision et d'intelligence artificielle, École de technologie supérieure, Université du Québec, Montreal, Canada*
[b] *Pattern Recognition and Applications Group, Dept. of Electrical and Electronic Engineering, University of Cagliari, Cagliari, Italy*

ARTICLE INFO

Article history:
Received 11 October 2013
Received in revised form 26 May 2014
Accepted 6 July 2014
Available online 18 July 2014

Keywords:
Multi-classifier system
Incremental learning
Adaptive biometric
Change detection
Face recognition
Video surveillance

ABSTRACT

Recognizing faces corresponding to target individuals remains a challenging problem in video surveillance. Face recognition (FR) systems are exposed to videos captured under various operating conditions, and, since data distributions change over time, face captures diverge w.r.t. stored facial models. Although these models may be adapted when new reference videos become available, incremental learning with faces captured under different conditions may lead to knowledge corruption. This paper presents an adaptive multi-classifier system (AMCS) for video-to-video FR in changing surveillance environments. During enrolment, faces captured in reference videos are employed to design an individual-specific classifier. During operations, a tracker allows to regroup facial captures for individuals in the scene, and accumulate the predictions per track for robust spatiotemporal FR. Given a new reference video, the corresponding facial model is adapted according to the type of concept change. If a gradual pattern of change is detected, the individual-specific classifier(s) are adapted through incremental learning. To preserve knowledge, another classifier is learned and combined with the individuals previously-trained classifier(s) if an abrupt change is detected. For proof-of-concept, the performance of a particular implementation of this AMCS is assessed using videos from the Faces in Action dataset. By adapting facial models according to changes detected in new reference videos, this AMCS allows to sustain a high level of accuracy comparable to the same system that is always updated using a learn-and-combine approach, while reducing time and memory complexity. It also provides higher accuracy than incremental learning classifiers that suffer the effects of knowledge corruption.

1. Introduction

The global market for video surveillance (VS) technologies has reached revenues in the billions of $US as traditional analogue technologies are replaced by IP-based digital ones. VS networks are comprised of a growing number of cameras, and transmit or archive massive quantities of data for reliable decision support. The ability to automatically recognize and track individuals of interest across these networks, and under a wide variety of operating conditions, may provide enhanced screening and situation analysis.

* Corresponding author.
E-mail addresses: cpagano@livia.etsmtl.ca (C. Pagano), eric.granger@etsmtl.ca (E. Granger), robert.sabourin@livia.etsmtl.ca (R. Sabourin), marcialis@diee.unica.it (G.L. Marcialis), roli@diee.unica.it (F. Roli).

http://dx.doi.org/10.1016/j.ins.2014.07.005
0020-0255/© 2014 Elsevier Inc. All rights reserved.

In decision support systems for VS, face recognition (FR) has become an important function in two types of applications. In *watch-list screening applications*, facial models[1] used for classification are designed using regions of interest (ROIs) extracted from the reference still images or mugshots of a watch-list. Then, still-to-video FR seeks to determine if faces captured in video feeds correspond to an individual of interest. In *person re-identification* for search and retrieval applications, facial models are designed using ROIs extracted from reference videos and tagged by a human operator. Then, video-to-video FR seeks to alert the operator when these individuals appear in either live (real-time monitoring) or archived (post-event analysis) videos.

This paper focuses on the design of robust classification systems for video-to-video FR in changing surveillance environments, as required in person re-identification. In this context, public security organizations have deployed many CCTV and IP surveillance cameras in recent years, but FR performance is limited by human recognition abilities. Indeed, accurate and timely recognition of ROIs is challenging under semi-controlled (e.g., in an inspection lane, portal or checkpoint entry) and uncontrolled (e.g., in cluttered free-flow scene at an airport or casino) capture conditions. Given the limited control during capture, the performance of state-of-the-art systems are affected by the variations of pose, scale, orientation, expression, illumination, blur, occlusion and ageing. Moreover, FRiVS is an open set problem, where only a small proportion of the faces captured during operations correspond to individuals of interest. Finally, ROIs captured in videos are matched against facial models designed a priori, using a limited number of high quality reference samples captured during enrolment. Accuracy of face classification is highly dependent on the representativeness of models, and thus number, relevance and diversity of these samples.

Some specialized classification architectures have been proposed for FRiVS. For instance, the open-set Transduction Confidence Machine-*k*NN (TCM-*k*NN) is comprised of a global multi-class classifier with a rejection option tailored for unknown individuals [37]. Classification systems for FRiVS should however be modeled as independent individual-specific detection problems, each one implemented using one- or two-class classifiers (i.e., detectors), with specialized thresholds applied to their output scores [48]. The advantages of class-modular architectures in FRiVS (and biometrics in general) include the ease with which face models (or classes) may be added, updated and removed from the systems, and the possibility of specializing feature subsets and decision thresholds to each specific individual. Individual-specific detectors have been shown to outperform global classifiers in applications where the reference design data is limited w.r.t. the complexity of underlying class distributions and to the number of features and classes [45,54]. Moreover, some authors have argued that biometric recognition is in essence a multi-classifier problem, and that biometric systems should co-jointly solve several classification tasks in order to achieve state-of-the-art performance [5].

Modular architectures for FRiVS have been proposed by Ekenel et al. [19], where 2-class individual-specific Support Vector Machines are trained on a mixture of target and non-target samples. Given the limited amount of reference samples and the complexity of environments, modular approaches have been extended by assigning a classifier ensemble to each individual. For example, Pagano et al. [48] proposed a system comprised of an ensemble of 2-class ARTMAP classifiers per individual, each one designed using target and non-target samples. A pool of diversified classifiers is generated using an incremental learning strategy based on dynamic PSO, and combined in the ROC space using a Boolean fusion function.

In person re-identification, new reference video become available during operations or through some re-enrolment process, and an operator can extract a set of facial ROIs belonging to a target individual. In order to adapt an individual's facial model in response to these new ROI samples, the parameters of a individual-specific classifier can be re-estimated through supervised incremental learning. For example, ARTMAP neural networks [9] and extended Support Vector Machines [52] have been designed or modified to perform incremental learning. However, these classifiers are typically designed under the assumption that data is sampled from a static environment, where class distributions remain unchanged over time [25].

Under semi- and uncontrolled capture conditions, ROI samples that are extracted from new reference videos may incorporate various patterns of change that reflect varying concepts.[2] While gradual patterns of change in operational conditions are often observed (due to, e.g., ageing over sessions), abrupt and recurring patterns (caused by, e.g., new pose angle versus camera) also occur in FRiVS. A key issue in changing VS environments is adapting facial models to assimilate samples from new concepts without corrupting previously-learned knowledge, which raises the *plasticity–stability* dilemma [26]. Although updating a single classifier may translate to low system complexity, incremental learning of ROI samples extracted from videos that reflect significantly different concepts can corrupt the previously acquired knowledge [13,50]. Incomplete design data and changing distributions contribute to a growing divergence between the facial model and the underlying class distribution of an individual.

Adaptive ensemble methods allow to exploit multiple and diverse views of an environment, and have been successfully applied in cases where concepts change in time. By assigning an adaptive ensemble to each individual, it is possible to adapt a facial model by updating the pool of classifiers and/or the fusion function [33]. For example, with iques like Learn++ [50] and other Boosting variants [47], a classifier is trained independently using new samples, and weighted such that accuracy is maximized. Other approaches discard classifiers when they become inaccurate or concept change is detected, while maintaining a pool with these classifiers allows to handle recurrent change. Classifier ensembles are well suited for adaptation in changing environments since they can manage the *plasticity–stability* dilemma at the classifier level – when samples are

[1] A *facial model* is defined as either a set of one or more reference captures (used in template matching systems), or a statistical model estimated through training with reference captures (used in neural or statistical classification systems) corresponding to a target individual.

[2] A *concept* can be defined as the underlying class distribution of data captured under specific condition, in our context due to different pose angle, illumination, scale, etc. [43].

significantly different, previously acquired knowledge can be preserved by initiating and training a new classifier on the new data. However, since the number of classifiers grows, benefits (accuracy and robustness) are achieved at the expense of system complexity.

In this paper, an adaptive multi-classifier system (AMCS) for video-to-video FR is proposed to maintain a high level of performance in changing surveillance environments. It is initially comprised of a single two-class incremental learning classifier (or detector) per individual, and a change detection mechanism. During enrolment of an individual, ROI samples are extracted from a reference video sequence, and employed to initiate and train the detector. Then, during operations, a face tracker is used to regroup ROIs for different people in the scene according to trajectory.[3] For robust spatio-temporal FR, the prediction of each individual-specific detector is accumulated over along different trajectories. The proposed system allows to update the facial model (detectors) of an individual in response to a new reference video. The change detection mechanisms determines the extent to which ROI samples of a trajectory extracted from new videos correspond to previously-learned concepts. To limit system complexity, if ROI samples incorporate a gradual pattern of change w.r.t. existing concepts, the corresponding pool of classifiers (and, if needed, fusion function) are updated through incremental learning. In contrast, to avoid issues related to knowledge corruption, the AMCS employs a learn-and-combine approach if ROI samples exhibit an abrupt pattern of change w.r.t. existing concepts. Another dedicated classifier is initiated and trained on the new data, and then combined with the individual's previously-trained classifiers.

Some approaches in literatures also exploit change detection to drive adaptation or online-learning of classification systems, such as the Diversity for Dealing with Drifts algorithm [42], the incremental learning strategies based on dynamic PSO [13], and a Just-in-Time architecture that regroups reference templates per concept [3]. However these approaches adapt to changing environments by focusing on the more recent concepts, though weighing or by discarding of previously-learned concepts. In the proposed system, change detection allows to compromise between *stability* (adapting classifiers to known concepts) and *plasticity* (generation of classifiers for new concepts), thereby preserving knowledge (and the ability to recognize) for previously-learned and recurring concepts.

For validation, a particular implementation of the AMCS was considered. During the enrolment of an individual, an histogram representation of the ROI sample distribution is stored, and an incremental learning strategy based on Dynamic Particle Swarm Optimization (DPSO) [13] is employed to generate and evolve a diversified pool of 2-class ARTMAP classifiers [9] using a mixture of target (individual) and non-target (universal and cohort model) samples. Then, when a new reference video (trajectory) becomes available, the change detection process evaluates whether its ROI samples exhibit gradual or abrupt changes w.r.t. to all previously stored histogram distributions using the Hellinger Drift Detection Method [15]. If the new reference samples exhibit a gradual change, the classifier trained with similar data is updated and re-optimized through the DPSO-based learning strategy. If the new reference samples present a significant change, a new histogram distribution is stored, and a new pool of classifiers is generated and optimized. For each pool, the best classifier is then selected to represent its corresponding concept in the AMCS. Each target individual is associated with a single classifier or an ensemble of classifiers, where outputs are combined using a weighted-average score fusion rule. The accuracy and resource requirements of this system is assessed using facial trajectories extracted from video surveillance streams of the Face In Action database [22]. It us comprised of over 200 individuals captured over several months, exhibiting gradual (e.g. expression, ageing) and abrupt (e.g. orientation, illumination) changes.

The rest of this paper is structured as follows. The next section briefly reviews the techniques and challenges of FRiVS. Then, an overview of the literature on change detection and adaptive biometrics is presented in Section 3. In Section 4, the adaptive MCS proposed for video-to-video FR is presented. The experimental methodology (video data, protocol and performance measures) used for validation is presented in Section 5. Finally, simulation results are presented and discussed in Section 6.

2. Background – face recognition in video surveillance

The problem addressed in this paper is the design of accurate and robust systems for video-to-video FR under semi- and uncontrolled capture conditions. Assume that FR is embedded as software executing inside some human-centric decision support system for intelligent video surveillance. In person re-identification applications, an operator may enroll an individual of interest appearing in a video sequence, and gradually design and update their facial models over time by analyzing one or more reference video feeds captured from the particular scene or other sources. Then, individuals of interest must be detected over a network of digital surveillance cameras by matching facial captures against their facial models.

2.1. A generic system for video face recognition

Fig. 1 presents a generic system for video-to-video FR. Each camera captures streams of 2D images or frames, and provides the system with a particular view of individuals populating the scene. This system first performs segmentation to isolate ROIs corresponding to the faces in a frame, from which invariant and discriminant features are extracted and

[3] A *facial trajectory* is defined as a set of ROIs (isolated through face detection) that correspond to a same high quality track of an individual across consecutive frames.

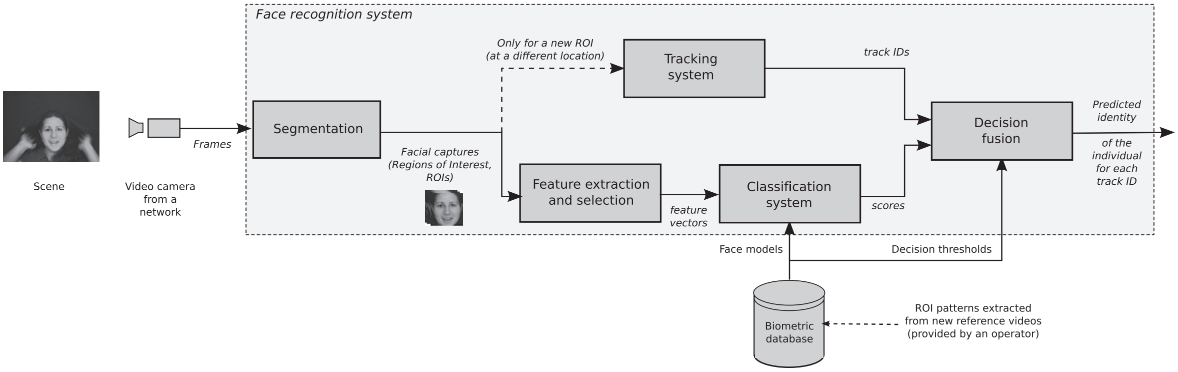


Fig. 1. A human centric system face video-to-video face recognition.

selected for classification (matching) and tracking functions. Some features are assembled into an input pattern, $\mathbf{q} = (q_1, \ldots, q_D)$ for classification, and pattern $\mathbf{b} = (b_1, \ldots, b_e)$ for tracking.

During enrolment, a sot of one or more reference patterns $\mathbf{a}^i[t]$ corresponding to an individual i are extracted from captured ROIs on one or more reference video streams provided by the operator at time t. These are employed to design the user's specific facial model to be stored in a biometric database, as a template or a statistical model. Recognition is typically implemented using a template matcher or using a neural or statistical classifier trained a priori to map the reference patterns to one of the predefined classes, each one corresponding to an individual enrolled to the system. Although each facial model may consist of a set of one or more templates (reference ROI patterns) for template matching, this paper assumes that a model consists of parameters estimated by training a classifier on the reference ROI patterns.

During operations, ROI patterns extracted for unknown individuals in the scene are matched against the model of individuals enrolled to the system. The resulting classification score $s_i(\mathbf{q})$ indicates the likelihood that pattern $\mathbf{q}$ corresponds to the individual i, for $i = 1, \ldots, I$. Each score is compared against the user-specific decision thresholds, θ^i, and the system outputs a list of all possible matching identities. To reduce ambiguities during the decision process, the face tracker may follow the motion and appearance of faces in the scene over successive frames. This allows to regroup ROIs of different individuals and accumulate their matching scores over time.

2.2. State-of-the-art in video surveillance

A common approach to recognizing faces in video consists in only exploiting spatial information, and applying techniques for still-to-still FR (like Eigenfaces or Elastic Bunch Graph Matching) only on high quality ROIs isolated during segmentation [58]. FRiVS remains a difficult task since the faces captured in video frames are typically lower quality and generally smaller than still images. Furthermore, faces captured from individuals in a semi- or unconstrained environment may vary considerably due to limited control over capture conditions (e.g., illumination, pose, expression, resolution and occlusion), and due to changes in an individual's physiology (e.g., aging) [40]. Given these difficulties, high quality faces may never be captured or recognized. More powerful front end processing (face capture and representation) and back-end processing (fusion or responses from cameras, templates, frames) is required for robust performance.

Despite these challenges of video-based FR, it is possible to exploit spatio-temporal information extracted from video sequences to improve performance (see Fig. 1). As mentioned, using face tracking, evidence in individual frames can be integrated over video streams, potentially leading to improved robustness and accuracy. For example, track-and-classify approaches combine information from the motion and appearance faces in a scene to reduce ambiguity (e.g., partial occlusion) [4].

Beyond spatio-temporal approaches, specialized classification architectures have also been proposed for accurate FRiVS. In this case, *open-set* or *open-world* FR operates under the assumption that most faces captured during operations do not correspond to an individual of interest [37]. The probability of detecting the presence of a restrained group of individuals of interest in scenes may be quite low, and facial models may incorporate a significant amount of uncertainty w.r.t. operational environments [49,51].

The Transduction Confidence Machine k-Nearest Neighbour (TCM-k-NN) has been proposed for open-set FR using a multi-class architecture and a specialized rejection option for individuals not enrolled to the system [37]. Kamgar-Parsi et al. propose a face space projection technique where a feed-forward network is designed for each individual [29]. In addition, some multi-verification architectures based on with an individual-specific reject option have been proposed by Ekenel et al. [19] and by Tax and Duin [54], where a specific one- or two-class classifier is assigned to each individual enrolled to the system. These systems allow to add and remove an individual without requiring a complete re-design of the system, and to select individual specific thresholds and feature sets [54]. This ability is particularly favourable in person re-identification, where new individuals are enrolled and monitored on-the-fly by the operator. In addition, separating a multi-class classification problem into more treatable one or two-class problems has been shown to improve the overall performance of the system, adopting the "divide and conquer" approach. For example, in a comparison of classification architectures for FRiVS based on ARTMAP neural networks, class-modular architectures exhibited significant performance improvements [48]. Similarly, in other biometric applications such as character recognition, the performance of a Multi-Layer Perceptron have been significantly improved by the introduction of a class-modular architecture [45,30].

Finally, several other biometric applications, such as speech recognition, operate in *open-set* environments, and exploit a universal background model (UBM) – a non-target population to generate samples from unknown persons from which the target individual can be discriminated – as well as cohort models (CMs) – a non-target population of other people enrolled in the system [8]. The use of such CM is of interest in class-modular architectures such as [54,19]. Sharing information among the different target persons in a class-modular architecture is necessary in order to achieve a high level of performance [5]. Indeed, using some common reference samples (target and non-target samples from a same CM) to design classifiers can be considered as information sharing between classifiers, and may improve the overall system performance.

2.3. Challenges

Systems for FRiVS encounter several challenges in practice. In particular, the facial models are often poor representatives of faces to be recognized during operations [51]. They are typically designed during an a priori enrolment phase, using limited number of reference ROI patterns $\mathbf{a}^i[t]$ from new sets of samples, linked to unknown probability distributions $p(\mathbf{a}[t]|i)$. The underlying data distribution corresponding to individuals enrolled to the system is complex mainly due to: (1) inter- and intra-class variability, (2) variations in capture conditions (interactions between individual and camera), (3) the large number of input features and individuals, and (4) limitations of cameras and signal processing techniques used for segmentation, scaling, filtering, feature extraction and selection, and classification [49]. The performance of FR systems may decline considerably because state-of-the-art neural and statistical classifiers depend heavily on the availability of representative reference data for design and update of face models. In addition, the probability distribution may change gradually or abruptly over time. All these factors contribute to a growing divergence between the facial model of an individual and its underlying class distribution.

Although limited reference data is initially available to design facial models, new reference video sequences may become available over time in a person re-identification application. The systems proposed in the literature for FRiVS usually focus on the matching accuracy, facial quality and the *open-set* context, but not on the update of the face models with ROIs from new and diverse reference videos.

In semi- or uncontrolled environments, faces captured for an individual can correspond to several *concepts* in the input feature space, which can all be relevant for different capture conditions during system operation. Reference video sequences may incorporate samples corresponding to different capture conditions, such as pose angles, illumination, and even ageing. While updating the face models with new videos from known concepts can reinforce the system's knowledge, incremental learning of new reference videos that incorporate different concepts can be a challenge. For example, updating a system with ROI patterns with a specific pose angle can corrupt previously-learned knowledge, learned from samples with other angles. A robust system for FRiVS should detect the presence of various types of changes in the underlying data distribution of individuals. When a new concept emerges, a suitable update strategy should be triggered to preserve pre-existing concepts.

3. Concept change and face recognition

In this paper, a mechanism is considered to detect changes in the underlying data distribution from new reference videos. This mechanism will then trigger different updating strategies. Concept change has been defined by several authors in statistical pattern recognition literature [34]. A *concept* can be defined as the underlying data distribution in $\mathbb{R}^D$ of the problem at some point in time [43]. Given a set of reference ROIs $\{\mathbf{a}^i[t]\}$ captured from target individual i at time t (sampled from the underlying class distribution), a class-conditional distribution of data $p(\mathbf{a}[t]|i)$ may be defined. A *concept change* encompasses various types of noise, trends and substitutions in the underlying data distribution associated with a class or concept. The main assumption is the uncertainty about the future: the data distribution from which the future instance is sampled, $p(\mathbf{a}_{t+1}|i)$ is unknown. To simplify the notation, the time t will be omitted for the remaining of this section, but all the data distribution will be assumed to be time dependent.

A statistical pattern recognition problem can incorporate change due to class priors, $p(i)$, class-conditional distributions $p(\mathbf{a}|i)$ and posterior distributions $p(i|\mathbf{a})$ [34]. A categorization of changes has been proposed by Minku et al. [41], based on severity, speed, predictability and number of re-occurrences, but the following four categories are mainly considered in the literature: noise and abrupt, gradual and recurring changes [35].

Concept changes in pattern recognition may be viewed in the context of FRiVS, where changes can originate from variations in an individual's physiology, as well as in observation conditions (see Table 1). They may range from minor random fluctuations or noise, to sudden abrupt changes of the underlying data distribution, and are not mutually exclusive in real-word environments. From a perspective of any biometric system, changes may originate from phenomena that are either static or dynamic in nature. In addition, those changes can originate from *hidden contexts*, like variations of

Table 1
Types of changes occurring in FRiVS environments.

Type of change	Examples in face recognition
Static environment with:	
– Random noise	– Inherent noise of system (camera, matcher, etc.)
– Hidden concepts	– Different known view points from a camera or of a face (e.g. illumination of images, new face pose or orientation) (Fig. 2(a))
Dynamic environment with:	
– Gradual changes	– Ageing of user (Fig. 2(b))
– Sudden abrupt changes	– New unknown view points on traits; change of camera (Fig. 2(a))
– Recurring contexts	– Unpredictable but recurring changes in capture conditions (e.g. lighting changes due to the weather) (Fig. 2(c))

illumination conditions or pose of the individual which have not been modeled in the system because of the limited representativeness of previously-observed reference samples.

This paper will focus on abrupt, gradual and recurring changes. Fig. 2 illustrates these types of change [35] as they may be observed over time for a concept in a 2 dimensional space (in this example, $\mathbf{a} = (a_1, a_2)$), assuming that it is observed at discrete time steps. It also shows the progression of a corresponding change detection measure.

In this paper, FRiVS is performed under semi- and uncontrolled capture conditions, and concept changes are observed in new reference ROI patterns that are sampled from the underlying data distribution. The refinement of previously-observed concepts (e.g., new ROIs are captured for a known face angle), corresponds to gradual changes (see Fig. 2(a)), and data corresponding to newly-observed concepts (e.g., new ROIs are captured under new illumination conditions), corresponds to abrupt changes (see Fig. 2(b)). In addition, a new concept (e.g., faces captured under natural vs. artificial lighting, or over a different face angle) can also correspond to a recurring change as specific observation conditions may be re-encountered in the future (see Fig. 2(c)). The rest of this section presents an overview of the different measures proposed in literature to detect changes, in order too choose the most adapted method for the proposed system. Then, specialized techniques that adapt classification systems to concept changes are reviewed, to introduce the proposed update strategies. Finally, a synthetic test case shows the benefit of using a change detection mechanism to guide the adaptation strategy used by the classification system according to the types of changes.

3.1. Detecting changes

In order to observe the occurrences of changes in the underlying data distribution, several families of measures have been proposed in the literature, which can be organised into techniques based on signal processing and pattern recognition. Prior to feature extraction, signal quality measures have been used to accept, reject, or reacquire biometric samples, as well as to select a biometric modality, algorithm, and/or system parameters [53]. In an FRiVS application, change detection can be performed by monitoring the values of an image-based quality over time. For example, several standards have been proposed to evaluate facial quality, such as ICAO 9303 [39], which cover image and face specific qualities. Other face quality measures compare input ROIs against facial references to assess image variations or distortions.

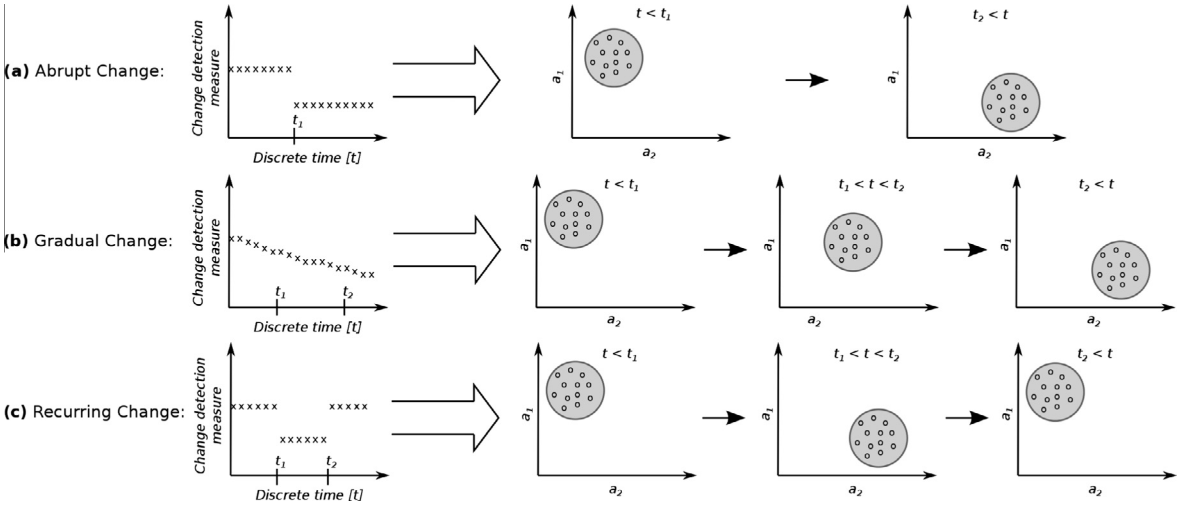


Fig. 2. Illustration of (a) abrupt, (b) gradual and (c) recurring changes occurring to a single concept over time. The first column presents an example of the evolution of values of a change detection measure, corresponding to variations to the 2-D data distribution to the right.

Change can also be measured after the feature extraction process of a biometric recognition system, and this paper focuses on pattern recognition techniques that rely on the feature distribution space, since the change detection process is designed to adapt the learning strategy of the classification module. These techniques fall into two categories: those that exploit classifier performance, and density estimation. Note that the accuracy of these measures depends heavily on the feature extraction and selection methods. Change detection mechanisms using classifier performance indicators have been considered for supervised learning applications [33]. For instance, changes can be detected in system performance using accuracy, recall or precision measures on the input data [23], or in the performance of a separate classifier dedicated to change detection, trained with the data corresponding to the last known change [2]. However, while directly monitoring the system's performance is a straightforward way to measure concept changes, it can also have several drawbacks. Relying on a classifier's performance for change detection may require a considerable amount of representative training data, especially when a classifier must be updated [2]. For this reason, the rest of this subsection will focus on density estimation measures and thresholding.

3.1.1. Density estimation measures

Although it may provide the most insight, detecting changes in the underlying distribution is very complex in the new data space. To reduce the computational complexity of change detection in the input feature space, several authors proposed to estimate the density of the data distribution. These techniques rely on fitting a statistical model to the previously-observed data, which distribution in the input feature space is unknown, and then applying statistical inference tests to evaluate whether the recently-observed data belong to the same model.

As presented by Kuncheva [36], clustering methods such as k-means or Gaussian Mixture Models (GMMs) may provide a compact representation of input data distributions in $\mathbb{R}^d$. In addition, Ditzler and Polikar [15] and Dries and Ruckert [17] proposed a non-parametric method that reduces the dimensionality of the incoming data blocks by representing them with feature histograms, with a fixed amount of bins. The following approaches have been proposed:

- Compute the **Likelihood** of the new data w.r.t. previously-generated model in order to quantify the probability that previous data blocks were sampled from the same concept. Kuncheva [36] proposed to detect changes monitoring the likelihood of new data, using GMM or k-means to model the previous concepts.
- Monitor the **model parameters**, such as mean vectors and covariance matrixes of k-means and GMM models, in order to evaluate their relative evolution [36], or polynomial regression parameters using the intersection of confidence interval rule, as proposed by Alippi et al. [2,3].
- Compare the estimated **densities** using measures like the Hellinger distances between consecutive histogram representation of data blocks [15], or a binary distance, assigning a binary feature to each histogram bin and evaluating their respective coverage [17].

Density estimation methods provide a lower level information than classifier performance indicators, and can therefore be more accurate for change detection. The performance indicators of classifiers trained over previously-encountered data are merely a consequence of possible changes in the underlying data distribution, while density estimation methods directly reflect the structure of underlying distributions. However, using a parametric estimation of density (e.g. GMM) makes strong assumptions concerning the underlying distribution of the input data [11], and the amount of representative data and the selection of the method parameters are critical factors in accurate estimation of densities. For these reasons, non-parametric density methods based on histogram representation will be considered in the proposed system.

3.1.2. Thresholding

The detection of changes for a one-dimensional data has been extensively studied in the area of quality control for monitoring process quality [36]. Assuming a stream of objects with a known probability p of being defective (given from product specifications) several control chart schemes have been proposed. According to the basic Shewhart control chart scheme, a batch or window of samples of V objects are inspected at regular intervals. The number of defective objects is counted, and an estimate $\bar{p}$ is plotted on the chart. Using a threshold of $f\sigma$, where $\sigma = \sqrt{p(1-p)/V}$ and the typical value of $f = 3$, a change is detected if $\bar{p} > p + f\sigma$. Among the numerous control chart approaches, the popular CUmulative Sum (CUSUM) proposed to monitor the cumulative sum of classification errors at time t. Similarly, change detection in pattern recognition usually monitor one or several classifier performance indicators over time, to observe various patterns of change in a stream of input patterns, producing the decision through thesholding. For example, in [32], the authors proposed a drift detection method to determine the optimal window size V containing the reference scores, which will be compared to the decision threshold. In the same way, the Hellinger Distance Drift Detection Method (HDDDM) method [15] proposes to reset the data distribution using density estimation of the current data block if the change is detected. As with Klinkenberg and Renz [32] and Gamma et al. [23], this method considers a growing window of samples (or data blocks), which reduces itself to the current data when a change is detected. In addition, decision is based on an adaptive threshold set on the previous values, adapting the final decision to the specific problem.

Given the changes that can occur in a FRiVS environment (and be observed from a set of ROI patterns extracted from a reference video), the system proposed in this paper will consider adaptive thresholding methods. In this case, decisions are based on the current capture conditions.

3.2. Adaptive classification for changing concepts

In the context of FRiVS, learning new reference ROI samples over time can raise the issue of preserving past knowledge, especially when new reference samples corresponding to new unknown concepts become available. Two categories of approaches have been proposed for supervised incremental learning of new concept in pattern recognition [33]:

1. updating a single incremental classifier, where new reference data are assimilated after their initial training;
2. adding or updating one or more classifiers to an ensemble trained with the new data.

Several monolithic classifiers have been proposed for supervised incremental learning of new labeled data, providing mechanisms to maintain an accurate and up-to-date class model [12]. For example, the ARTMAP [9] and Growing Self-Organizing [21] families of neural network classifiers have been designed with the inherent ability to perform incremental learning. Other popular classifiers such as the Support Vector Machine [52], the Multi-Layer Perceptron [10] and Radial Basis Function neural networks [46] have been adapted to perform incremental learning. However, these classifiers are typically designed under the assumption that data is sampled from a static environment, where class distributions remain unchanged over time.

Recently, Connolly et al. [13] proposed a Dynamic Particle Swarm Optimization (DPSO) based incremental learning strategy allowing to optimize and evolve all parameters of an ARTMAP neural network classifier, performing incremental learning an pursuing the optimization process to adapt to newly available data. However knowledge corruption is an issue with monolithic classifiers [50]. Incremental learning of significantly different and noisy data can degrade the previously-acquired knowledge. For example, with ARTMAP networks, learning such data can lead to a proliferation of category neurons on the hidden layer, causing a reduction in discrimination for older concepts and an increased computational complexity. As highlighted by the *plasticity–stability* dilemma [26], a classifier should remain stable w.r.t. previously-learned concepts, yet allow for adaptation w.r.t. relevant new concepts that emerge in new reference data.

In contrast, adaptive ensemble methods have been proposed, combining diversified classifiers into an ensemble to improve the system's overall performance and plasticity to new reference data. They can be divided into three general categories [34]:

1. *horse racing* methods, which train monolithic classifiers beforehand, and only adapt the combination rule dynamically [7,57];
2. methods using new data to update the parameters of ensemble's classifiers, in an online-learning fashion, like in [23]. In addition, Connolly et al. [14] proposed a DPSO-based incremental learning strategy to maintain an ensemble of optimized ARTMAP [9] classifiers.
3. hybrid approaches, adding new base classifiers as well as adapting the fusion rule, such as the Learn++ algorithm [50], based on the popular Adaboost [20], incrementally generates new classifiers for every new block of reference samples, and combines classifiers using weighted majority voting, the weights depending on the average normalized error computed during the generation process.

First of all, *horse racing* approaches cannot accommodate to new reference data since the classifiers in the ensemble are fixed, only the fusion rule changes. In addition, ensembles formed by online learners suffers from the same knowledge corruption issues than monolithic incremental classifiers. For example, in [14], the ARTMAP classifiers of the MCS updated with new reference data over time are subject to knowledge corruption, as with the monolithic architectures using such classifiers. However, hybrid approaches provide a compromise between *stability* and *plasticity* to new data. Classifiers trained on previously acquired data, remains intact, while new classifiers are trained for the new reference data. For example, using the Learn++ algorithm [50], an ensemble is incrementally grown using, at each iteration, a weight distribution giving more importance to reference samples previously mis-classified, thus generating new classifiers specialized on the most difficult samples. Those systems may avoid knowledge corruption, but at the expense of growing system complexity, as new classifiers or reference samples are added to the ensemble for every new block of data. In addition, the update of the fusion rule tends to favor more recent concepts, as the weights of previously learned classifiers tend to decline.

More recently, approaches using a change detection mechanism to drive ensemble or incremental based adaptation strategies have been proposed. Minku and Yao [42] proposed the Diversity for Dealing with Drifts algorithm, which maintains two ensembles with different diversity levels, one low and one high, in order to assimilate a new concept emerging in the observed data. When a significant change is detected though the monitoring of the system's error rate, the high diversity ensemble is used to assimilate new data and converge to a low diversity ensemble, and a new high diversity one is generated and maintained through bagging. Alippi et al. [3] also proposed a Just-in-Time classification algorithm, using a density-based change detection to regroup reference samples per detected concept, and update a on-line classifier using this knowledge when the observed data drift toward a known concept.

While these methods effectively rely on change detection and ensemble or incremental learning to adapt in changing environments, they emphasize newer concepts, through weighing or by discarding of the classifiers trained on previously-learned concepts. This can corrupt a FRiVS system's performance where every newer, and older and recurring concepts, are equally important.

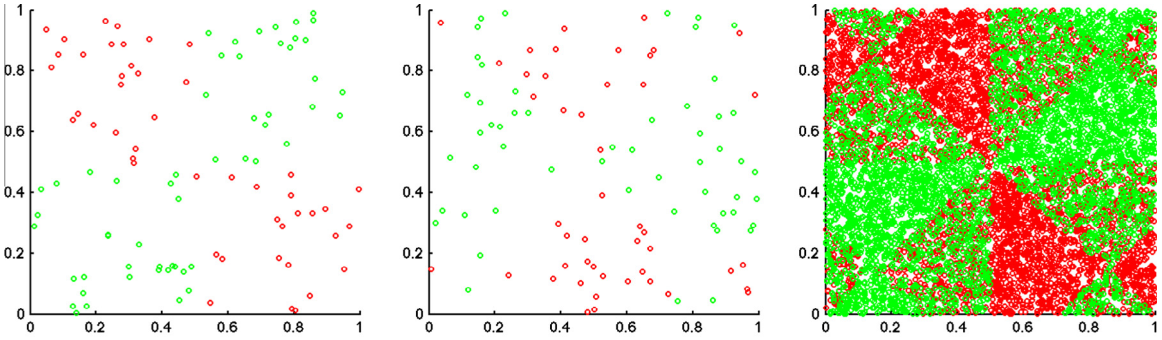
In Section 4, a new approach is proposed to adapt ensembles to new ROI reference patterns in a video-surveillance environment. It relies on the hypothesis that, when new reference data become available to adapt a facial models, and that the data incorporate an abrupt pattern of change w.r.t. to previously-learned concepts, previously-learned knowledge is better preserved with a learn-and combine strategy, instead of updating the previously-trained ones. As opposed to in literature, the resulting ensemble is specialized in every detected concept. The complexity of the system is controlled by the change detection mechanism, where a classifier is only added if significantly different reference data is presented, and knowledge of different concepts is updated over time when similar reference data are presented.

3.3. Synthetic test case: influence of changing concepts

A synthetic test case is now presented, to validate the intuition that, when new reference data incorporating abrupt changes w.r.t. previously-learned ones is presented to the system, it is more beneficial to employ a *learn-and-combine* strategy than updating previously-trained classifiers. This test case simulates a video person re-identification scenario: the FRiVS system operates in an environment where face captures may be sampled from different concepts (such as face orientation angle). This test case seeks to illustrate that when two significantly different sequences of data (abrupt change) are presented to a FRiVS system by the operator for update, training dedicated classifiers for each different concept provides better performance.

Consider a system designed to detect ROI samples from a target individual, among ROI samples from unknown non-taget individual. Two tagged data blocs, $Vs[1]$ and $Vs[2]$, are presented to the system, at time $t = 1$ (initial training) and the other at $t = 2$ (update during later operations), by the operator. Those blocks are comprised of reference patterns from the target and the non-target class, generated from two synthetic 2-dimensional sources, inspired from the rotating checkerboard classification problem [33] (Fig. 3), which provides samples distributed along a 2 × 2 checkerboard. In order to simulate the arrival of a new concept, $Vs[1]$ is composed of patterns from the initial checkerboard (Fig. 3(a)), and $Vs[2]$ from the checkerboard rotated by an angle of $\pi/4$ (Fig. 3(b)). At $t = 2$, the introduction of $Vs[2]$ represents an abrupt change w.r.t. to the patterns from $Vs[1]$. These are sampled from a different concept than the one modeled by the system at $t = 1$.

The operational mode is simulated by a combination of test blocks Vs_{test}, composed by target and non-target patterns originating from both concepts (Fig. 3(c)). As $Vs[1]$ and $Vs[2]$ incorporate data corresponding to two different concepts present in Vs_{test}, the update of the system with $Vs[2]$ should not corrupt previously-acquired knowledge, as it also corresponds to relevant information about Vs_{test}. This simulates a FRiVS scenario, where the operator gradually present the systems with tagged reference video sequences containing data from different concepts, e.g. different observation conditions such as



(a) Vs_1 (b) Vs_2 (c) Vs_{test}

Fig. 3. Reference and operational video sequences for the synthetic test case [33]. Target class samples are represented in gray, and non-target ones in black.

Table 2
Performance for the rotating checkerboard data of a PFAM-based system updated through incremental learning and through the learn-and-combine strategy. In the latter case, the classifiers are fused using the average score-level rule. Arrow ↑ (↓) represents a measure that should be maximised (minimised). Performance measures are defined in Section 5.4.

Performance measures	Enrolment with data from with a single classifier $Vs[1]$	Update with data from $Vs[2]$ with	
		Incremental	L&C
$pAUC(5\%)(\uparrow)$	7.2% ± 0.4	6.2% ± 0.6	**8.2**% ± 0.6
$fpr(\downarrow)$	17.07% ± 1.06	**9.85%** ± 0.96	14.94% ± 1.02
$tpr(\uparrow)$	37.72% ± 2.17	16.19% ± 1.61	**32.5%** ± 2.27
$F_1(\uparrow)$	46.41% ± 1.94	23.63% ± 2.01	**41.35%** ± 2.21
Complexity(↓)	6.14 ± 0.23	**13.1** ± 0.3	17.0 ± 0.42

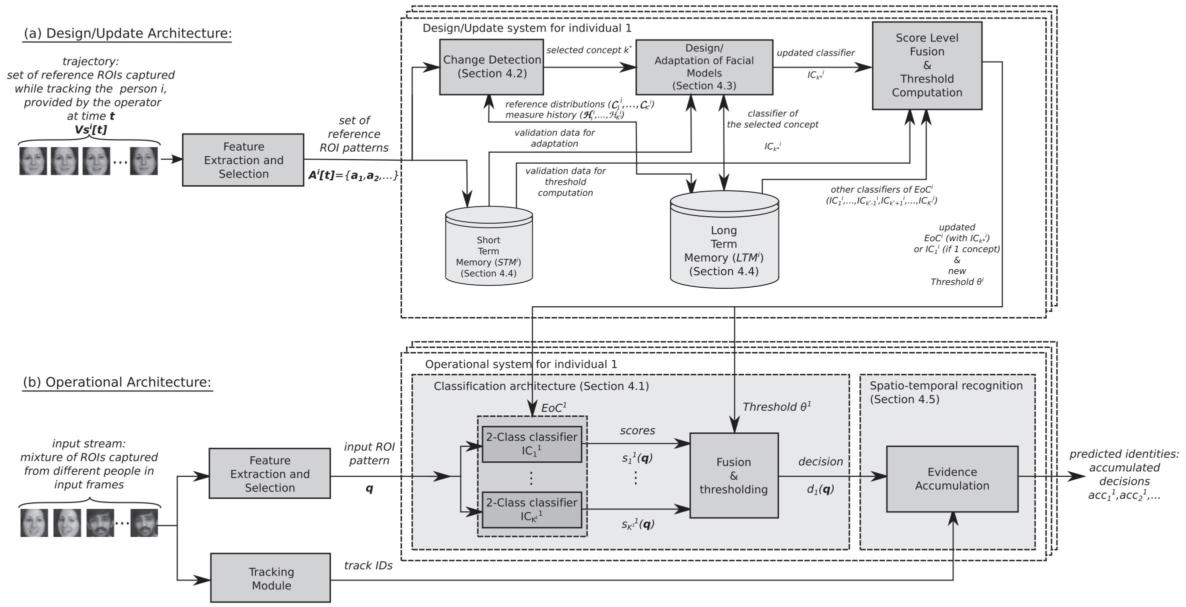

(a) Design/Update Architecture:
trajectory: set of reference ROIs captured while tracking the person i, provided by the operator at time t
$Vs^i[t]$
Feature Extraction and Selection
set of reference ROI patterns
$A^i[t]=\{a_1,a_2,...\}$
Design/Update system for individual 1
Change Detection (Section 4.2)
selected concept k^*
Design/ Adaptation of Facial Models (Section 4.3)
updated classifier
$IC_{k^*}^i$
Score Level Fusion & Threshold Computation
reference distributions $(\mathcal{C}_1^i,...,\mathcal{C}_{K^i}^i)$
measure history $(\mathcal{H}_1^i,...,\mathcal{H}_{K^i}^i)$
validation data for adaptation
classifier of the selected concept
$IC_{k^*}^i$
validation data for threshold computation
Short Term Memory (STM^i) (Section 4.4)
Long Term Memory (LTM^i) (Section 4.4)
other classifiers of EoC^i
$(IC_1^i,...,IC_{k^*-1}^i,IC_{k^*+1}^i,...,IC_{K^i}^i)$
updated EoC^i (with $IC_{k^*}^i$) or IC_1^i (if 1 concept) & new Threshold θ^i
(b) Operational Architecture:
input stream: mixture of ROIs captured from different people in input frames
Feature Extraction and Selection
input ROI pattern
q
Tracking Module
track IDs
Operational system for individual 1
Classification architecture (Section 4.1)
EoC^1
2-Class classifier IC_1^1
2-Class classifier $IC_{K^1}^1$
scores
$s_1^1(\mathbf{q})$
$s_{K^1}^1(\mathbf{q})$
Threshold θ^1
Fusion & thresholding
decision
$d_1(\mathbf{q})$
Spatio-temporal recognition (Section 4.5)
Evidence Accumulation
predicted identities: accumulated decisions $acc_1^1,acc_2^1,...$

Fig. 4. Architecture of the proposed $AMCS_{CD}$ for FRiVS. The design and update architecture for individual of interest i is presented in (a), and the operational architecture (for all individuals) in (b).

camera angle, that are equally important in the system's operation, as future input patterns (ROIs) the system will capture in operations can correspond to any of those concept (face angle).

Two different training strategies are compared. With the *incremental* strategy, $Vs[1]$ and then $Vs[2]$ are learned incrementally by a Probabilistic Fuzzy ARTMAP (PFAM) [38] classifier. With the *learn and combine* strategy, $Vs[1]$ and $Vs[2]$ are learned by two different PFAM classifiers, forming an ensemble which output is combined by the *average* score fusion rule. The *learn and combine* strategy is an implementation of the proposed approach, by assuming a perfect mechanism to detect an abrupt change (a new concept) with the sequence $Vs[2]$ at $t = 2$ w.r.t. to $Vs[1]$.

Each PFAM classifier is trained with standard hyper-parameters values $\mathbf{h} = (\alpha = 0.001, \beta = 1, \varepsilon = 0.001, \bar{\rho} = 0, r = 2)$. $Vs[1]$ and $Vs[2]$ are composed of 50 target and 50 non-target patterns, and Vs_{test} of 2000 target and 2000 non-target patterns originating from both concepts (1000 patterns for each concept). In addition, two validation blocks $Vv[1]$ and $Vv[2]$ of 25 target and 25 non-target patterns each are considered to select a threshold that respects the operational constraint of $far \leqslant 5\%$. The operating point is selected based on ROC curve produced by systems adapted using the incremental and learn and combine strategy over the two validation datasets, and the performance is measured in terms of partial AUC ($pAUC$) for $fpr \in [0, 00.05], fpr, tpr$ and F_1 measures for selected operating points. In addition, the complexity of the systems are evaluated by counting the sum of F_2 layer neurons (category prototypes) of the PFAM classifiers. As the dataset is randomly generated from the sources, the simulations have been repeated for 100 replications. Results presented in Table 2 are the average values and the standard deviation, computed using a Student distribution and a confidence interval of 10%.

As shown in Table 2, after updating classifiers with data from the second concept of $Vs[2]$, the $pAUC(5\%)$ of the *incremental* PFAM strategy declines slightly, which is a consequence of knowledge corruption (due to the learning of new data exhibiting significant concept change). While the increase of complexity is lower for the *incremental* strategy, it can be noted that the number of prototypes doubles when the system is presented with data from $Vs[2]$. This proliferation is a consequence of the incremental learning of two significantly different blocks of data.

Both systems start with the same performance level (after training with $Vs[1]$), but the training for the second concept with the *learn and combine* strategy generates significant increase in performance in terms of $pAUC, tpr$ and F_1. Although the *fpr* decreases more with the *incremental* strategy, the decline in *tpr* and F_1 is considerably lower compared to the *learn and combine* strategy. Overall, this synthetic test case shows the benefit of training new classifiers to learn from new reference data that exhibit significant (abrupt changes). When presented with data from $Vs[2]$, the *learn-and-combine* strategy enabled to increase the system's performance, while the *incremental* strategy is unable to preserve previously acquired knowledge.

4. An adaptive multi-classifier system with change detection

Fig. 4 presents an Adaptive Multi-Classifier System with Change Detection ($AMCS_{CD}$) specialized for video-to-video FR, with a novel updating strategy based on change detection. The main intuition at the origin of this contribution is that, when new reference samples become available to adapt a facial model, and that the data incorporate an abrupt change compared to existing concepts in the system, it is more beneficial to design a new dedicated classifier on the data and combine it to previously-learned classifiers in an ensemble (learn-and combine strategy), instead of updating the previously-trained ones (incremental learning strategy). This enables to maintain an up-to-date representation of every concept encountered in the reference data, and avoid knowledge corruption when presented with new reference video encompassing new concepts in the feature space (e.g. face poses, illumination conditions,...).

For each individual $i = 1, \ldots, I$ enrolled to the system, this modular system is composed by an ensemble of K^i incremental two-class classifiers $EoC^i = \{IC^i_1, \ldots, IC^i_{K^i}\}$, where $K^i \geqslant 1$ is the number of concepts detected in the individual's reference videos, and a user specific threshold θ^i. The supervised learning of new reference sequences by the incremental classifiers is handled by a design and adaptation module, guided by change detection. For each individual, this module relies on long-term memory LTM^i to store the concept representations $\{C^i_1, \ldots, C^i_{K_i}\}$, and a short term memory STM^i to store reference data for design or adaptation and for validation.

Algorithm 1. Strategy to design and update the facial model of individual i

```
1  Input: Sequence of reference ROIs for individual i $Vs^i[t]$, provided by the operator at time t.;
2  Output: Updated ensemble $EoC^i$;
3  - Compute $\mathbf{A}^i[t]$, the set of reference ROI patterns obtained after feature extraction and selection of ROIs of
   $Vs^i[t]$ ;
4  - $STM^i \leftarrow \mathbf{A}^i[t]$;
5  for each concept $k \leftarrow 1$ to $K^i$ do
6      - Measure $\delta^i_k[t]$ the distance between $\mathbf{A}^i[t]$ and the concept representation $C^i_k$;
7      - Compare $\delta^i_k[t]$ to the change detection threshold $\beta^i_k[t]$ of the concept $k$;
8  if $\delta^i_k[t] > \beta^i_k[t]$ for each concept $k = 1,...,K_i$, or $K_i = 0$ then
9      //An abrupt change is detected or no concepts are stored;
10     - $K^i \leftarrow K^i + 1$;
11     - Set index of the chosen concept $k^* \leftarrow K^i$;
12     - Generate the concept representation $C^i_{K^i}$ from $\mathbf{A}^i[t]$ and store in $LTM^i$;
13     - Initiate and train new classifier $IC^i_{K^i}$ and the user-specific threshold $\theta^i$ using (target and non-target) data
       from $STM^i$;
14     - Update $EoC^i \leftarrow \{EoC^i, IC^i_{K^i}\}$;
15 else
16     //A moderate change has been detected;
17     - Determine the index of the closest concept $k^* \leftarrow \min\{\delta^i_k[t] : k = 1,...,K^i\}$;
18     - Update the corresponding incremental classifier $IC_{k^*}$ of $EoC^i$ and the user-specific threshold $\theta^i$ using data
       from $STM^i$;
```

Overall training/update process: The class-modular architecture for the proposed AMCS allows to design and update facial models independently for each individual of interest (see Algorithm 1 and Fig. 4(a)). When a new reference video sequence $Vs^i[t]$ is provided by the operator at time t, relevant features are first extracted and selected from each ROI in order to produce the set of input patterns $\mathbf{A}^i[t]$ (Algorithm 1, line 1). STM^i temporarily stores validation data used for classifier design and threshold selection (Algorithm 1, line 4). The change detection process assess whether the underlying data distribution of $\mathbf{A}^i[t]$ exhibits significant changes compared to previously-learned data. For this purpose, the previously-observed concepts $\{C^i_1, \ldots, C^i_{K^i}\}$ stored in LTM^i are compared to a histogram representation of $\mathbf{A}^i[t]$ (Algorithm 1, lines 6–7). If a significant (abrupt) change (Fig. 2 and Table 1) is detected w.r.t. all the stored concept models, or if $Vs^i[t]$ is the first reference sequence for the individual (no previous concept has been stored), a new concept is assumed (Algorithm 1, line 8). In this case, K^i is incremented, and a new incremental classifier $IC^i_{K^i}$ is designed for the concept (IC^i_1 if the first concept) and the user-specific threshold θ^i is updated (or created) using the training and adaptation module with the data from STM^i (Algorithm 1, line 10–13). Note that the training of the classifier is done using non-target reference patterns (from other individuals) mixed to target reference patterns from $\mathbf{A}^i[t]$. When a moderate (gradual) change is detected, the classifier $IC^i_{k^*}$ corresponding to the closest concept representation $C^i_{k^*}$ is updated and evolved through incremental learning, and the user-specific threshold θ^i is updated as well (Algorithm 1, lines 17 and 18). Finally, if several concepts are stored in the system, the EoC^i is updated to combine the most accurate classifiers of the known concepts: if a new concept has been detected, a new classifier $IC^i_{K^i}$ is added to EoC^i (Algorithm 1, line 14), and if a known concept k^* is updated, the corresponding classifier $IC^i_{k^*}$ is updated (Algorithm 1, line 18). If only one concept has been detected, a single classifier is assigned to the individual, $EoC^i = IC^i_1$.

Algorithm 2. Operational strategy for one individual *i*

```
1  Input: Stream of input ROIs of the observed individuals, ensemble of classifiers $EoC^i$ for individual $i$;
2  Output: Accumulated decisions $\{acc^i_1, ..., acc^i_J\}$ for the J tracks detected in the set of input ROIs.;
3  for each ROI $r = 1, 2, ...$ do
4      - Perform feature extraction and selection to obtain input pattern $\mathbf{q}_r$;
5      - Determine the track ID $tr(\mathbf{q}_r)$;
6      for each concept $k \leftarrow 1$ to $K^i$ do
7          - Compute the positive matching score for $\mathbf{q}_r$ with the $k^{th}$ classifier $s^i_k(\mathbf{q}_r)$;
8      - Perform fusion of the $K^i$ scores and apply user specific thresholds $\theta^i$ to obtain the ensemble decision $d^i(\mathbf{q}_r)$;
9  for each detected track $j \leftarrow 1$ to $J$ do
10     for each ROI $r = 1, 2, ...$ do
11         if $tr(q_r) = j$ then
12             - $acc^i_j \leftarrow acc^i_j + d^i(\mathbf{q}_r)$;
```

Overall operational process: During operations, the AMCS functions according to Algorithm 2 and Fig. 4(b). When a ROI is detected in a new area of the input scene, a face tracker is initiated with the ROI, assigning it a track ID number $j = 1, \ldots, J$. Then, the tracker produces the same track ID number for that face in subsequent frames. An input stream is thus a mixture of ROIs from different people, each one is associated with a track ID number $j = 1, \ldots, J$. In parallel, the system extracts and selects input ROI patterns $\mathbf{q}$ in the same way than the update process (Algorithm 2, line 3). Each input $\mathbf{q}_r$ is associated with its track number $tr(\mathbf{q}_r) \in (1, \ldots J)$. For each individual i enrolled to the system, the final decision from the EoC^i $d^i(\mathbf{q}_r)$ is computed from the independent scores $s^i_k(\mathbf{q}_r)$ $(k = 1, \ldots, K^i)$ of the classifiers (Algorithm 2, line 7), fusing them in the score or decision level (Algorithm 2, line 8) and applying user-specific thresholds θ^i. Finally, the identity predictions are generated through the accumulation of decisions per track using the track IDs: for each track $j = 1, \ldots, J$, the decisions based on ROI patterns associated with this ID are accumulated to output the final decision (Algorithm 2, line 12) according to:

$$acc^i_j = \left\{ \sum_{\mathbf{q}_r \in inputstream} d^i(\mathbf{q}_r); \; tr(\mathbf{q}_r) = j \right\} \tag{1}$$

The rest of this section provides more details on the different modules inside the AMCS. For each module, a particular implementation is also described, in order to build a fully-functional system.

4.1. Classification architecture

In operational mode (Algorithm 2), the classification system seeks to produce a binary decision $d^i(\mathbf{q}_r)$ in response to each input pattern $\mathbf{q}_r$ submitted to the system for each module i. If $d^i(\mathbf{q}_r) = 1$, the system has matched the facial capture $\mathbf{q}_r$ to the enrolled individual i. Module i is comprised of a single 2-class incremental classifier IC^i_1 or an ensemble $EoC^i = \{IC^i_1, \ldots, IC^i_{K^i}\}$ per enrolled individual i, as well as a user-specific decision threshold θ^i. Usually, $\mathbf{q}_r$ is a pattern generated from an ROI sample extracted from a continuous video stream.

A specific implementation: The classification architecture is composed of IC^i_k that are 2-class Probabilistic Fuzzy ARTMAP (PFAM) [38] incremental classifiers, where each one is trained using a balanced sets of references samples from the target individual (from trajectories) against a random selection of non-target data from an universal and cohort model (UM and CM). PFAM classifier is a versatile classifier that is known to provide a high level of accuracy with moderate time and memory complexity [38]. It is promising for face matching due to its ability to perform fast, stable, on-line, unsupervised or supervised, and incremental learning from limited amount of training data. Although trained for different concepts, the classifiers of every ensemble are designed using ROIs of the same individual, and can thus be considered as correlated. For this reason, following the recommendations in [31], the score-level *average* fusion rule is used to combine the decisions in operational mode, producing the final ensemble's decision through the averaging of the classifier's scores. Finally, the authors have previously compared three classification architectures for a FRiVS system [48]: (1) a *global or monolithic architecture* composed of a single multi-class PFAM classifier, trained to detect the presence of all individuals of interest, (2) a *class-modular architecture* composed of a 2-class PFAM classifier per individual, and (3), a *class-modular architecture with ensembles of classifiers* composed of an ensemble of 2-class PFAM classifiers per individual. The latter was known to outperform other architectures when working with real video-based data.

The original fuzzy-ARTMAP classifier [9] is composed by three layers: (1) the input layer $F1$ of $2D$ neurons (D being the dimensionality of the feature space), (2) a competitive layer $F2$ in which each of the N neuron corresponds to a category hyper-rectangle in the feature space, and (3), a map field of L output neurons (the number of classes, in that case $L = 2$). Connections between F_1 and F_2 are represented by a set of real-valued weights $\mathbf{W} = \{w_{dn} \in [0,1] : d = 1, 2, \ldots, D;\ n = 1, 2, \ldots, N\}$, and a category n is defined by a prototype vector $w_n = (w_{1n}, w_{2n}, \ldots, w_{Dn})$. The F_2 layer is also connected to the F^{ab} layer through the binary-valued weight set $W^{ab} = \{w^{ab}_{nl} \in 0, 1 :\ n = 1, 2, \ldots, N;\ l = 1, 2, \ldots, L\}$. Vector $w^{ab}_n = (w^{ab}_{n1}, w^{ab}_{n2}, \ldots, w^{ab}_{nL})$ represents the link between the F_2 category node n and one of the L F^{ab} class nodes. In supervised training mode, the synaptic weights are adjusted to the training patterns by (1) learning category hyper-rectangles in the feature space and (2) associating them to the corresponding output classes. PFAM classifier [38] relies on the fuzzy ART clustering and MAP field in order to approximate to the underlying data distribution as a mixture of Gaussian distributions in the feature space, and generates of prediction scores instead of binary decisions. In addition to FAM category hyper-rectangles and $F_2 - F^{ab}$ connexions, PFAM also learns prior probabilities $p(i)$ for each class i, categories center $\mathbf{w}^{ac}_n$ and covariance matrices Σ_n for each category n. PFAM dynamics are governed by a vector of five hyper-parameters $\mathbf{h} = (\alpha, \beta, \varepsilon, \bar{\rho}, r)$: the choice parameter $\alpha > 0$, the learning parameter $\beta \in [0,1]$, the match-tracking parameter $\varepsilon \in [-1,1]$, the vigilance parameter $\bar{\rho} \in [0,1]$, and the smoothing parameter $r > 0$.

As FAM or PFAM classifiers categorize the feature space into hyper-rectangles or Gaussian distributions (priors, centers and covariance matrices) during training, their memory complexity and processing time in operations depend on the number of categories, or prototypes (Gaussian centers). The operational memory complexity of classification systems using PFAM classifiers can thus be compared based on the number of prototypes.

4.2. Change detection

A change detection (CD) module (see Fig. 5) is proposed to distinguish abrupt from gradual changes that have emerged from the underlying distribution. It allows to trigger one of the strategies to adapt facial models in the AMCS. For each individual i, this module relies on a set of concept representations $\{\mathcal{C}^i_1, \ldots, \mathcal{C}^i_{K^i}\}$ and an history of distance measures $\{\mathcal{H}^i_1, \ldots, \mathcal{H}^i_{K^i}\}$ between all the previously-learned sequences of reference ROI patterns for individual i, and each concept representation. When a new reference pattern set $A^i[t]$ (extracted from a sequence $Vs^i[t]$) is presented to the system, the CD module detects if it differs significantly from previously-learned concepts. The input distribution $\mathcal{A}$ is extracted, and change is measured w.r.t. all stored concept representations $\{\mathcal{C}^i_1, \ldots, \mathcal{C}^i_{K^i}\}$. For each stored concept k, the measure $\delta^i_k[t]$ is compared to an adaptive threshold $\beta^i_k[t]$, computed from the measure history of the concept $\mathcal{H}^i_k$. The most appropriate concept k^* is then selected and provided to the adaptation module.

A specific implementation: Changes are detected using the HDDM presented in [15], and the concepts are represented as histograms $\mathcal{C}^i_k$. This method provides a non-parametric low complexity detection measure though discretization of the feature space, which is a compromise between the precision of low level change detection of the density methods and the low complexity of the performance-based ones. In addition detections are based on the current contextual environment thanks to the adaptive threshold computation.

Algorithm 3. Specific implementation of the HDDM based CD procedure for individual i.

```
Input: Set of ROI patterns for individual i provided by the operator at time t, $\mathbf{A}^i[t] = \{\mathbf{a}_1, \mathbf{a}_2, ...\}$;
Output: Index $k^*$ of the selected concept;
- Generate histogram $\mathcal{A}$ of $\mathbf{A}^i[t]$;
if $K_i = 0$ then
    - $newConcept \leftarrow true$;
else
    for $k \leftarrow 1$ to $K^i$ do
        - $\delta^i_k[t] \leftarrow$ Hellinger distance between $\mathcal{A}$ and $\mathcal{C}^i_k$;
        - Update threshold $\beta^i_k[t]$;
        if $\delta^i_k[t]) \leq \beta^i_k[t]$ then
            - $newConcept = newConcept \& false$;
if $newConcept == true$ then
    - $K^i \leftarrow K^i + 1$;
    - Store $\mathcal{C}^i_{K^i} \leftarrow \mathcal{A}$ into $LTM^i$; //Initialization of the measure history $\mathcal{H}^i_{K^i}$
    for $r \leftarrow nRep$ do
        - Separate $\mathcal{C}^i_{K^i}$ into two sub-blocks $\mathbf{c}_1$ and $\mathbf{c}_2$ using the k-means algorithm;
        - Compute $\delta_m(r)$, the Hellinger distance between $\mathbf{c}_1$ and $\mathbf{c}_2$;
    - Re-organize measures $\{\delta_m(1), ..., \delta_m(nRep)\}$ in descending order;
    - Initialize $\mathcal{H}^i_{K^i} \leftarrow \{\delta_m(1), \delta_m(2)\}$;
else
    - Select the index of the concept to update $k^* = \min\{\delta^i_k[t]; \delta^i_k[t] \leq \beta^i_k[t], k = 1...K^i\}$ ;
    - Update the concept model $\mathcal{C}^i_{k^*} \leftarrow \mathcal{C}^i_{k^*} + \mathcal{A}$;
    - Update the measure history $\mathcal{H}^i_{k^*} \leftarrow \{\mathcal{H}^i_{k^*}, \delta^i_{k^*}[t]\}$;
```

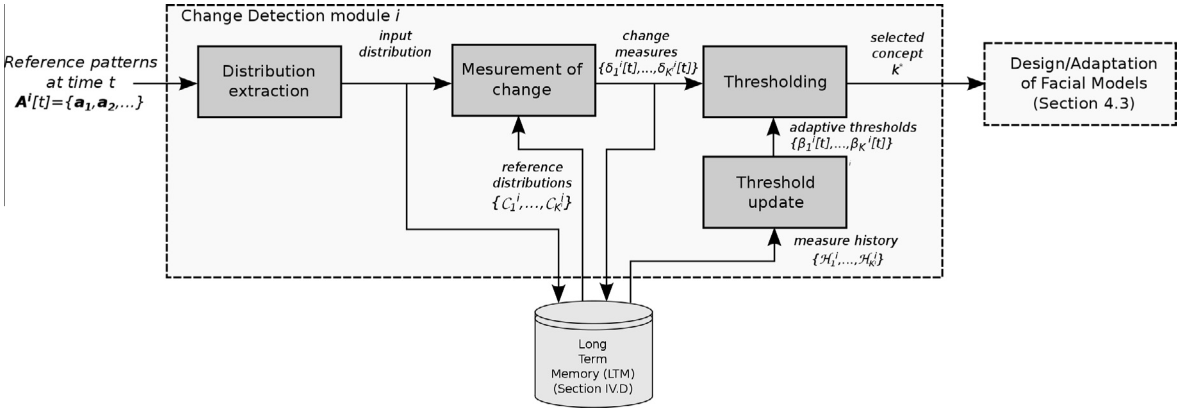

Change Detection module i
Reference patterns at time t
$\mathbf{A}^i[t]=\{\mathbf{a}_1,\mathbf{a}_2,\ldots\}$
Distribution extraction
input distribution
Mesurement of change
change measures
$\{\delta_1^i[t],\ldots,\delta_K^i[t]\}$
Thresholding
selected concept
k^*
Design/Adaptation of Facial Models (Section 4.3)
reference distributions
$\{\mathcal{C}_1^i,\ldots,\mathcal{C}_K^i\}$
adaptive thresholds
$\{\beta_1^i[t],\ldots,\beta_K{}^i[t]\}$
Threshold update
measure history
$\{\mathcal{H}_1^i,\ldots,\mathcal{H}_K^i\}$
Long Term Memory (LTM) (Section IV.D)

Fig. 5. Architecture of the CD module *i*.

The HDDM-based [15] CD process for each individual i is presented in Algorithm 3. The reference sequence's histogram $\mathcal{A}$ is first computed from the patterns $\mathbf{A}^i[t]$, after feature extraction and selection (Algorithm 3, line 3). Then, for each saved concept $k = 1, \ldots, K^i$, the Hellinger distance $\delta_k^i[t]$ is computed between histogram $\mathcal{A}$ and the concept representation $\mathcal{C}_k^i$ (Algorithm 3, line 8), following:

$$\delta_k^i[t] = \frac{1}{D}\sum_{d=1}^{D}\sqrt{\sum_{b=1}^{B}\left(\sqrt{\frac{\mathcal{A}(b,d)}{\sum_{b'=1}^{B}\mathcal{A}(b',d)}} - \sqrt{\frac{\mathcal{C}_k^i(b,d)}{\sum_{b'=1}^{b}\mathcal{C}_k^i(b',d)}}\right)^2} \tag{2}$$

where D is the dimensionality of the feature space, B the number of bins in $\mathcal{A}$ and $\mathcal{C}_k^i$, $\mathcal{A}(b,d)$ and $\mathcal{C}_k^i(b,d)$ the frequency count in bin b of feature d. An abrupt change between the histogram $\mathcal{C}_k^i$ of concept k and $\mathcal{A}$ is detected if $\delta_k^i[t] > \beta_k^i[t]$, where $\beta_k^i[t]$ an adaptive threshold computed from the previous distance measures according to [15]:

$$\beta_k[t] = \hat{\mathcal{H}}_k^i + t_{\alpha/2}\cdot\frac{\hat{\sigma}}{\sqrt{\Delta_t}} \tag{3}$$

where α is the confidence interval of the t-statistic test, Δ_t the total amount of past distance measures stored in $\mathcal{H}_k^i$, and $\hat{\mathcal{H}}_k^i$ and $\hat{\sigma}$ the average and variance of those measures. If an abrupt change is detected for all the concepts (Algorithm 3, line 12), or if $A^i[t]$ is the first sequence of reference ROI patterns provided for the individual i (Algorithm 3, line 5), a new concept is added to the system. The number of concepts K^i for the individual i is incremented, and $\mathcal{A}$ is memorized into LTM^i as histogram $\mathcal{C}_{K^i}^i$ (Algorithm 3, line 14).

The measure history $\mathcal{H}_{K^i}^i$ is initialized by: (1) separating of $\mathcal{A}$ into two sub-blocks $\mathbf{c}_1$ and $\mathbf{c}_2$ using the k-means algorithm (Algorithm 3, line 15) and (2) computing the Hellinger distance δ_m between the 2 sub-blocks (Algorithm 3, line 16). As the initialization of the k-means algorithm is random, this process is repeated for several replications *nRep*, and the 2 longest distances are stored in the concept's memory $\mathcal{H}_{K^i}^i$ (Algorithm 3, lines 17 and 18). The choice of the longest distances enables to generate a more permissive threshold for the subsequent reference sequences. It is considered as an estimation of the longest tolerable distance between reference sequences from the same concept. In addition, at least 2 measures must be selected in order to compute a proper variance for the next change detection.

If at least one comparison does not trigger CD, the closest reference histogram $\mathcal{C}_{k^*}^i$ (Algorithm 3, line 19) is selected, and updated using $\mathcal{A}$ following $\mathcal{C}_{k^*}^i \leftarrow \mathcal{C}_{k^*}^i + \mathcal{A}$ (Algorithm 3, line 20). The distance $\delta_{k^*}^i[t]$ is added into the concept's measure history $\mathcal{H}_{k^*}^i$ (Algorithm 3, line 21).

This CD mechanism allows for selective windowing over the training data, as several reference distributions $\{\mathcal{C}_1^i, \ldots, \mathcal{C}_{K_i}^i\}$ are stored. In addition, each histogram representations of a distribution $\mathcal{C}_k^i$ is paired with an adaptive threshold $\beta_k^i[t]$ in order to adapt the decision for specific reference samples. Finally, this strategy can handle recurring concept changes if $A^i[t]$ is composed of data similar to a previously encountered concept k^*. In this case, only the corresponding classifier will be updated.

4.3. Design and adaptation of facial models

This module is dedicated to the design and update of incremental-learning classifiers IC_k^i limited to individual i. It relies on the last state (internal and hyper-parameters) of the previously-learned classifiers, reference target and non-target ROI patterns from STM^i as well as the output k^* of the CD module. If $k^* = K^i$, i.e. a new concept is detected, a new incremental-learning classifier $IC^i_{K^i}$ is initiated and trained on $A^i[t]$. Otherwise, the classifier $IC^i_{k^*}$ is updated incrementally with $A^i[t]$.

A specific implementation: A Dynamic Particle Swarm Optimization (DPSO) training strategy is employed to train and optimize the PFAM classifiers. This incremental learning strategy that evolves pools of incremental learning classifiers in the hyper-parameter space has been described and applied to adaptive FR systems in [13].

For each individual *i*, this module relies on a pool of PFAM classifiers $\mathcal{P}^k_1$ per concept *k* ($k = 1, \ldots K^i$). Each pool consists of classifiers trained with reference ROI samples from concept *k*, to produce the best (global best) classifier IC^i_k. It may be combined in EoC^i with best classifiers from other concepts. This DPSO incremental-learning strategy allows to co-jointly optimize PFAM parameters (internal weights $\mathbf{W}, \mathbf{W}^{ab}, \mathbf{W}^{ac}$ and Σ, hyper-parameters $\mathbf{h}$, and architecture) of the classifiers in $\mathcal{P}^i_k$ such that the fitness function (classification accuracy) is maximized. The DPSO algorithm has been chosen for its convergence speed, and the DPSO training strategy has already been successfully applied in state-of-the art adaptive face recognition systems in video [13].

More precisely, PSO is a population based stochastic optimization technique inspired by the behaviour of a flock of birds [18]. In this implementation, each particle of a swarm moving in the optimization space is defined by the five hyper-parameters $\mathbf{h} = (\alpha, \beta, \varepsilon, \bar{\rho}, r)$ of a PFAM classifier. The particles move in the optimization space according to two factors: (1) their *cognitive influence* (previous search experience) and (2), the *social influence* (other particles' experience, in a neighbourhood). At a discreet iteration τ, the position (hyper parameters) of each particle (classifier) $\mathbf{h}(\tau)$ changes according to its inertia and the *cognitive* and *social* influences following Eq. (4), with w_0, w_1 and w_2 the inertia, cognitive and social weights, and r_0, r_1 and r_2 random parameters.

$$\mathbf{h}(\tau) = r_0 \cdot w_0(\mathbf{h}(\tau) - \mathbf{h}(\tau - 1)) + r_1 \cdot w_1(\mathbf{h}_{cog} - \mathbf{h}(\tau)) + r_2 \cdot w_2(\mathbf{h}_{soc} - \mathbf{h}(\tau)) \tag{4}$$

During optimization, each particle thus begins at its current location, then continues moving in the same direction it was going according to the inertia weight while being attracted by each source of influence:

- Its best known position $\mathbf{h}_{cog}$, the cognitive influence, also known as its memory.
- The best position of the swarm $\mathbf{h}_{soc}$, the social influence.

The best position is defined using a fitness function, which is, in this case, the classification performance over validation data stored in STM^i of the classifiers trained with training data, with the hyper-parameters corresponding to the positions of the particles. When new reference data become available, or if an abrupt change is detected, a new pool of classifiers $\mathcal{P}^i_{K^i}$ is initiated: the positions of the particles (the hyper-parameters of the PFAM classifiers) are randomly initialized in the optimization space, and the classifiers (their internal weights $\mathbf{W}, \mathbf{W}^{ab}, \mathbf{W}^{ac}$ and Σ) are empty. On the other hand, if a gradual change is detected, previously-trained classifiers of pool $\mathcal{P}^i_{k^*}$ are updated through supervised incremental learning: their starting position (hyper-parameters) and internal weights are the final state of the previous optimization, when a similar concept had been encountered and learned.

Finally, in order to adapt to the optimization space according to gradual changes, and pursue the training of the classifiers after a previous optimization, the adaptation and training module is implemented with a dynamic variant of the PSO algorithm. The PSO algorithm has been adapted for dynamic optimization problems though 2 types of mechanisms to: (1) maintain the diversity in the optimization space through a modification of the social influence (such as [44]) and (2) increase the diversity in the optimization space after convergence when a change is detected in the objective function (using the memory of the particles) (such as [6]). For this specific implementation, the DNPSO variant presented in [44] is used. DNPSO maintains diversity within a pool $\mathcal{P}^i_k$ in the optimization space by: (1) relying on a local neighbourhood topology to generate *sub-swarms* of particles around *local bests* (which are the best particles in a local neighbourhood), (2) allowing the evolution of free particles (not in any subswarms) to explore the optimization space independently, and (3), reinitializing the free particles with low velocities. The social source of influence is determined within each sub-swarm. The choice of the DNPSO variant is motivated by the greater exploration of the optimization space through the generation of sub-swarms. This enables to consider all optima during the optimization process, instead of restarting at the convergence area when new reference data used for adaptation that exhibit a gradual change.

4.4. Short and long term memories

The long term memory LTM^i stores the different parameters and models necessary to pursue system training and detect changes when new reference samples become available for an individual *i*. On the other hand, the short term memory STM^i is not memorized from one training session to another, and serves as a temporary storage for reference validation samples.

A specific implementation: For each detected concept $k = 1, \ldots, K^i$, LTM^i stores the following:

1. A pool of 2-class PFAM classifiers $\mathcal{P}^i_k$. The hyper-parameter vector $\mathbf{h}$ as well as the PFAM's internal parameters ($\mathbf{W}, \mathbf{W}^{ab}, \mathbf{W}^{ac}$ and Σ). This pool is evolved and updated using the DNPSO incremental learning strategy (see Section 4.3).

2. An histogram concept representation $\mathcal{C}_k^i$, with the frequency of bins defined by the reference patterns corresponding to the concept.
3. The history of past change detection measures $\mathcal{H}_k^i$, which stores the Hellinger distances computed between the histogram representation of the previously-acquired reference data and the concept k, in order to be able to compute the adaptive change detection threshold.

The data stored in STM^i is used to perform the optimization of the classifiers in the different pools $\{\mathcal{P}_1^i, \ldots, \mathcal{P}_{K^i}^i\}$, and choose the user specific threshold θ^i, for the classifier IC_1^i or the ensemble EoC^i (after *average* score-level fusion), according to false alarm specifications.

4.5. Spatio-temporal recognition – accumulation of responses

As shown in Fig. 1, systems for FRiVS typically rely on face detection tracking and classification. Fig. 4 is an example of a system that combines spatial and temporal computations into separate, but mutually interacting processing streams that cooperate for enhanced detection of individuals of interest. The general track-and-classify strategy has been shown to provide a high level of performance in video-based FR [40]. Since classification and tracking co-occur in parallel, they can collaborate to improve overall face recognition.

During operations, face tracking follows the position and motion of different faces appearing in the scene. The objective of the tracker is to regroup ROIs that belong to a same person, and is defined by a high quality track, in order to provide a robust decision for each track through evidence accumulation.

A specific implementation: Fusion of responses from the ensembles and the tracker is accomplished via evidence accumulation, which emulates the brain process of working memory [4]. For each initiated track j, for each individual i enrolled in the AMCS, and for each consecutive ROI $\mathbf{q}_r$ associated with this track, the dedicated ensemble EoC^i generates a binary decision $d^i(\mathbf{q}_r)$ (true, the individual is recognized, or false). The accumulated response is computed with a moving overlapping window of size V ROIs, following:

$$acc_j^i(r) = \sum_{u=r-V/2}^{r+V/2} d^i(\mathbf{q}_u) \tag{5}$$

Then, the presence of the individual i in the track j can be confirmed if the accumulated response goes over a user-defined threshold Γ^i of a consecutive number of activations.

5. Experimental methodology

The performance of the proposed AMCS is evaluated for the detection of individuals of interest with video captured in person re-identification applications. In particular, experiments focus the impact of employing a change detection mechanism (see Section 4.2) within the AMCS to drive the adaptation of facial models from new reference videos exhibiting various forms of concepts change. The objective of the experimental methodology is to validate our main hypothesis: it is beneficial to incorporate new data from different and abruptly changing concepts with a learn-and-combine strategy than with an incremental one.

5.1. Video-surveillance data

The Carnegie Mellon University Face In Action (FIA) face database [22] is composed by 20-s videos capturing the faces of 221 participants in both indoor and outdoor scenario, each video mimicking a passport checking scenario. Videos have been captured with 6 Dragonfly Sony ICX424 cameras at a distance of 0.83 m from the subjects, mounted on carts at three different horizontal angles (0° and $\pm 72.6^\circ$), and with two different focal length (4 and 8 mm) for each. Cameras have a VGA resolution of 640x480 pixels and capture 30 images per second. Data have been captured in three separate sessions of 20 s, at least one month apart. During the first session, 221 participants were present, 180 of whom returned for the second session, and 153 for the third. Only indoor sequences were considered in this paper.

5.1.1. Pre-processing

To extract the ROIs, segmentation has been performed using the OpenCV v2.0 implementation of the Viola-Jones face and eye detection algorithm [55], and the faces have been rotated to align the eyes in order to minimize intra-class variations [24]. Then ROIs have been scaled to a common size of 70 × 70 pixels. Examples of ROIs captured for two individuals are shown in Fig. 6. Features have finally been extracted from ROIs with the Multi-Bloc Local Binary Pattern (LBP) [1] algorithm

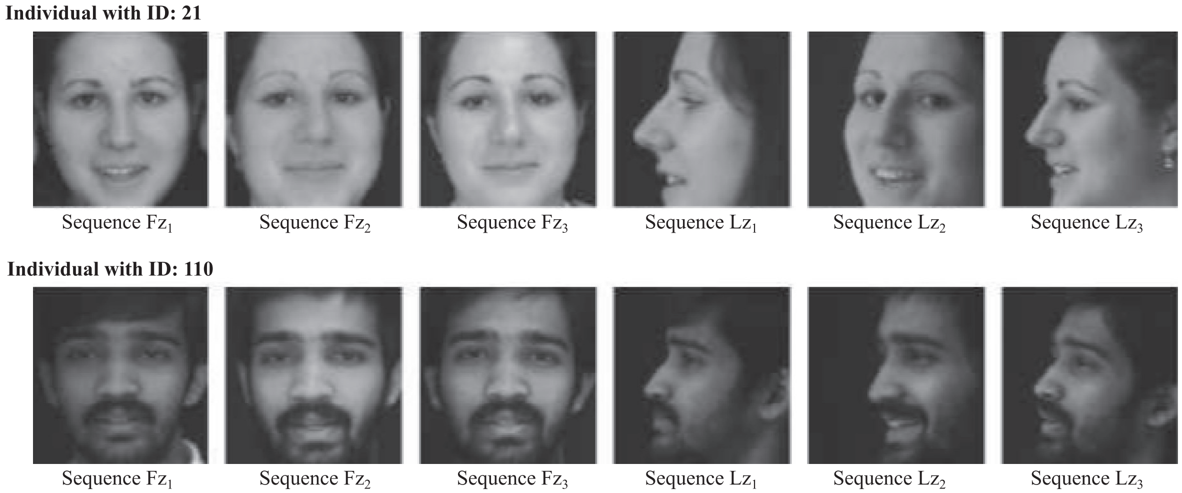


Fig. 6. Examples of ROIs captured by the segmentation algorithm from the cameras array of 6 during the different sessions, for individuals with ID 21 and 110.

for block sizes of 3 × 3, 5 × 5 and 9 × 9 pixels, concatenated with the grayscale pixel intensity values, and reduced to ROI patterns of $D = 32$ features using Principal Component Analysis.

For each one of the 3 sessions, and for each individual, the FIA dataset have been separated into 6 video subsets, according to the different cameras (left, right and frontal view, with 2 different focal length, 4 and 8 mm), resulting into the following sequences with notation:

- F_1 (F_2, F_3), L_1 (L_2, L_3) and R_1 (R_2, R_3): respectively the sequences composed by the samples from the Frontal, Left (-72.6°) and Right (72.6°) view of Session 1 $(2, 3)$, with a 4-mm focal length.
- Fz_1 (Fz_2, Fz_3), Lz_1 (Lz_2, Lz_3) and Rz_1 (Rz_2, Rz_3): the sequences composed by the same samples from the cameras with zoom, 8-mm focal length.

The average number of detected ROIs per individual is presented in Table 3. It can be noted that there are fewer ROIs for the right orientation than for other poses. This can be explained by the fact that the OpenCV Viola & Jones algorithm has only been trained for frontal and left orientations. Therefore, sequences for the right facial orientation subset are not considered for experimental evaluation.

The individuals of interests have been selected among individuals appearing in all 3 sessions, as those with at least 30 ROIs for every frontal and left sequences. Of those, 10 individuals fulfil this requirement, individuals with IDs: 2, 21, 69, 72, 110, 147, 179, 190, 198 and 201. The remaining samples are mixed and separated into two Universal Model (UM) subsets: one half are used to generate the training UM, while the remaining consists in unknown UM classes appearing in test.

5.1.2. Simulation scenario

The following scenario is proposed to simulate video-to-video FR as seen in person re-identification applications.

Design and update of the face models: To simulate the role of the FRiVS operator providing the system with new reference sequences over time to update its facial models, the reference sequences of ROI patterns $Vs[t]$ are presented, after pre-processing, for every discrete time step $t = 1, 2, \ldots, 9$. To avoid a possible bias due to the more numerous ROI detected from the frontal sessions, the original *FIA* frontal sequences have been separated into two sub-sequences, forming a total of 9 sequences, presented in Table 4.

Table 3
Average number of ROI captured per person over 3 indoor sessions (s = 1, 2, 3) of the FIA database.

Orientations	ROIs per camera					
	F_s	Fz_s	R_s	RZ_s	L_s	Lz_s
Session 1	81 ± 4	131 ± 5	11 ± 1	23 ± 2	33 ± 2	40 ± 2
Session 2	88 ± 5	143 ± 7	11 ± 1	20 ± 2	34 ± 3	36 ± 3
Session 3	85 ± 6	141 ± 9	10 ± 1	20 ± 2	42 ± 4	39 ± 3

Table 4
Correspondence between the 9 reference video sequences used to adapt proposed AMCSs and the original *FIA* video sequences.

Time step t	1	2	3	4	5	6	7	8	9
Reference sequences	$Vs[1]$	$Vs[2]$	$Vs[3]$	$Vs[4]$	$Vs[5]$	$Vs[6]$	$Vs[7]$	$Vs[8]$	$Vs[9]$
Corresponding FIA sequence	Fz_1 (S_1)		Fz_2 (S_2)		Fz_3 (S_3)		Lz_1 (S_1)	Lz_2 (S_2)	Lz_3 (S_3)

Video sequences used for design are populated using the samples from the cameras with 8-mm focal length (sequences $Fz_1, Fz_2, Fz_3, Lz_1, Lz_2$ and Lz_3) in order to provide better face capture quality for learning samples. ROIs captured during 3 different sessions and orientations may be sampled from different concepts. The transition from sequence 6 to 7 represents most abrupt concept change in the reference samples, as it involves a change of camera angle. Changes observed from one session to another, such as from sequences 2 to 3, 4 to 5, 7 to 8 and 8 to 9 depends on the individual. As faces are captured over intervals of several months, some abrupt changes can be detected, such as changes in hairstyle, make-up or facial hair. Finally, intra-session changes, from sequences 1 to 2, 3 to 4 and 5 to 6 represent more gradual changes since all sequences were captured with frontal cameras from the same sessions.

Operational evaluation: In order to present different facial captures that the one used for adaptation, only the cameras with 4-mm focal length (sequences $F_1, F_2, F_3, L_1, L_2, L_3$) are considered for operational evaluation. While the scaling normalizes every facial capture to a same size, the short focal length adds additional noise (lower quality ROIs), thus accounting for reference samples that do not necessarily originate from the observation environment in a real-life surveillance scenario.

For each time step $t = 1, 2, \ldots, 9$, the systems are evaluated after adaptation, simulating the arrival of different individuals one by one, at a security checkpoint at the airport. For each of the 3 sessions and 2 considered camera angles, they are presented with the ROI patterns of the corresponding sequences for each individual, one after the other. Evaluation is performed with input data from every session and camera angle for every time step. This simulates a FRiVS scenario where different concepts may be observed during operations, but where the reference videos are not available at the same time. Instead, they are gradually tagged and submitted to the system for adaptation. Every possible concept (face orientation, facial expression, illumination condition, etc.) present in the operational data, is presented to the systems over time.

5.2. Reference systems

For validation of the proposed *AMCS* with change detection (called $AMCS_{CD}$), its performance is compared to the following systems that do not exploit change detection:

- **Incremental AMCS**, $AMCS_{incr}$: Instead of detecting a changes in concepts as $AMCS_{CD}$, a unique concept is considered. This simulates an $AMCS_{CD}$ which never detects any changes, and systematically adapts one single classifier. The system is only composed by a single classifier per individual of interest, and its parameters are updated incrementally when new reference sequences become available. This approach is an implementation of the adaptive classification system presented in [13].
- **Learn and combine AMCS**, $AMCS_{LC}$: This system does not include a change detection mechanism either, simulating an $AMCS_{CD}$ which always detect a change. For every new reference sequence, it systematically triggers the generation of a new concept in the system. It is composed of an ensemble of classifiers per individual, each classifier designed with a different reference sequence.

The comparison between $AMCS_{CD}$ and these two variants enable to evaluate the benefits of using change detection to govern the adaptation strategy. In addition, the proposed $AMCS_{CD}$ is compared the reference open-set TCM-*k*NN [37] presented in Section 2.2. As the TCM-*k*NN is a global (non class-modular) classifier, $AMCS_{CD}$ is also compared to a reference class-modular system using probabilistic class-modular *k*-NN classifier, adapted to the FRiVS application, VS*k*NN. A separate *k*-NN classifier using Euclidean distance is considered for each individual of interest *i*, trained using positive reference samples from video sequences of target individual *i*, and a mixture of negative reference samples from the UM and CM, as with the other *AMCS*. A score is then computed through the *probabilistic kNN* approach [28]: the probability of the presence of the individual *i* is the proportion, among the *k* nearest neighbours, of reference samples from the same individual. The value of *k* is also validated through (2 × 5 folds) cross validation, along with the final decision threshold θ^i.

To improve the scalar performance of the proposed $AMCS_{CD}$ for the selected operating point in validation, a variant called $AMCS_w$ is also tested, where fusion of ensembles is performed at score level. It uses a weighted average to favour scores that are highest w.r.t. their threshold, and filter out possible ambiguities. For an individual *i* with a concept-specific threshold θ^i_k (determined with validation data for concept *k*), and for each score $s^i_k(\mathbf{q})$, the weight ω^i_k is defined by the confidence measure $\omega^i_k = \max(0, (s^i_k(\mathbf{q}) - \theta^i_k)$. This weight reflects the quality of the input pattern $\mathbf{q}$ in reference to concept *k*. The output score is then the result of the weighted average $\sum_{k=1}^{K^i} \omega^i_k \cdot s^i_k$.

5.3. Experimental protocol

Algorithm 4. Experimental protocol for performance evaluation.

```
1  for Each time step, $t \leftarrow 1$ to 9 do
2      for Each individual, $i \leftarrow 1$ to 10 do
3          - Perform change detection using $A^i[t]$ to the stored concept representations $\{C_1^i, \dots, C_{K^i}^i\}$ to determine
           the closest concept index $k^*$. If a new concept is detected, following Alg. 3, $k^* = K^i + 1$;
4          - Generate $dbLearn^i$ dataset. Positive or target samples are selected from the pattern reference sequence
           $A^i[t]$, and relevant negative or non-target samples from $UM^i$ and $CM^i$ (from sequences corresponding to
           the same time stamp) are selected with the CNN method [28] ;
5          for each independent replication, $rep \leftarrow 1$ to 10 do
6              if $rep = 5$ then
7                  - Randomize samples order in $dbLearn^i$;
8              - Separate $dbLearn^i$ into $dbTrain^i$, $dbVal_{ep}^i$, $STM^i$;
9              - Randomly separate $STM^i$ into $dbValPSO_1^i$ and $dbValPSO_2^i$;
10             - Adapt the PFAM network pool corresponding to concept $k^*$, $\mathcal{P}_{k^*}^i$, with the DNPSO training
               strategy, with $dbVal_{ep}^i$ for the stopping criterion of the training epochs of the PFAM classifiers,
               $dbValPSO_1^i$ to compute particles' fitness and $dbValPSO_2^i$ to select the global best classifier $IC_{k^*}^i$;
11             - Assemble $EoC^i = \{IC_1^i, \dots, IC_{K^i}^i\}$ with the updated (or new) $IC_{k^*}^i$;
12             - Select the threshold $\theta^i$ (operating point) corresponding to a $far$ of 5% in validation from the ROC
               curve produced by $EoC^i$ presented with data from $STM^i$;
13             for each operational pattern sequence do
14                 - Computation of the independent, frame by frame, decisions (transnational analysis);
15                 - Accumulation of the decisions of the sequence (time analysis);
```

For each system, simulations follow a (2×5 fold) cross-validation process for 10 independent replications for each experiment, with pattern order randomization at the 5th replication. The full protocol is presented in Algorithm 4. For each time step $t = 1, \dots, 9$, and each individual $i = 1, \dots, I$, the design or update of the system is first performed. Change is first detected (Algorithm 4 line 3), in order to determine the index of the concept k^* closest to the patterns in $A^i[t]$. In the case of $AMCS_{incr}$ ($AMCS_{LC}$), change detection is bypassed, and k^* is automatically set to 1 ($K^i + 1$). Dataset $dbLearn^i$ is then generated (Algorithm 4 line 4), it is used to perform training and optimization of the PFAM networks. It remains unchanged for the two sets of five replications for the results to remain comparable, and is composed of reference patterns from $A^i[t]$, as well as twice the same amount of non target patterns equally selected from the UM dataset and CM^i. More precisely, selection of non-target patterns is achieved using the Condensed Nearest Neighbor (CNN) algorithm [27]. The same amount of target and non-target patterns is selected using CNN, and combined with the same amount (picked at random) of patterns not selected by the algorithm. This enables to select non-target patterns that are close to the decision boundaries, as well as patterns that represent the center of mass of the non-target population. For each independent replication $rep = 1, \dots, 10$, $dbLearn^i$ is divided into the following subsets (Algorithm 4 line 8), based on the 2×5 cross-validation methodology:

- $dbTrain^i$ (2 folds): the training dataset used to design and update the parameters of PFAM networks,
- $dbVal_{ep}^i$ (1-fold): the first validation dataset, used to validate the number of PFAM training epochs (the amount of presentations of patterns from $dbTrain^i$ to the networks) during the PSO optimization,
- STM^i (2 folds): the second validation dataset.

STM^i is randomly divided into two PSO validation datasets $dbValPSO_1^i$ and $dbValPSO_2^i$ (Algorithm 4 line 9). Then, the pool of classifiers $\mathcal{P}_{k^*}^i$ corresponding to the selected concept k^* are trained through the DNPSO learning strategy [13] (Algorithm 4 line 10), using the following parameters: 60 particles per swarm; max of 30 iterations; neighborhoods of 6 particles; max of 40 subswarms; max of 5 particles per sub-swam; early stopping if the best solution ever encountered remains fixed for 5 iterations. The bounds for PFAM parameters during optimization are: $0 \leqslant \bar{\rho} < 1$; $0 \leqslant \alpha \leqslant 1$; $0 \leqslant \beta \leqslant 1$; $-1 \leqslant \varepsilon \leqslant 1$; $0.0001 \leqslant r \leqslant 200$. The fitness computation follows three steps: (1) the training dataset $dbTrain^i$ is presented to the PFAM network, and its performance is evaluated with $dbVal1^i$ – to avoid over-training, this step is repeated for several epochs until the performance converges or decreases, and the stopping criterion is that performance does not increase for two consecutive epochs, (2) the fitness function is evaluated using $dbValPSO_1^i$, and (3) the best particles are determined using the second validation dataset, $dbValPSO_2^i$, and are stored in a archive for each iteration of the optimization. This methodology has been proposed in [16] in order to overcome over-fitting through the selection of particles with the best generalization performance.

When an previously-learned concept is updated, an existing pool $\mathcal{P}^i_{k^*}$ is be evolved through this DNPSO incremental-learning strategy. The optimization resumes from the last state – each classifier of the pool keeps its previous state (network parameters), and incrementally learns the new data. On the other hand, when a significant change is detected, the proposed $AMCS_{CD}$ generates a new pool that is optimized for the new concept $C^i_{K^i}$. The classifiers from each concept are then combined into $EoC^i = \{IC^i_1, \ldots, IC^i_{K^i}\}$, and a validation ROC curve is generated, characterizing the performance of EoC^i over all the samples of STM^i (Algorithm 4 lines 11 and 12). The threshold θ^i corresponding to a $fpr \leqslant 5\%$ is stored for the evaluation of the operational performances of the system. Finally, patterns from the operational sequences (sequences from $F_1, F_2, F_3, L_1, L_2, L_3$, and for every individual in the dataset, see Section 5.1.2) are presented to the systems one sequence at a time. Individual predictions are generated to evaluate the transaction (ROI match) level performance of EoC^i (Algorithm 4 line 14). Then, EoC^i predictions are accumulated over time according to a trajectory of individuals appearing in a scene. This time analysis allows to evaluate the complete system performance (Algorithm 4 line 15).

5.4. Performance measures

Transaction-level performance: Given the responses of a detector (or the final decision of an EoC) for a set of test samples, the true positive rate (*tpr*) is the proportion of positives correctly classified over the total number of positive samples. The false positive rate (*fpr*) is the proportion of negatives incorrectly classified (as positives) over the total number of negative samples. A ROC curve is a parametric curve in which the *tpr* is plotted against the *fpr*. In practice, an empirical ROC curve is obtained by connecting the observed (tpr, fpr) pairs of a soft detector at each threshold. The area under the ROC curve (AUC) or the partial AUC (for a range of *fpr* values) has been largely suggested as a robust scalar summary of 1- or 2-class classification performance. The AUC assesses ranking in terms of class separation – the fraction of positive–negative pairs that are ranked correctly. For instance, with an $AUC = 1$, all positives are ranked higher than negatives indicating a perfect discrimination between classes. A random classifier has an $AUC = 0.5$, and both classes are ranked at random. To focus on a specific part of the ROC curve, the partial AUC *pAUC* can also be computed, as the partial area for a *fpr* less or equal to a specified value.

In a video-surveillance application, non-target individuals are often much greater than the target ones. ROC measure may be inadequate as it becomes biased towards the negative class [56]. For this reason, the precision-recall space has been proposed to remain sensitive to this bias. Indeed, the precision is defined as the ratio $TP/(TP + FP)$ (with TP and FP the number of true and false positives), and the recall is an another denomination of the *tpr*. Precision allows to assess the accuracy for target patterns. The precision and recall measures can be summarized by the F_1 scalar measure, which can be interpreted as the harmonic mean of precision and recall. Finally, a classifier can also be characterized by its *precision-recall* operating characteristics (P-ROC) curve, and the area under the P-ROC curve (*AUPROC*) can be considered as a robust performance measure.

Therefore, considering each ROI match independently, the systems' transaction-level performance will be assessed using:

- Local measures: $tpr, fpr, precision$ and F_1. Those measures are specific to the operating point (threshold Θ^i), determined during system design.
- Global measures: $AUC, pAUC$ and *AUPROC*. Those measures are a more general evaluation of the systems performance over the entire range of the possible operating points.

Performance of the full system over time: To evaluate the performance of the entire system proposed in this paper, individual-specific predictions of each ensemble are accumulated over a trajectory for robust decisions. More precisely, for each individual, the predictions are accumulated with a moving window of $V = 30$ ROIs in a trajectory. The individual is detected when the accumulated activations go past a defined threshold Γ^i.

An example is presented in Fig. 7(a), where 3 sequences of 600 frames have been concatenated. The first and the last one correspond to unknown individuals in the UM, while the second one correspond to target individual 21. The predictions are generated by $AMCS_{CD}$ after the 9th training sequence (session Lz_3), dedicated to the individual 21. It can be observed that genuine predictions go significantly higher than the impostor's.

To assess the overall performance of the different systems for every individual i, an overall accumulation ROC curve is generated, with threshold Γ^i going from 0 to 30 (the size of the moving window). For each target sequence, a true positive occurs when the maximum value of the accumulated predictions goes over Γ^i. In the same way, a false positive occurs when the maximum value of the accumulated predictions for non-target sequences goes over the threshold. An example is presented in Fig. 7(b). To summarize the system performances, the AUC of the overall accumulated ROC curves is used as with the transaction-level measures.

5.5. Memory complexity measures

The systems complexity is evaluated in operational mode, in order to compare resources required to predict the identity associated to an input ROI pattern.

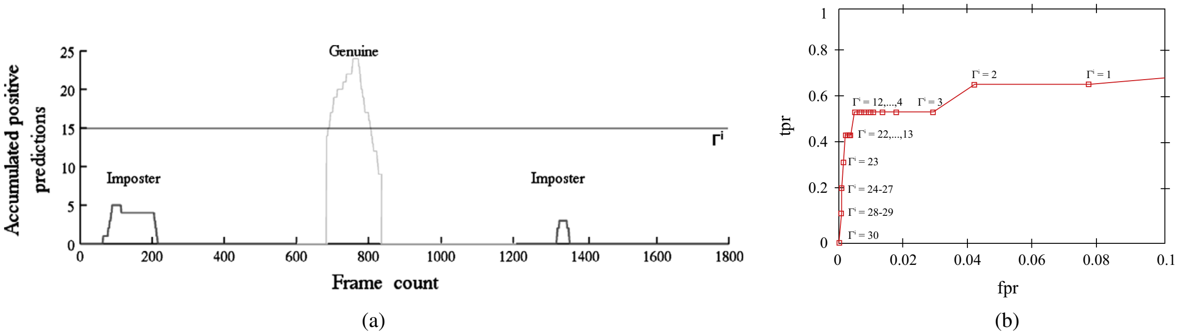


Fig. 7. As an example, assume that individual 21 is enrolled to the $AMCS_{CD}$. After training sequence 9 (*Lz3*), the number of positive predictions accumulated over a fixed-size time window in presented in (a). Three sequences of 600 frames, from 3 different individuals, have been concatenated, with first a sequence from an impostor (in black), then from the genuine individual (21, in gray), and then from another impostor (in black). In (b), the overall accumulation ROC curve characterizing the $AMCS_{CD}$ performance for individual 21 over all the test sequences.

As mentioned in Section 4.1, a PFAM network operational behaviour is one of a GMM, where cluster centres are the prototypes in the *F*2 layer. For each input ROI pattern, the final score is computed from the likelihoods of the different clusters. As a consequence, the memory and time complexity required to classify a facial ROI in operations is proportional to the number of prototypes in PFAM networks. For this reason, the operational memory complexity of *AMCS* systems will be compared based on the sum of the number of *F*2 layer neurons for all the PFAM classifiers in the ensembles.

Similarly, TCM-*k*NN and VS*k*NN both rely on a *k*NN classifier. For each input ROI pattern, an euclidean distance is computed for each reference pattern stored for *k*NN classifier. In VS*k*NN, those distances are then ordered to compute probabilistic scores as presented in Section 5.2. TCM-*k*NN adds more computational complexity, as the score computation relies on strangeness measures for each input ROI, requiring additional re-orderings. The operational memory complexity of the identity prediction from an input ROI is thus also proportional to the number of reference patterns stored in the system, which will be used for comparison.

6. Results and discussion

6.1. Change detection performance

For each individual of interest, Table 5 presents the update sequences for which changes have been detected, as well as the total number of detections. The first sequence corresponds to the initialization of the first concepts of each individual. The maximum number of detection for a sequence of 10, meaning that a change is detected for every individual. The 3 highest detection counts occur for the sequences 3, 5 and 7, and for 6, 8 and 8 of the individuals, respectively. These changes correspond to the introduction of training samples from the 2nd frontal session, the 3rd, and the 1st left session. Although the apparition of changes depends on the specific individuals (haircut change, hat, glasses, etc.), this result is expected since those 3 sessions are the most likely to exhibit significant abrupt changes: the two former occurred at least 2 and 3 months after the first update sequence, and the latter is the first introduction of samples captured from a different angle.

For a more detailed analysis, individuals 21 and 110 were considered. Changes detected in these cases can be correlated with ROIs shown in Fig. 6. For the individual 21, abrupt changes have been detected for update sequences 5 (introduction of patterns Fz_3), 7 (Lz_1) and 9 (Lz_3). As shown in Fig. 6, the changes detected with the introduction of sequences 5 and 9

Table 5
Changes detected per individual of interest (marked as a X) for each update sequence.

Individual ID	Update sequences (time step *t*)									Total per individual
	1	2	3	4	5	6	7	8	9	
2	X				X			X		3
21	X				X		X		X	4
69	X		X			X	X			4
72	X		X				X			3
110	X		X		X		X			4
147	X		X		X		X			4
179	X		X		X			X		4
190	X				X		X			3
198	X				X		X			3
201	X		X		X		X		X	5
Total per sequence	10	0	6	0	8	1	8	2	2	

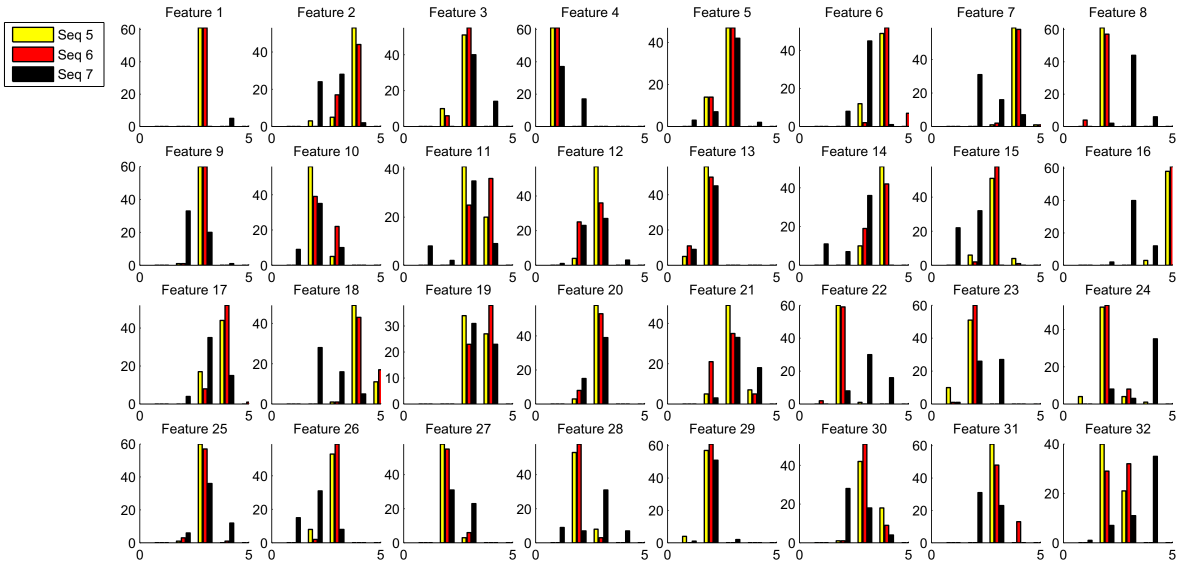


Fig. 8. Histograms representation of the 5th, 6th and 7th sequence of patterns for the individual 21, in the feature space of input ROI patterns ($D = 32$ dimensions). The Hellinger distances between the sequence 5 and 6, and between 6 and 7 are respectively 0.0253 and 0.1119.

correspond to a change in make-up and hair style, while the change detected with sequence 7 is the introduction of left oriented samples. Similarly, as shown in Fig. 6, changes for individual 110 have been detected with sequence 3 (Fz_2), corresponding to hair-style and skin tone change, 5 (Fz_3), also corresponding to skin tone change and 7 (Lz_1), which is the introduction of the samples with left camera angle.

The abrupt change detected with sequence 7 can also be observed in the feature space, as illustrated by the significant differences in histogram representations of the sequences 5, 6 and 7 (see Fig. 8). Sequence 7 is visibly different for most features from sequences 5 and 6 (which belong to the same enrolment session, Fz_3). This difference is also shown in the Hellinger distance between the sequences 6 and 7, which is significantly higher than between the sequences 5 and 6.

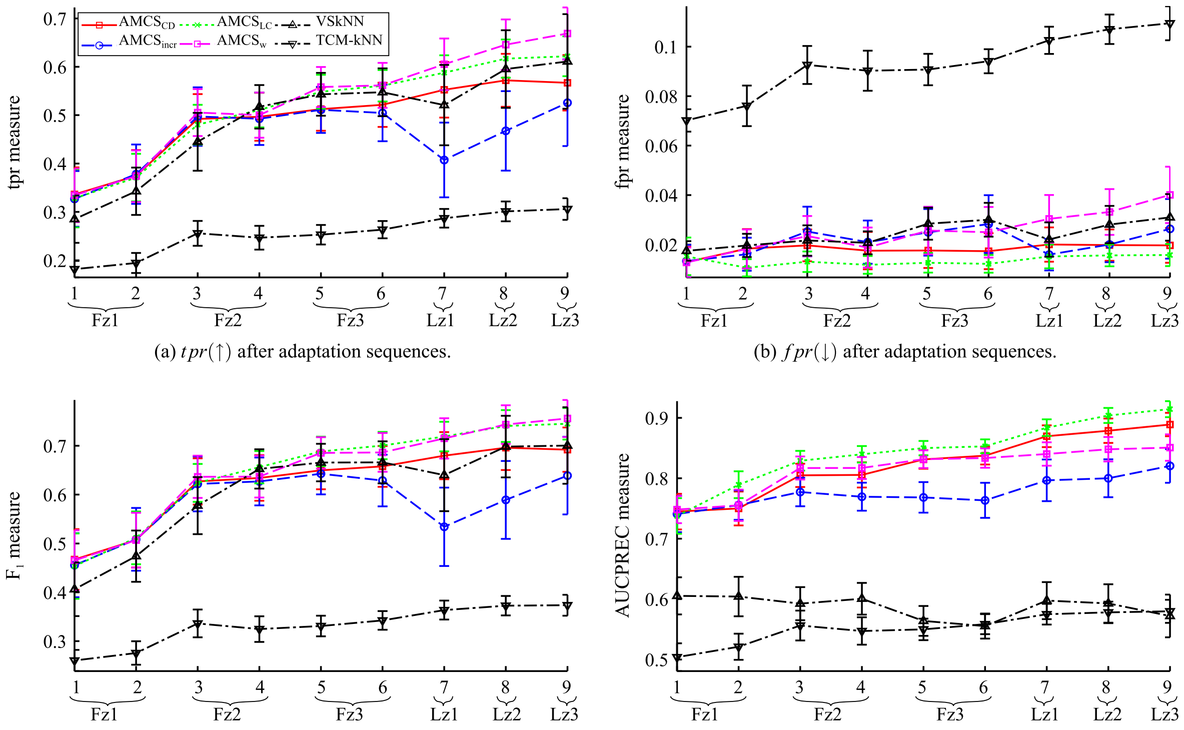
Results confirm that the change detection module proposed for $AMCS_{CD}$ (Fig. 4) can efficiently detect abrupt concept changes in sequences of facial captures of the *FIA* dataset. In response to new reference video sequences, this module allows $AMCS_{CD}$ to adapt facial models according to different strategies, either incremental learning or *learn-and-combine*.

6.2. Transaction-level performance

Average results: Fig. 9 presents the average overall transaction-level performance of proposed and reference systems, for the 10 individuals of interest according to fpr, tpr and F_1 measures (Fig. 9(a)–(c)) at an operating point selected (during validation) to respect the constraint $fpr \leqslant 5\%$, and the global *AUPROC* measure over all *fpr* values (Fig. 9(d)). Performance is assessed on predictions for each ROI captured in test sequences, after the systems are updated on each adaptation sequence.

In Fig. 9(d), $AMCS_{CD}$, $AMCS_{incr}$ and $AMCS_{LC}$ exhibit a significantly higher level of *AUPROC* performance than VS*k*NN and TCM-*k*NN. After learning the 9th update sequence, VS*k*NN and TCM-*k*NN have an average *AUPROC* of 0.57 ± 0.04, while $AMCS_{incr}, AMCS_{CD}$ and $AMCS_{LC}$ are respectively at $0.82 \pm 0.03, 0.89 \pm 0.02$ and 0.91 ± 0.01. Performing a *Kruskal–Wallis* test for those three measures using a *p-value* of 0.1, indicates that $AMCS_{incr}$ performance is significantly lower than $AMCS_{CD}$ and $AMCS_{LC}$, which are comparable. In addition, while these 3 *AMCS* yield similar performance after the first 2 sequences, $AMCS_{CD}$ and $AMCS_{LC}$ improve their performance more significantly than $AMCS_{incr}$ when samples from session Fz_2 are integrated into the systems. Average *AUPROC* performance goes from 0.75 ± 0.03 and 0.79 ± 0.02 to 0.81 ± 0.02 and 0.83 ± 0.02 for $AMCS_{CD}$ and $AMCS_{LC}$, while it only goes from 0.76 ± 0.02 to 0.78 ± 0.02 for $AMCS_{incr}$. Sequence Fz_3 represents the most abrupt changes for the frontal faces, captured several months later, along *left* pose, Lz_1, Lz_2 and Lz_3. The $AMCS_{CD}$ and $AMCS_{LC}$ benefit the most from learning this new data as their *AUPROC* performance continues to diverge w.r.t. that of $AMCS_{incr}$ until the last update sequence.

In terms *fpr* performance (Fig. 9(b)) it can be first observed that all systems except TCM-*k*NN remain under the constraint of $fpr \leqslant 5\%$, with $AMCS_{LC}$ and $AMCS_{CD}$ significantly lower than $AMCS_{incr}$ and VS*k*NN. It can be noted that $AMCS_{CD}$ provides a lower *fpr* on test sequences. After learning update sequence 9, the average *fpr* for $AMCS_{CD}$ is at $1.97\% \pm 0.70$. On the other hand, the *fpr* of the $AMCS_w$ variant is more affected by the introduction of the new orientation after sequence 7, at it increases, after sequence 9, to $4.0\% \pm 1.13$. A closer reveals that false positives are mainly a consequence of an increase of score values for negative samples for one of the classifiers in each ensemble. In most of the cases, when a change is detected and the *fpr* increases, most of the false positives are triggered by classifiers that correspond to newly-added concepts, not specialized to differentiate positive from negative samples of the other concepts. This produces a positive



(a) $tpr(\uparrow)$ after adaptation sequences.

(b) $fpr(\downarrow)$ after adaptation sequences.

(c) $F_1(\uparrow)$ after adaptation sequences.

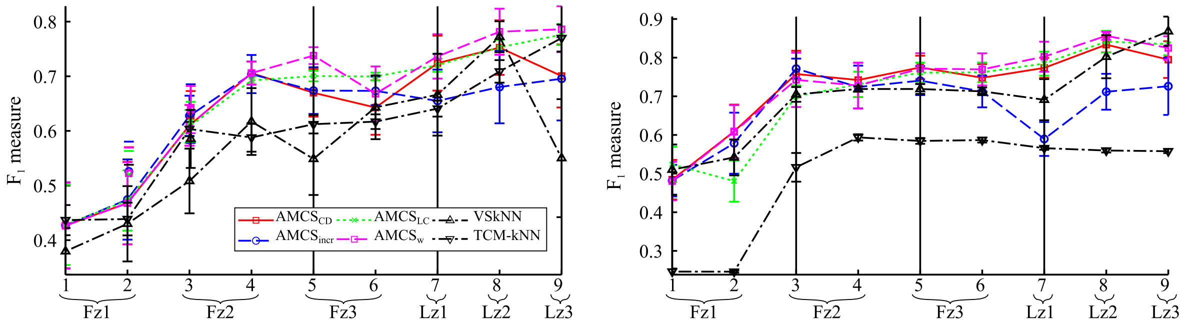
(d) $AUPROC(\uparrow)$ after adaptation sequences.

Fig. 9. Average overall transaction-level performance of proposed and reference systems, after the integration of the 9 adaptation sequences. The average value of performance measures and confidence interval over 10 replications are averaged for the 10 individuals of interest.

prediction for a non-target ROI from a different concept than the one of its training patterns. An increase of *fpr* can indeed be observed after learning from Lz_1 as the majority of changes (and thus the classifier addition) are detected at those transitions. Although always below the 5% constraint imposed in validation, $AMCS_w$ has the tendency to increase the false positives of different ensembles, as those scores are increased by the normalizations which set to zero other lower scores.

The F_1 measure (Fig. 9(c)) gives a condensed view of the precision and recall (*tpr*) for the selected operating point, and it allows to observe the performance on target samples. The F_1 performances all systems except TCM-*k*NN are comparable until the update sequence 5. Updating on reference sequences from session Fz_3 enables $AMCS_w$ and $AMCS_{LC}$ to differentiate themselves, at respectively 0.69 ± 0.03 and 0.70 ± 0.03. However, the most significant decline in F_1 performance occurs for update sequence 7 (Lz_1 sessions), where $AMCS_{incr}$ performance decreases from 0.63 ± 0.05 to 0.53 ± 0.08, and the system requires two more sequences of *left* oriented captures to recover. The decrease of the F_1 performance is a consequence of a decrease in *tpr*, from $50.46\% \pm 5.9$ to $40.73\% \pm 7.7$ after learning sequence 7. This is a manifestation of the knowledge corruption that can occur in an incremental system, as the introduction of significantly different training patterns decreased its ability to effectively detect positive ones. $AMCS_{CD}, AMCS_w, AMCS_{LC}$ and VS*k*NN, on the other hand, do not suffer from the same effects, and their performance continues to improve over the last 3 update sequences. After learning the 9th update sequence, $AMCS_w$ and $AMCS_{LC}$ exhibit the highest level of F_1 performance at respectively 0.76 ± 0.04 and 0.75 ± 0.03. $AMCS_{CD}$ and VS*k*NN both end at about 0.70. The *tpr* boost induced by the fusion function of $AMCS_w$ provides the best F_1 performance, despite the higher *fpr* values. Finally, the performance of TCM-*k*NN remains significantly lower throughout the experiments.

Focus on individuals 21 and 110: Fig. 10 presents average transaction-level F_1 performance obtained for individuals 21 and 110. They provide a bad (individual 21) and a good (individual 110) case for the proposed $AMCS_{CD}$. The F_1 performance of individual 110 leads to similar observations as with the average overall evaluations: the performance declines in F_1 of $AMCS_{inc}$ at the 7th sequence while $AMCS_{CD}$ and $AMCS_{LC}$ continue to improve, and TCM-*k*NN performance remains below all the others. However, for individual 21, while the 7th sequence triggers a change detection, all systems exhibit similar performances and behaviour, without any decline in F_1 for $AMCS_{incr}$. With a closer examination of the videos, about 92% of the ROIs in the Lz_1, Lz_2 and Lz_3 sequence for individual 110 are profile orientation, while the remainder are mostly 3/4 frontal views captured during a movement of the individual's head. In contrast only 51% of the ROIs of individual 21 correspond to a profile orientation, with a majority in the Lz_3 session (9th sequence), at the end of the simulation. Individual 21 can be considered as a case where the change detection process may be too sensitive - the new reference patterns provided in the sequences 7, 8 and 9 are not different enough for new classifiers to have a considerable impact on transaction-level



(a) Individual 21, F_1 after each adaptation sequence.

(b) Individual 110, F_1 after each adaptation sequence.

Fig. 10. Average transaction-level performance after learning the 9 update sequences. Significant (abrupt) changes are indicated as vertical lines.

performance. Results for individual 110 shows that learning significantly diversified samples can be more effective with the proposed $AMCS_{CD}$. Finally, in both cases, $AMCS_w$ still exhibits similar F_1 performance to $AMCS_{LC}$.

Although the $AMCS_{CD}$ and VS*k*NN exhibits similar transaction-level performance, Fig. 11 shows that the amount of prototypes (sum of the number of *F*2 layer neurons for all the PFAM classifiers in an ensemble) needed by the 3 *AMCS* is significantly lower than the number of reference patterns needed by VS*k*NN and TCM-*k*NN. The memory complexity of VS*k*NN and TCM-*k*NN grows to about 900 prototypes after the 9 adaptation sequences. The complexity of $AMCS_{CD}$ ($AMCS_w$), $AMCS_{incr}$ and $AMCS_{LC}$ remain comparable until the update sequence 5. Their sizes continue to grow until the last sequence, with $AMCS_{incr}$ the smaller system (200.84 ± 28.2), and $AMCS_{LC}$ the bigger one (322 ± 16.8). $AMCS_{CD}$ ends with 250 ± 13.7 prototypes. Considering that a prototype or reference sample weights 128 bytes (a vector of 32 *floats* of 32 bits), the reference sample stored by VS*k*NN and TCM-*k*NN after the 9 adaptation sequences use up to 115 kB, while the prototypes of $AMCS_{CD}$ ($AMCS_w$), $AMCS_{incr}$ and $AMCS_{LC}$ respectively use around 32, 25.6 and 42.2 kB.

Overall, the proposed $AMCS_{CD}$ provides a compromise between the $AMCS_{incr}$ (low complexity but lower performance) and $AMCS_{LC}$ (significantly greater complexity but comparable performance). In this simulation, the $AMCS_{incr}$ exhibits the knowledge corruption problem, while the reference $AMCS_{CD}$ and VS*k*NN are prone to a increase in the system complexity. Those two problems have been presented in Section 3.2 as the main issues of the adaptive classification system in the literature. The proposed $AMCS_{CD}$ can achieve transaction-level performance comparable to the reference $AMCS_{LC}$ and VS*k*NN systems, but with a significantly lower computational complexity. In addition, $AMCS_{CD}$'s performance is significantly higher than the *open-set* TCM-*k*NN. By virtue of the change detection mechanism, it can also avoid the decline in performance due to knowledge corruption (seen with $AMCS_{incr}$) when learning significantly different adaptation sequences. Finally, although exhibiting higher *fpr* (but below the validation constraint) the $AMCS_w$ achieve significantly better performance in terms of *tpr* and similar F_1 than $AMCS_{CD}$, without being negatively affected by the introduction of different adaptation sequences as $AMCS_{CD}$.

6.3. Performance of the full system over time

In the proposed architecture (see Fig. 4), the face tracker groups ROIs corresponding to tacking trajectories initiated in each video sequence. Classification prediction for each ROI in each trajectory are accumulated over time. Considering that the transaction-level performance of the *open-set* TCM-*k*NN was consistently lower than the other systems, and that the system has not originally been designed to be used with an accumulation strategy, TCM-*k*NN's accumulation performance has not been evaluated.

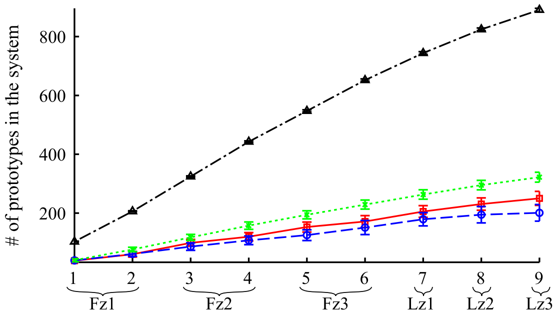


Fig. 11. Average memory complexity. Amount of *F*2 prototypes for the *AMCS* systems, and amount of reference patterns for VS*k*NN and TCM-*k*NN, after learning of adaptation sequences. $AMCS_{CD}$ and $AMCS_w$ have the same amount of prototypes, as well as VS*k*NN and TCM-*k*NN.

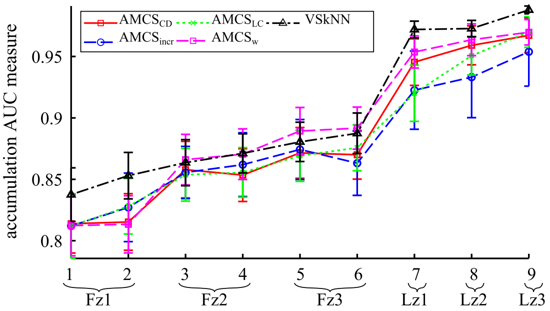
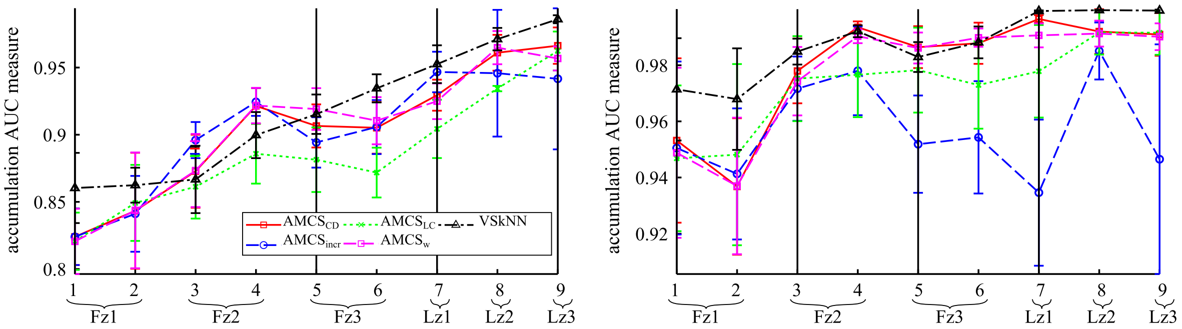


Fig. 12. Average accumulation AUC performance after learning the 9 update sequences.



(a) Individual 21.

(b) Individual 110

Fig. 13. Accumulation AUC performance after learning the 9 update sequences.

Average results: The average accumulation performance are presented in Fig. 12. The accumulation performance of $AMCS_{CD}, AMCS_{LC}$ and $AMCS_{incr}$ are similar from sequence 1 to 6. VS*k*NN provides the best performance level for the two first sequences (0.84 ± 0.02 and 0.85 ± 0.02), and $AMCS_w$ exhibits similar performance to VS*k*NN from sequences 3 to 6. At sequence 6, $AMCS_{LC}, AMCS_w$, VS*k*NN exhibit accumulation performance comparable to $AMCS_{CD}, AMCS_{incr}$ and $AMCS_{LC}$. Then, it can be seen that the accumulation process filters out the irregularities, and their accumulation performance increases after the introduction of sequences from Lz_1. The increase is however more important for VS*k*NN, $AMCS_w$ and $AMCS_{CD}$, which respectively go to $0.97 \pm 0.01, 0.95 \pm 0.01$ and 0.95 ± 0.01. $AMCS_{incr}$ shows less improvement, up to 0.92 ± 0.03, and requires two more sequences (8 and 9) to reach a level comparable to the others. After 9 sequences, VS*k*NN exhibits the better accumulation performance, (0.99 ± 0.003) closely followed by $AMCS_{CD}, AMCS_w$ and $AMCS_{LC}$ (0.97 ± 0.01). $AMCS_{incr}$ exhibits the lowest performance, at 0.95 ± 0.03.

As with the transactional-level results, the proposed $AMCS_{CD}$ is capable of exhibiting similar accumulation performance than VS*k*NN and $AMCS_{LC}$ variant, but with a significantly lower level of complexity, while outperforming the $AMCS_{incr}$ classifier, which requires more data to accommodate to significantly different concepts.

Focus on individuals 21 and 110: The accumulation performances of the five systems for individual 21 and 110 is presented in Fig. 13, and reveals the same observations. With individual 21, which data exhibit less abrupt changes (because only half of the ROIs from Lz_1, Lz_2 and Lz_3 have a profile orientation) all systems perform comparably, as confirmed by a *Kruskall–Wallis* test (with a *p-value* of 0.1). However, for individual 110, the presentation of update sequence 5 (Fz_3 session) decreases $AMCS_{incr}$ accumulation performance from 0.98 ± 0.02 to 0.95 ± 0.02 while the $AMCS_{CD}$ performance remains more stable around 0.99. The significance of this decrease is also confirmed by the *Kruskall–Wallis* test, which confirms that those two system performances are significantly different after the 5th sequence. Similar behaviour can be observed after the presentation of sequence 7 (session Lz_1).

As with the transactional-level analysis, time analysis of the full system reveals the benefits of the proposed change detection strategy. The proposed $AMSC_{CD}$ and $AMCS_w$ are less negatively affected by the introduction of update sequences that incorporate significant concept changes than $AMCS_{incr}$. Yet they achieved comparable performance to VS*k*NN and $AMCS_{LC}$ with a significantly reduced computational complexity.

7. Conclusion

In this paper, a new adaptive multi-classifier system is proposed for video-to-video face recognition in changing environments, as found in person re-identification applications. This modular system is comprised of a classifier ensemble per individual that allows to adapt the facial model of target individuals in response to new reference videos, through either incremental learning or ensemble generation. When a new video trajectory is provided by the operator, a change detection mechanism is used to compromise between plasticity and stability. If the new data incorporates an abrupt pattern of change w.r.t. previously-learned knowledge (representative of a new concept), a new classifier is trained on the data and combined to an ensemble. Otherwise, previously-trained classifiers are incrementally updated. During operations, faces of each different individual are tracked and grouped over time, allowing to accumulate positive predictions for robust spatio-temporal recognition.

A particular implementation of this framework has been proposed for validation. It consists of an ensemble of 2-class Probabilistic Fuzzy-ARTMAP classifiers for each enrolled individual, where each ensemble is generated and evolved using an incremental training strategy based on a dynamic Particle Swarm Optimization, and the Hellinger Drift Detection Method to detect concept changes. Simulation results indicate that the proposed $AMCS_{CD}$ is able to maintain a high level of performance when significantly different reference videos are learned for an individual. It exhibits higher classification performance than a probabilistic *k*NN based system adapted to video-to-video FR, as well as a reference open-set TCM-*k*NN system, with a significantly lower complexity. The scalable architecture employs the change detection mechanism to mitigate the effects of knowledge corruption while bounding its computational complexity.

A key assumption of the adaptive multi-classifier system proposed in this paper is that each trajectory only contains ROI patterns that have been sampled from one concept. In future work, this framework should be extended in order to detect possible sub-concepts in the same trajectory (i.e. changes in facial pose and expression), using for example some windowing strategy. In addition, the particular implementation used for validation has been tested on a large-scaled data set where reference videos have a limited length. Performance should be assessed on other data sets that are representative of person-re-identification (or search and retrieval) applications. Finally, a practical implementation of this framework would require a strategy to purge irrelevant concepts and validation data over time, and bound the system's memory consumption.

Acknowledgements

This work was partially supported by the Natural Sciences and Engineering Research Council of Canada, and the Ministère du Développement Économique, de l'Innovation et de l'Exportation du Québec.

References

[1] T. Ahonen, A. Hadid, M. Pietikainen, Face description with local binary patterns: application to face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 28 (42) (2006) 2037–2041.
[2] C. Alippi, G. Boracchi, M. Roveri, A just-in-time adaptive classification system based on the intersection of confidence intervals rule, Neural Networks 24 (8) (2011) 791–800.
[3] C. Alippi, G. Boracchi, M. Roveri, Just-in-time classifiers for recurrent concepts, IEEE Trans. Neural Networks Learn. Syst. 24 (4) (2013) 620–634.
[4] M. Barry, E. Granger, Face recognition in video using a what-and-where fusion neural network, in: Int. J. Conf. on Neural Networks, 2007, pp. 2255–2260.
[5] S. Bengio, S. Marithoz, Biometric person authentication is a multiple classifier problem, in: Int. Workshop on MCS, Prague, 2007.
[6] T. Blackwell, J. Branke, Multi-swarm optimization in dynamic environments, Appl. Evol. Comput. (2004) 489–500. Coimbra.
[7] A. Blum, Empirical support for winnow and weighted-majority algorithms: results on a calendar scheduling domain, Mach. Learn. 26 (1) (1997) 5–23.
[8] A. Brew, P. Cunningham, Combining cohort and UBM models in open set speaker detection, Multimedia Tools Appl. 48 (2010) 141–159.
[9] 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 Trans. Neural Networks 3 (5) (1992) 698–713.
[10] D. Chakraborty, N.R. Pal, A novel training scheme for MLPs to realize proper generalization and incremental learning, IEEE Trans. Neural Networks 14 (1) (2003) 1–4.
[11] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: a survey, ACM Comput. Surv. 41 (2009).
[12] J.F. Connolly, E. Granger, R. Sabourin, Supervised incremental learning with the fuzzy ARTMAP neural network, in: Artificial Neural Networks in Pattern Recognition, Paris, France, 2008.
[13] J.F. Connolly, E. Granger, R. Sabourin, An adaptive classification system for video-based face recognition, Inform. Sci. 192 (2012) 50–70.
[14] J.F. Connolly, E. Granger, R. Sabourin, Dynamic multi-objective evolution of classifier ensembles for video-based face recognition, Appl. Soft Comput. 13 (6) (2013) 3149–3166.
[15] G. Ditzler, R. Polikar, Hellinger distance based drift detection for nonstationary environments, in: IEEE Sym. on Comp. Intelligence in Dynamic and Uncertain Environments, Paris, April 11–15, 2011, pp. 41–48.
[16] E.M. Dos Santos, R. Sabourin, P. Maupin, Overfitting cautious selection of classifier ensembles with genetic algorithms, Inform. Fusion 10 (2009) 150–162.
[17] A. Dries, U. Ruckert, Adaptive concept drift detection, Stat. Anal. Data Mining 2 (2009) 311–327.
[18] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proc. of the 6th Int. Symp. on Micro Machine and Human Science, 1995, pp. 39–43.
[19] H.K. Ekenel, L. Szasz-Toth, R. Stiefelhagen, Open-set FR-based visitor interface system, LNCS 5815 (2009) 43–52.
[20] Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm, in: Proc. of 13th Int. Conf. on Machine Learning, 3–6 July, 1996, pp. 148–156.
[21] B. Fritzke, Growing self-organizing networks-why? in: Proc.of European Symposium on Artificial Neural Networks, Bochum, 24–26 April, 1996, pp. 61–72.
[22] R. Goh, Lihao Liu, Xiaoming Liu, Tsuhan Chen, The CMU face in action (FIA) database, in: Proc.of Analysis and Modelling of Faces and Gestures, Berlin, 2005, pp. 255–263.
[23] J. Gama, P. Medas, G. Castillo, P. Rodrigues, Learning with drift detection advances in artificial intelligence, in: Proc. of 17th Brazilian Symp.on Artificial Intelligence, 29 September–1 October, 2004, pp. 286–295.
[24] D.O. Gorodnichy, Video-based framework for face recognition in video, in: Proc. of 2nd Canadian Conference on Computer and Robot Vision, Victoria, 9–11 May, 2005, pp. 330–338.
[25] E. Granger, J.-F. Connolly, R. Sabourin, A comparison of fuzzy ARTMAP and Gaussian ARTMAP neural networks for incremental learning, in: Int. J. Conf. on Neural Networks, Hong Kong, 2008, pp. 3305–3312.

[26] S. Grossberg, Nonlinear neural networks: principles, mechanisms, and architectures, Neural Networks 1 (1) (1988) 17–61.

[27] P.E. Hart, The condensed nearest neighbor, IEEE Trans. Inform. Theory IT-14 (1968) 515–516.

[28] C.C. Holmes, N.M. Adams, A probabilistic NN method for statistical pattern recognition, J. Roy. Stat. Soc. Ser. (2002) 295–306.

[29] B. Kamgar-Parsi, W. Lawson, B. Kamgar-Parsi, Toward development of a face recognition system for watchlist surveillance, IEEE Trans. PAMI 33 (10) (2011) 1925–1937.

[30] M.N. Kapp, C.O. Freitas, R. Sabourin, Methodology for the design of NN-based month-word recognizers written on Brazilian bank checks, Image Vis. Comput. 25 (2007) 40–49.

[31] J. Kittler, F.M. Alkoot, Sum versus vote fusion in multiple classifier systems, Trans. Pattern Anal. Mach. Intell. 25 (1) (2003) 110–115.

[32] R. Klinkenberg, I. Renz, Adaptive information filtering: learning in the presence of concept drifts, in: Workshop Learning for Text Categorization, 1998, pp. 33–40.

[33] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley, 2004.

[34] L.I. Kuncheva, Classifier ensembles for changing environments, in: Proc. of the Int. Workshop on MCS, Cagliari, Italy, 2004, pp. 1–15.

[35] L.I. Kuncheva, Classifier ensembles for detecting concept change in streaming data: overview and perspectives, in: 2nd Workshop SUEMA, 2008.

[36] L.I. Kuncheva, Using Control Charts for Detecting Concept Change in Streaming Data, Bangor University, 2009.

[37] F. Li, H. Wechsler, Open-set face recognition using transduction, IEEE Trans. PAMI Intell. 27 (11) (2005) 1686–1697.

[38] C.P. Lim, R.F. Harrison, Probabilistic fuzzy ARTMAP: an autonomous neural network architecture for bayesian probability estimation, in: Proc. of 4th Int. Conf. on Artificial Neural Networks, 1995, pp. 148–153.

[39] Machine Readable Travel Documents. International Civil Aviation Organization, August 2006.

[40] F. Matta, J.-L. Dugelay, Person recognition using facial video information: a state of the art, J. Visual Lang. Comput. 20 (2009) 180–187.

[41] L. Minku, A. White, X. Yao, The impact of diversity on on-line ensemble learning in the presence of concept drift, IEEE Trans. Knowl. Data Eng. 99 (1) (2009).

[42] L.L. Minku, X. Yao, DDD: a new ensemble approach for dealing with concept drift, IEEE Trans. Knowl. Data Eng. 24 (4) (2012).

[43] A. Narasimhamurthy, L.I. Kuncheva, A framework for generating data to simulate changing environments, in: Proc. of the 25th IASTED International Multi-Conf.: Artificial Intelligence and Applications, 2007, pp. 384–389.

[44] A. Nickabadi, M.M. Ebadzadeh, R. Safabakhsh, DNPSO: a dynamic niching particle swarm optimizer for multi-modal optimization, in: 2008 IEEE Congress on Evolutionary Comp., 2008, pp. 26–32.

[45] I.-S. Oh, C.Y. Suen, A class-modular FF NN for handwriting recognition, Pattern Rec. 35 (2002) 229–244.

[46] K. Okamoto, S.Ozawa, S. Abe, A fast incremental learning algorithm with long-term memory, in: IEEE Int. Joint Conf. on Neural Networks, Portland, 2003.

[47] N.C. Oza, Online Ensemble Learning, Technical Report, PhD Thesis, University of California, Berkeley, 2001.

[48] C. Pagano, E. Granger, R. Sabourin, D. Gorodnichy, Detector ensembles for face recognition in video surveillance, in: Int. J. Conf. on Neural Networks, Brisbane, 2012, pp. 1–8.

[49] J.N. Pato, L.I. Millett, Biometric Recognition: Challenges and Opportunities, Whither Biometrics Committee, National Research Council of the NSA, National Academies Press, 2010.

[50] R. Polikar, L. Upda, S.S. Upda, V. Honavar, Learn++: an incremental learning algorithm for supervised neural networks, IEEE Trans. Syst. Man Cybernet. 31 (2001) 497–508.

[51] A. Rattani, Adaptive Biometric Systems based on Template Update Procedures, Dept. of Elect. and Comp. Eng., University of Cagliari, PhD Thesis, 2010.

[52] S. Ruping, Incremental learning with support vector machines, in: IEEE Int. Conf. on Data Mining, San Jose, 2001.

[53] H. Sellahewa, S.A. Jassim, Image-quality-based adaptive face recognition, IEEE Trans. Instrum. Meas. 59 (2010) 805–813.

[54] D.M.J. Tax, R.P.W. Duin, Growing a multi-class classifier with a reject option, Pattern Rec. Lett. 29 (10) (2008) 1565–1570.

[55] P. Viola, M.J. Jones, Robust real-time face detection, Int. J. Comput. Vis. 57 (2004) 137–154.

[56] G. Weiss, The Effect of Small Disjuncts and Class Distribution on Decision Tree Learning, Ph.D. Dissertation, Department of Computer Science, Rutgers University, May 2003.

[57] Z. Xingquan, W. Xindong, Y. Ying, Dynamic classifier selection for effective mining from noisy data streams, in: Proc. IEEE Int'l Conf. on Data Mining, 2004, pp. 305–312.

[58] W. Zhao, R. Chellappa, P.J. Phillips, A. Rosenfeld, Face recognition: a literature survey, ACM Comput. Surv. 35 (4) (2003) 399–458.