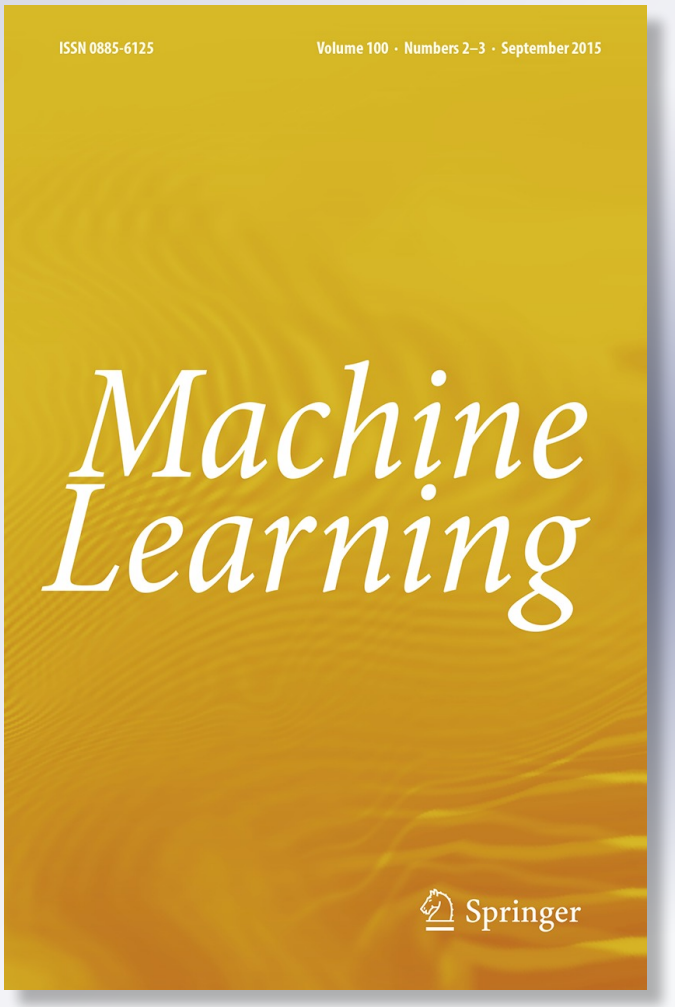
Learning from evolving video streams in a multi-camera scenario

Samaneh Khoshrou, Jaime S. Cardoso & Luís F. Teixeira

Machine Learning

ISSN 0885-6125
Volume 100
Combined 2-3

Mach Learn (2015) 100:609-633
DOI 10.1007/s10994-015-5515-y



Springer

Mach Learn (2015) 100:609–633
DOI 10.1007/s10994-015-5515-y

Learning from evolving video streams in a multi-camera scenario

Samaneh Khoshrou[1] · **Jaime S. Cardoso**[1] · **Luís F. Teixeira**[2]

Received: 30 November 2014 / Accepted: 8 June 2015 / Published online: 15 July 2015

Abstract Nowadays, video surveillance systems are taking the first steps toward automation, in order to ease the burden on human resources as well as to avoid human error. As the underlying data distribution and the number of concepts change over time, the conventional learning algorithms fail to provide reliable solutions for this setting. In this paper, we formalize a learning concept suitable for multi-camera video surveillance and propose a learning methodology adapted to that new paradigm. The proposed framework resorts to the universal background model to robustly learn individual object models from small samples and to more effectively detect novel classes. The individual models are incrementally updated in an ensemble-based approach, with older models being progressively forgotten. The framework is designed to detect and label new concepts automatically. The system is also designed to exploit active learning strategies, in order to interact wisely with operator, requesting assistance in the most ambiguous to classify observations. The experimental results obtained both on real and synthetic data sets verify the usefulness of the proposed approach.

Keywords Video surveillance · Parallel streams · Active learning

Editors: João Gama, Indre Žliobaite, Alípio M. Jorge, and Concha Bielza.

✉ Samaneh Khoshrou
samaneh.khoshrou@inescporto.pt
http://www.inesctec.pt

Jaime S. Cardoso
jaime.cardoso@inescporto.pt
http://www.inesctec.pt

Luís F. Teixeira
luisft@fe.up.pt

1 INESC TEC, Campus da FEUP, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal

2 Faculdade de Engenharia da Universidade do Porto (FEUP), Campus da FEUP, Rua Dr. Roberto Frias, n 378, 4200-465 Porto, Portugal

1 Introduction

Much of the history of learning algorithms has focused on some idealized settings, in where independent and identically distributed observations are drawn from a fixed yet unknown distribution, and fixed and known concepts are available. In practice, these assumptions are unlikely to be all exactly true. Still, since they are a good approximation to some of real-life problems, there is a broad range of solutions for this task. However, in many of the real world applications some of the previous assumptions are violated, rendering the conventional algorithms suboptimal or impractical. For instance, if the process is not strictly stationary (the distribution of data changes over time), the future field data may come from a distribution different from the primitive field data on which the model was developed in the first place. Hence, the model would fail to reflect the latest concept(s).

Video surveillance is a crucial application where traditional learning settings do not hold, becoming the main source of big and challenging data these days (Huang 2014). While it is relatively easy and inexpensive to acquire a large amount of un-labelled data, obtaining labelled data is particularly acute. Additionally, due to the typical video acquisition setup, the concepts of interest are not known beforehand and the statistics of the collected data evolve in time.

In this paper, we discuss a learning setting appropriate to learn from data streams generated in a multi camera scenario in where all the previously identified assumptions are violated. We extend and explore a preliminary study (Khoshrou et al. 2014a) in various directions. First, the new learning concept is better formulated. Second, a new learning methodology based in the Universal Background Model is proposed as a natural solution for the new learning paradigm. The learning method presented in the preliminary study is also framed under the same paradigm. A further development is the instantiation of the framework with different visual descriptors, highlighting the robustness of the framework. The experiments reported at the end of this paper include a thorough testing on synthetic and real data, greatly extending the initial results.

In Sect. 2, we review the current incremental learning algorithms for visual data. The Universal Background Model is briefly introduced in Sect. 3. We provide a detailed presentation of the proposed learning method in Sect. 4, starting with an overview and then filling in the details. We discuss the experimental methodology in Sect. 5 and in Sect. 6 we present the results of our method on a variety of synthetic and real datasets. Finally, conclusions are drawn in Sect. 7.

1.1 Relevance and problem definition

Over the last decades, video surveillance began spreading rapidly, specifically targeted at public areas. Recording for hours, days, and possibly years, provides massive amounts of information coming from an evolving environment in where traditional learning methods fail to reflect evolutions taking place (Dick and Brooks 2003). In such environments, the underlying distribution of data changes over time—often referred to as *concept drift*—either due to intrinsic changes (pose change, movement, etc.), or extrinsic changes (lighting condition, dynamic background, complex object background, changes in camera angle, etc.). Thus, models need to be continually updated to represent the latest concepts. The problem is further aggravated when new objects enter the scene—referred to as *class evolution* in machine learning literature—as new models need to be trained for the novel classes.

Figure 1 demonstrates a typical surveillance scenario. Depending on the view angle and the quality of the camera, every surveillance camera covers an area called Field of View (FoV).

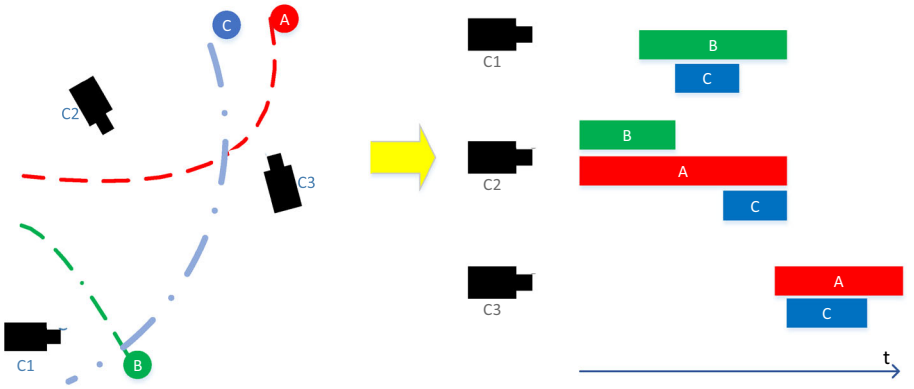
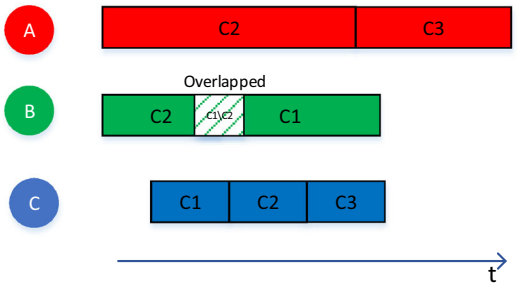


Fig. 1 Typical surveillance scenario (Khoshrou et al. 2014b)

Fig. 2 Desired outcome



When entering the scene, the object will enter the coverage area of at least one of the cameras. In such environments where objects move around and cross the FoV of multiple cameras, it is more than likely to have multiple streams, potentially overlapping in time, recorded at different starting points with various lengths, for the same individual object (Fig. 1). The surveillance system will have to track that object from the first moment it was captured by a camera and across all cameras whose fields of view overlap the object's path. Thus, a suitable outcome of the framework could be a timeline graph assigning each stream in each camera to an identity for the indicated presence period, as illustrated in Fig. 2. This graph can be used for behaviour analysis as well as security purposes. In this simple scenario the typical tracking systems are likely to encounter problems. In fact, mutual occlusion may occur if persons B and C cross. Consequently, their identities can be switched. Moreover, prolonged occlusion might occur, which might lead to track loss or mistaken identities (Teixeira and Corte-Real 2009). Since the cameras are supposed to track all objects in their coverage area, the definition of a global identity for each object is necessary. Multiple appearances of objects captured by the same or by different cameras are identified in the process, allowing also to know the path followed by a given object. This setting is inherently different from person re-identification scenarios, either image-to-image (Farenzena et al. 2010) or video-to-video (Pagano et al. 2014), that seek to determine if the images (videos) correspond to the person(s) of interest (Vezzani et al. 2013). Whereas, this framework focuses on the design of a system, where no pre-defined class of interest is available. Moreover, typical person re-identification works assume that the acquired data has enough detail to support identification on facial data, while in our setting appearance-based approaches are more likely to be successful.

Learning in such multi-camera scenario can be characterized as follows:

Definition Let $\mathscr{D}$ be a set of time-series $\mathscr{D}_i$. The starting points $t_{i,0}$ of the streams $\mathscr{D}_i$ may differ, the same being true for the ending points $t_{i,f}$. Each observation x within each stream is in a d-dimensional space, $x \in R^d$. Within each stream, the class explaining the observations is not the same for the whole duration of the stream. At a given time instance the same class may be responsible for the observations in multiple streams. Finally, the representation of a given class is not stationary, drifting with time.

Requirements An effective and appropriate algorithm to fit in our scenario is required to: (a) learn from multiple streams; (b) mine streams with various lengths and starting points (uneven streams); (c) handle concept drift; (d) accommodate new classes; (e) deal with partially labelled or unlabelled data; (f) be of limited complexity; (g) handle multi-dimensional data.

Herein we put forward a framework to learn continuously from parallel video streams with partially labelled data and that allow us to learn novel knowledge, reinforce existing knowledge that is still relevant, and forget what may no longer be relevant. The framework receives directly the tracked sequences outputted by the tracking system and maintains a global object identity common to all the cameras in the system.

Considerable body of multi-camera surveillance research assumes that adjacent camera view overlap (Chang and Gong 2001; Kuo et al. 2010; Hamid et al. 2014; Wang 2013), whereas (Javed 2005; Shan et al. 2005; Javed and Shah 2008; Pflugfelder and Bischof 2010; Matei et al. 2011) assume non-overlapping views. While our proposed method makes no assumption of overlapping or non-overlapping views. Hence, it can be applied in either settings.

1.2 Main contributions

In this paper, we present an ensemble of generative models that includes the maximum a-posteriori (MAP) adaptation of the *universal background model (UBM)*. This framework applies a double threshold strategy in order to detect novel classes and unreliable decisions. The decisions are categorized into three groups: novel classes when the existing classes are unable to explain satisfactorily the observed data; unreliable, leading to a request of user input; and reliable when there is strong evidence in favor of one of the existing classes. The adopted batch approach enables to achieve a good balance between the need to have enough data to make reliable decisions and the need to adapt quickly enough to drifts and new concepts in the data streams.

2 Literature review

Intelligent video surveillance (IVS) is a multi-disciplinary field, related to computer vision, pattern recognition, signal processing, communication, embedded computing and image sensors (Wang 2013); however, much of the history of IVS systems has addressed the problem employing computer vision techniques (Lim et al. 2003; Kuo et al. 2010; Matei et al. 2011; Zheng et al. 2011; Berclaz et al. 2011).

Whereas various SSL (Semi-Supervised Learning) methods have been proposed for video annotation (Song et al. 2005; Wang et al. 2009; Xu et al. 2012), deploying such methods in IVS systems is less explored. In Balcan et al. (2005), the person identification task is posed as a

graph-based semi-supervised learning problem, where only a few low quality webcam images are labelled. The framework is able to track various objects in limited drifting environments. The classification of objects that have been segmented and tracked without the use of a class-specific tracker has been addressed with an SSL algorithm in Teichman and Thrun (2011). Given only three hand-labelled training examples of each class, the algorithm can perform comparably to equivalent fully-supervised methods, but it requires full-length tracks (it is therefore an off-line process) generated by a perfect tracker (each stream represents a single object), which would be challenging for real applications, where multiple streams are available simultaneously. The underlying assumption made by most learning algorithms simply do not hold in real-world surveillance environment.

Learning from time-changing data streams has mostly appeared in data mining context and various approaches have been proposed (Gama et al. 2013; Keogh and Kasetty 2003). Ensemble-based approaches constitute a widely popular group of these algorithms to handle concept drift (Ackermann et al. 2012; Kolter and Maloof 2007) and in some recent works class evolution (Elwell and Polikar 2011), as well. Learn++.NSE (Elwell and Polikar 2011) is one of the latest ensemble-based classification methods in literature, that generates a classifier using each batch of training data and applies a dynamic weighting strategy to define the share of each ensemble in the overall decision. As success is heavily dependent on labelled data, this method would not be applicable in wild scenarios. Masud et al. (2011) proposed an online clustering algorithm for single stream that employs an active strategy in order to minimize oracle collaboration.

COMPOSE (Dyer et al. 2014) is designed for learning from non-stationary environment facing gradual drift but it cannot support neither abrupt drift nor class evolution. Although (Capo et al. 2013; Zliobaite et al. 2011; Ditzler and Polikar 2011) can handle more dramatic changes in data distributions, novel concept detection is an issue. Masud et al. (2010) presents ActMiner, which addresses the problem of concept-drift as well as concept evolution in a single infinite length data stream.

Since we look at the problem as learning from multiple data streams (herein, visual data) in wild environments, that views segments of a stream as a unique element to classify, single stream mining methods cannot be employed. Most of the methods proposed for parallel stream mining (Beringer and Hüllermeier 2006; Rodrigues et al. 2008; Chen 2009; Chen et al. 2012) require equal-length streams coming from a fixed number of sources. Thus, they would fail to leverage information from time-varying video tracks.

We addressed the problem of mining uneven streams in prior works (Khoshrou et al. 2014a, b). NEVIL (Khoshrou et al. 2014b) exploits an ensemble of *discriminative classifiers* in order to actively classify parallel video streams. If the reliability of a decision is below a user-defined threshold (either due to a class evolution or abrupt concept drift), NEVIL queries the oracle for labelling. It does not provide more information about the source of confusion, being unable to detect novel classes. Moreover, the classifier becomes biased towards the most frequent class in the case of severe class imbalance. Subsequent work (Khoshrou et al. 2014a) lessens the problem, using a class-based ensemble of Gaussian Mixture Models in the framework (NEVIL.g). Class-based ensemble was firstly introduced in Al-Khateeb et al. (2012) where a model is trained for each class in a chunk. The ensemble keeps a fixed size micro-ensemble of each class and it has been shown that this approach is more robust than traditional ensembles. Although NEVIL.g produces superior performance compared to NEVIL, stability in high-dimensional visual data is still a big issue and the novel class detection is unreliable due to the difficulty of setting a suitable threshold. In here we address those issues by adopting a UBM-normalized strategy and class-based ensembles.

3 Universal background model

Universal background modeling is a common strategy in the field of voice biometrics (Povey et al. 2008). It can be easily understood if the problem of biometric verification is interpreted as a basic hypothesis test. Given a biometric sample Y and a claimed ID, S, we define:

H_0: Y belongs to S
H_1: Y does not belong to S

as the null and alternative hypothesis, respectively. The optimal decision is taken by a *likelihood-ratio test*:

$$\mathscr{S}(Y|H_0) = \frac{p(Y|H_0)}{p(Y|H_1)} \begin{cases} \geq \theta & \text{accept} \quad H_0 \\ \leq \theta & \text{accept} \quad H_1 \end{cases} \tag{1}$$

where θ is the decision threshold for accepting or rejecting H_0, and $p(Y|H_i), i \in \{0, 1\}$ is the likelihood of observing sample Y under hypothesis i. Biometric recognition can, thus, be reduced to the problem of computing the likelihood values $p(Y|H_0)$ and $p(Y|H_1)$. Note that H_0 should characterize the hypothesized individual, while, alternatively, H_1 should be able to model *all the alternatives to the hypothesized individual*.

From such formulation arises the need for a model that successfully covers the space of alternatives to the hypothesized identity. The most common designation in literature for such a model is *universal background model* or *UBM* (Reynolds 2002). Such model must be trained on a large set of data, so as to faithfully cover a representative user space and a significant amount of sources of variability.

3.1 Hypothesis modeling

Gaussian Mixture Models (GMM) are typically chosen to model both the UBM, i.e. H_1, and the individual specific models (IDSM), i.e. H_0. Such models are capable of capturing the empirical probability density function (PDF) of a given set of feature vectors, so as to faithfully model their intrinsic statistical properties (Reynolds et al. 2000). The choice of GMM to model feature distributions in biometric data is extensively motivated in many works of related areas. From the most common interpretations, GMMs are seen as capable of representing broad "hidden" classes, reflective of the unique structural arrangements observed in the analysed biometric traits (Reynolds et al. 2000). Besides this assumption, Gaussian mixtures display both the robustness of parametric unimodal Gaussian density estimates, as well as the ability of non-parametric models to fit non-Gaussian data (Reynolds 2008). This duality, alongside the fact that GMM have the noteworthy strength of generating smooth parametric densities, confers such models a strong advantage as generative model of choice.

3.1.1 H_1: UBM parameter estimation

To train the Universal Background Model a large amount of "impostor" data, i.e. a set composed of data from all the enrolled individuals, is used, so as to cover a wide range of possibilities in the individual search space (Shinoda and Inoue 2013). The training process of the UBM is simply performed by fitting a k-mixture GMM to the set of feature vectors extracted from all the "impostors".

If we interpret the UBM as an "impostor" model, its "genuine" counterpart can be obtained by adaptation of the UBM's parameters using individual specific data. For each enrolled individual, ID, an *individual specific model* (IDSM) is therefore obtained.

3.1.2 H_0: MAP adaptation of the UBM

IDSMs are generated by the *tuning of the UBM parameters* in a maximum *a posteriori* (MAP) sense, using individual specific biometric data. This approach provides a tight coupling between the IDSM and the UBM, resulting in better performance and faster scoring than uncoupled methods (Xiong et al. 2006), as well as a robust and precise parameter estimation, even when only a small amount of data is available (Shinoda and Inoue 2013). This is indeed one of the main advantages of using UBMs. The determination of appropriate initial values (i.e. seeding) of the parameters of a GMM remains a challenging issue. A poor initialization may result in a weak model, especially when the data volume is small. Since the IDSM is learnt only from each individual data, it is more prone to a poor convergence that the GMM for the UBM, learned from a big pool of individuals. In essence, UBM constitutes a good initialization for the IDSM.

3.2 Recognition and decision

After the training step of both the UBM and each IDSM, the typical recognition phase in biometric systems is somewhat trivial. As referred in the previous sections, the identity check is performed through the projection of the new test data, X_{test}, onto both the UBM and either the claimed IDSM (in verification mode) or all such models (in identification mode). The recognition score is obtained as the likelihood-ratio. This is a second big advantage of using UBM. The ratio between the IDSM and the UBM probabilities of the observed data is a more robust decision criterion than relying solely on the IDSM probability. This results from the fact that some subjects are more prone to generate high likelihood values than others, i.e. some people have a more "generic" look than others. The use of a likelihood ratio with an universal reference works as a normalization step, mapping the likelihood values according to their global projection. Without such step, finding a global optimal value for the decision threshold, θ, presented in Eq. (1), would be a far more complex process.

4 Never ending visual information learning with UBM

In this section we present our framework named Never Ending Visual Information Learning with UBM (NEVIL.ubm). NEVIL.ubm is designed for non-stationary data environments in which no labelled data is available but the learning algorithm is able to interactively query the user to obtain the desired outputs at carefully chosen data points. The algorithm is an one-pass class-based ensemble of classifiers that trains a separate model (h_t^j) for a class j at every time slot t. It also keeps models of each class in a separate ensemble (*Micro-Ensemble*). A time-adjusted weighting strategy combines the probabilities outputted by the models in order to make the final decision.

4.1 Algorithm overview

Algorithm 1 outlines our approach. The framework receives multiple visual streams, generated by a typical tracking algorithm, which analyses sequential video frames and outputs the movement of targets between the frames. Inside each frame the data corresponds to some pre-selected object representation (e.g. bag of words, histogram). Experimentally we will evaluate the stability of NEVIL.ubm with several object representations. Environmental challenges such as varying illumination, lack of contrast, bad positioning of acquisition

Algorithm 1 NEVIL.ubm

Input: $\mathscr{D}_t^{m_i}, i = 1, \ldots, M$
$W_0 \leftarrow \frac{1}{k}$
$H_0 \leftarrow W_0$
while $\mathscr{D}_t$ is $True$ **do**
  Batch label prediction (Sect. 4.1.1)
  $\mathscr{S}(C_k|\mathscr{D}_t^{m_i}, H_{t-1}) \leftarrow (\mathscr{D}_t^{m_i}, H_{t-1})$
  Novelty Detection (Sect. 4.1.2)
  $\max_{C_k} \mathscr{S}(C_k|\mathscr{D}_t^{m_i}, H_{t-1}) < T \Rightarrow \mathscr{D}_t^{m_i} \subset$ novel class
  Batch Confidence Level Estimation (Sect. 4.1.2)
  $BCL \leftarrow \mathscr{S}(C_k|\mathscr{D}_t^{m_i}, H_{t-1})$
  Model design (Sect. 4.1.3)
  $h_t^j \leftarrow \mathscr{D}_t^j, \ j = 1, \ldots, k$
  Composite model structure and update (Sect. 4.1.4)
  $ME_t^j \leftarrow h_t^j, \ \ j = 1, \ldots, k$
  $H_t \leftarrow (ME_t^1, \ldots, ME_t^k, \ H_{t-1}, \ W_t)$
end while

devices, blurring caused by motion as well as occlusion make data often noisy and/or partially missing. We address these challenges by a batch divisive strategy, as learning from a data batch may reduce the noise and fill the gaps caused by miss-tracking. Initially, the composite model is initialized to yield the same probability to every possible class (uniform prior). When the batches $\mathscr{D}_t^{m_i}$ in time slot t become available, the framework starts computing the scores $\mathscr{S}(\mathscr{D}_t^{m_i}|C_k, H_{t-1})$ for each batch $\mathscr{D}_t^{m_i}$ in the time slot. The scores are obtained from the likelihood ratio test of the batch data obtained by the individual class model C_k and the UBM Fig. 3. This kind of on-line learning approach addressed in this work can suffer if labelling errors accumulate, which is inevitable. Unrelated objects will sooner or later be assigned the same label or different labels will be assigned to different views of the same object. To help mitigate this issue, we allow the system to interact wisely with a human, to help it stay on track. Once $\mathscr{S}(\mathscr{D}_t^{m_i}|C_k, H_{t-1})$ is obtained, a batch confidence level (BCL) is estimated; if BCL is high enough (above a predefined threshold), the predicted label

$$\arg\max_{C_k} \mathscr{S}(\mathscr{D}_t^{m_i}|C_k, H_{t-1})$$

is accepted as correct; if the BCL is very low (lower than a pre-determined second threshold), the batch data is assigned to a novel class; otherwise the user is requested to label the data batch. The labelled batches (either automatically or manually) are used to generate new separate models h_t^k (k runs over all the classes available in t), which are then integrated in the composite model, yielding H_t. Four tasks need now to be detailed: (a) the batch label prediction (by the composite model); (b) novelty detection and batch confidence level estimation (c) the individual class model design in current time slot; (d) the composite model structure and update.

4.1.1 Batch label prediction

A batch $\mathscr{D}_t^{m_i}$ is a temporal sequence of frames $\mathscr{D}_{t,f}^{m_i}$, where f runs over 1 to the batch size B. The composite model, H_{t-1}, can be used to predict directly $p(\mathscr{D}_{t,f}^{m_i}|C_k, H_{t-1})$ but not $p(\mathscr{D}_t^{m_i}|C_k, H_{t-1})$. The individual scores per frame $\mathscr{S}(\mathscr{D}_{t,j}^{m_i}|C_k, H_{t-1})$ can be immediately obtained as $\mathscr{S}(\mathscr{D}_{t,j}^{m_i}|C_k, H_{t-1}) = \frac{p(\mathscr{D}_{t,j}^{m_i}|C_k, H_{t-1})}{p(\mathscr{D}_{t,j}^{m_i}|UBM)}$. The batch label prediction can be analysed

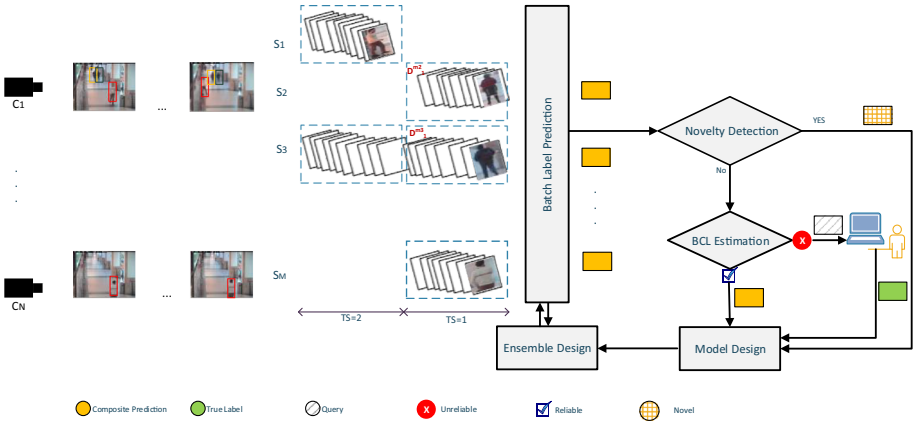


Fig. 3 NEVIL.ubm high-level overview

as a problem of combining information from multiple (B) classification decisions. Considering that, per frame, the composite model produces approximations to the likelihoods/scores for each class, different combination rules can be considered to build the batch prediction from the individual frame predictions (Alexandre et al. 2001; Kittler et al. 1998). Assuming independence between the scores of the individual frames, the score per batch is readily obtained as

$$\mathscr{S}(\mathscr{D}_t^{m_i}|C_k, H_{t-1}) = \sqrt[B]{\prod_{j=1}^{B} \mathscr{S}(\mathscr{D}_{t,j}^{m_i}|C_k, H_{t-1})} \tag{2}$$

Some authors have shown that the arithmetic mean outperforms the geometric mean in the presence of strong noise (Alexandre et al. 2001; Kittler et al. 1998). Thus, as a second option we defined the BCL as:

$$\mathscr{S}(\mathscr{D}_t^{m_i}|C_k, H_{t-1}) = \frac{\sum_{j=1}^{B} \mathscr{S}(\mathscr{D}_{t,j}^{m_i}|C_k, H_{t-1})}{B} \tag{3}$$

In our scenario, it is very likely to obtain outlier values for some frames in a batch due to occlusion or miss tracking. The median might be seen as a better indication of central tendency than the arithmetic mean in such cases, since it is less susceptible to the exceptionally large or small values in data. Hence, as a third option we consider estimating the score of a given batch by:

$$\mathscr{S}(\mathscr{D}_t^{m_i}|C_k, H_{t-1}) = \mathscr{M}edian\ \{\mathscr{S}(\mathscr{D}_{t,j}^{m_i}|C_k, H_{t-1}),\ j = 1, \ldots, B\} \tag{4}$$

Although other robust statistics could be considered from the individual frame scores, experimentally we will only compare the three options.

In the end, NEVIL.ubm assigns each batch to the class maximizing $\mathscr{S}(\mathscr{D}_t^{m_i}|C_k, H_{t-1})$.

4.1.2 Novelty detection and batch confidence level (BCL) Estimation

In our scenario, the number of classes is unknown beforehand. When a previously unobserved person enters the area of coverage by the camera network, the system should create a new

model to represent the novel class. We consider automating this decision. Applying a threshold to detect novel classes is extensively explored in the literature (Markou and Singh 2003).

In our NEVIL.ubm framework, if the scores associated to all observed classes ($\mathscr{S}(C_j|\mathscr{D}_t^{m_i}, H_{t-1})$, $j = 1, \ldots, k$) are significantly low (below a predetermined threshold), it is very likely that this class has not observed before and it is considered novel:

$$\max_{C_k} \mathscr{S}(C_k|\mathscr{D}_t^{m_i}, H_{t-1}) < T \Rightarrow \text{data belongs to a novel class}$$

Having decided that the batch data belongs to an existing class, one needs to decide if the automatic prediction is reliable and accepted or rather a manual labelling needs to be requested.

Various criteria have been introduced as uncertainty measures in literature for a probabilistic framework (Settles 2009). Perhaps the simplest and most commonly used criterion relies on the probability of the most confident class, defining the confidence level as

$$\max_{C_k} \mathscr{S}(C_k|\mathscr{D}_t^{m_i}, H_{t-1}) \tag{5}$$

This criterion only considers information about the most probable label. Thus, it effectively "throws away" information about the remaining label distribution (Settles 2009).

To correct for this, an option is to adopt a margin confidence measure based on the first and second most probable class labels under the model:

$$\mathscr{S}(C^*|\mathscr{D}_t^{m_i}, H_{t-1}) - \mathscr{S}(C_*|\mathscr{D}_t^{m_i}, H_{t-1}), \tag{6}$$

where C^* and C_* are the first and second most probable class labels, respectively. Intuitively, batches with large margins are easy, since the classifier has little doubt in differentiating between the two most likely class labels. Batches with small margins are more ambiguous, thus knowing the true label would help the model discriminate more effectively between them (Settles 2009).

Herein, we put forward a variant of the margin measure. We also evaluate experimentally the BCL base on the *ratio* of the first and second most probable class labels:

$$\mathscr{S}(C^*|\mathscr{D}_t^{m_i}, H_{t-1})/\mathscr{S}(C_*|\mathscr{D}_t^{m_i}, H_{t-1}), \tag{7}$$

4.1.3 Model design

The Universal Background Model is trained offline, before the deployment of the system. UBM is designed from a large pool of streams aimed to be representative of the complete set of potentially observable 'objects'. The training process of the UBM is simply performed by fitting a GMM to the set of feature vectors extracted from the complete pool.

At time slot t, we obtain a new set of batches that are automatically or manually labelled. We assume all the frames belonging to a batch are from the same object and that the M batches in a time slot correspond to $L < M$ labels (some batches can have the same label).

At each time slot, the data from the batches predicted to belong from the same class is used to generate the class model by *tuning of the UBM parameters*, in a maximum *a posteriori* (MAP) sense. This approach provides a tight coupling between the individual model and the UBM, resulting in better performance and faster scoring than uncoupled methods, as well as a robust and precise parameter estimation, even when only a small amount of data is available (Shinoda and Inoue 2013). The adaptation process consists of two main estimation

steps. First, for each component of the UBM, a set of sufficient statistics is computed from a set of M class specific feature vectors, $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$ computed from the batch data:

$$n_i = \sum_{m=1}^{M} p(i|\mathbf{x}_m) \tag{8}$$

$$E_i(\mathbf{x}) = \frac{1}{n_i} \sum_{m=1}^{M} p(i|\mathbf{x}_m)\mathbf{x}_m \tag{9}$$

$$E_i(\mathbf{x}\mathbf{x}^t) = \frac{1}{n_i} \sum_{m=1}^{M} p(i|\mathbf{x}_m)\mathbf{x}_m\mathbf{x}_m^t \tag{10}$$

where $p(i|\mathbf{x}_m)$ represents the probabilistic alignment of $\mathbf{x}_m$ into each UBM component. Each UBM component is then adapted using the newly computed sufficient statistics, and considering diagonal covariance matrices. The update process can be formally expressed as:

$$\hat{w}_i = [\alpha_i n_i/M + (1-\alpha_i)w_i]\,\xi \tag{11}$$

$$\hat{\boldsymbol{\mu}}_i = \alpha_i E_i(\mathbf{x}) + (1-\alpha_i)\boldsymbol{\mu}_i \tag{12}$$

$$\hat{\Sigma}_i = \alpha_i E_i(\mathbf{x}\mathbf{x}^t) + (1-\alpha_i)(\boldsymbol{\sigma_i}\boldsymbol{\sigma_i}^t + \boldsymbol{\mu_i}\boldsymbol{\mu_i}^t) - \hat{\boldsymbol{\mu}}_i\hat{\boldsymbol{\mu}}_i^t \tag{13}$$

$$\boldsymbol{\sigma_i} = \mathrm{diag}(\Sigma_i) \tag{14}$$

where $\{w_i, \boldsymbol{\mu_i}, \boldsymbol{\sigma_i}\}$ are the original UBM parameters and $\{\hat{w}_i, \hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\sigma}}_i\}$ represent their adaptation to the specific class. To assure that $\sum_i w_i = 1$ a weighting parameter ξ is introduced. The α parameter is a data-dependent adaptation coefficient. Formally it can be defined as:

$$\alpha_i = \frac{n_i}{r + n_i} \tag{15}$$

The relevance factor r weights the relative importance of the original values and the new sufficient statistics.

4.1.4 The composite model structure and update

Obtaining a meaningful stability-plasticity balance is a key issue in learning from non-stationary environments. The human learning system has addressed this issue by reinforcing existing knowledge that is still relevant, as well as forgetting what may no longer be relevant. The forgetting curve supports the process of forgetting that occurs with the passage of time (Schacter et al. 2011), which is exponential in nature. Inspired by human learning system, a strategic combination of an ensemble of classifiers, that employs dynamically assigned weights, is proposed in Elwell and Polikar (2011). Herein, we applied a time weighted strategy that gives more credit to more recent knowledge. Inspired by the forgetting curve, weights are chosen from the Taylor expansion of an exponential. The IDSM associated to the jth class, h_t^j, is stored in the jth ensemble, the so called Micro ensemble ME_t^j. Contrasting to the classic ensembles, a Micro ensemble includes models that are incrementally trained on incoming batches of a specific class, not all the batches (potentially from multiple classes) in a given timeslot. The composite model H_t is an ensemble of Micro ensembles ME_t^j, $j = 1, \ldots, K_t$, where K_t is the number of classes observed until time t. Each ME_t^j includes models h_t^j that are trained on incoming batches of the jth class since its appearance until the current time. The outputs of the individual models h_t^j are combined in ME_t^j using a weighted majority voting, where the weights are dynamically updated with respect to the classifiers'

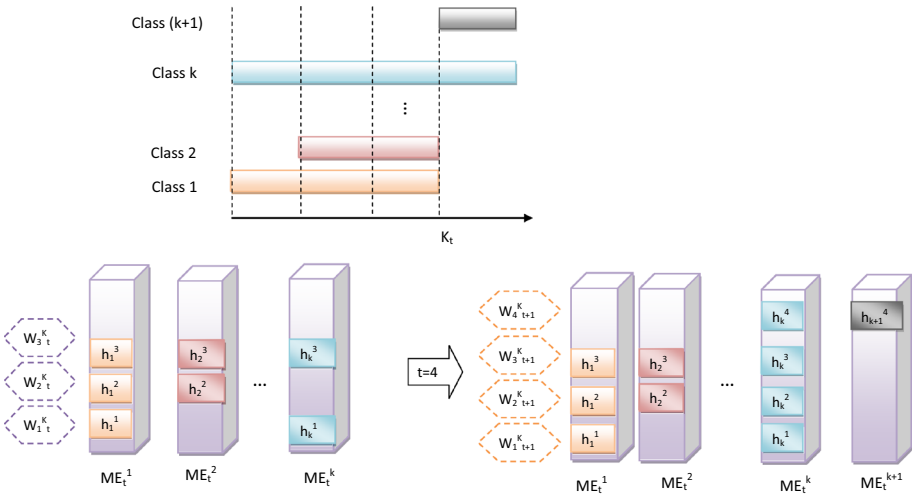


Fig. 4 An example of composite structure. Once a new class enters the scene (e.g. t = 4) , a new micro-ensemble is added to the composite

time of design. The prediction outputted by the composite model ME_t^j for a given frame $\mathscr{D}_{t,f}^{m_i}$ is

$$\mathscr{S}(C_k|\mathscr{D}_{t,f}^{m_i}, ME_t^j) = \sum_{\ell=1}^{t} W_\ell^t \mathscr{S}_\ell^j(C_K|\mathscr{D}_{t,f}^{m_i}),$$

where $\mathscr{S}_\ell^j(.)$ is the score outputted by $h_\ell^j(.)$ (the model trained from batches of jth class at TS ℓ), and W_ℓ^t denotes the weight assigned to model h_ℓ^j, adjusted for time t. The weights are chosen from a Taylor expansion of an exponential $(1, \ldots, (1-\alpha)^\ell)$ and are updated and normalised at each time slot to give more credit to more recent knowledge.

Figure 4 shows an example of how the composite is updated in a simplified scenario (a class is represented by a single stream). The IDSM associated to each class is trained and stored in the corresponding micro ensemble. For example, classes 1, 2 and k are available in the second timeslot. Hence three IDSM (h_t^1, h_t^2, h_t^k) associated to these classes are stored at ME_2^1, ME_2^2, and ME_2^k, respectively. Once a new class (k+1) appears at $t = 4$ (a new person enters the scene), a new micro ensemble ME_4^{k+1} is built. In order to get a decision on a frame (assign a score to the frame), the outputs of the models are combined using a weighted strategy that gives more credit to the more recent knowledge. Note that these weights are updated at every timeslot.

5 Experimental methodology

5.1 Datasets

We conducted our experiments on synthetic as well as real datasets. The synthetic data is generated in the form of (X, y), where X is a 2-dimensional feature vector, drawn from a Gaussian distribution $N(\mu_X, \delta_X)$, and y is the class label. Since in real applications visual

Table 1 The datasets characteristics

Dataset	No. of streams	Range	No. classes	Imbalance degree	No. of cameras	Setting
Scenario I	8	(4 – 10 k)	5	1	–	–
Scenario II	8	(4 – 10 k)	5	0.25	–	–
Scenario III	6	(1 – 5 k)	3	0.4	–	–
Scenario IV	15	(2.5 – 10 k)	7	0.23	–	–
OneLeaveShopReenter1	3	(85 – 160)	2	0.28	2	Overlapped
OneLeaveShopReenter2	3	(63 – 347)	2	0.11	2	Overlapped
WalkByShop1front	6	(40 – 225)	4	0.22	2	Overlapped
EnterExitCrossingPaths1	6	(34 – 216)	4	0.23	2	Overlapped
OneStopEnter2	7	(51 – 657)	4	0.19	2	Overlapped
OneShopOneWait1	10	(36 – 605)	4	0.25	2	Overlapped
OneStopMoveEnter1	42	(10 – 555)	14	0.14	2	Overlapped
PETS2009	19	(85 – 576)	10	0.13	2	Overlapped
SAIVT-SOFTBIO	33	(21 – 211)	11	0.12	8	Overlpped, nonoverlpped

Imbalance degree (Nguyen et al. 2009) is defined by the ratio of sample size of minority class to that of the majority ones ; range is defined by the length of shortest and longest streams in a given dataset, respectively

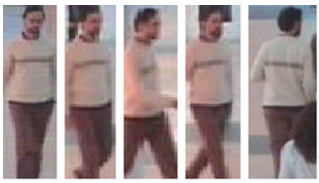


Fig. 5 An example of diversity in appearance

data may suffer from both gradual and abrupt drift, we tried to simulate both situations in our streams by changing μ_X and δ_X in the parametric equations. In this experiment, we generated 7 classes ($C_1, C_2, \ldots, C_7$); for some (C_5, C_6) data changes gradually while others also experience one (C_1, C_4, C_7), or three (C_2, C_3) dramatic drifts. This process is similar to the one used in Elwell and Polikar (2011). The dataset was organized in four different scenarios with different levels of complexity, including streams with gradual drift, abrupt drift, re-appearance of objects and non-stationary environments where we have gradual and abrupt concept drift as well as class evolution:

- *Scenario* I: gradually drifting streams of five classes.
- *Scenario* II: streams with abrupt drifts of five classes.
- *Scenario* III : re-appearance of objects.
- *Scenario* IV: a non-stationary environment with class evolution as well as concept drift.

More information is available at Khoshrou et al. (2014b).

Besides synthetic datasets, we run our experiments on public indoor (CAVIAR Project Consortium 2001), SAVIT-SOFTBIO (Bialkowski et al. 2012)] and outdoor (PETS) datasets. Seven scenarios of CAVIAR (*OneLeave ShopReenter1, Enter ExitCrossingPaths1, OneShopOneWait1, OneStop Enter2, WalkBy Shop1front*) have been used. Each clip was recorded

from two different points of view: view of the corridor, and a frontal view of the scenario. Due to the presence of different perspectives of the same person, streams are drifting in time (see Fig. 5). Two views of scenario S2.L1 of PETS2009 have been applied in our experiments. We carry out experiments on a subset the SAVIT-SOFTBIO, as well. This dataset consists of 11 people moving through a network of 8 cameras. Subjects move in an uncontrolled manner, which provides a highly unconstrained environment reflecting real-world conditions. These sequences present challenging situations with cluttered scenes, high rates of occlusion, different illumination conditions as well as different scales of the person being captured. We employed an automatic tracking approach (Teixeira et al. 2012) to track objects in the scene and generate streams of bounding boxes, which define the tracked objects' positions. As the tracking method fails to perfectly track the targets, a stream often includes frames of distinct objects. We wrap up this section in Table 1, presenting a qualitative look at the characteristics of the datasets applied in our work. Various factors have been considered in the table including: imbalance degree (Nguyen et al. 2009) that is defined by the ratio of sample size of minority class to that of the majority one; range in the length of streams that defines the length of shortest and longest streams in a given dataset, respectively.

5.2 Image representation

The choice of visual features, or image representation, is a key choice in the design of any classic image classification system (Chatfield et al. 2014). Seems fair to say that most of improvement in such system performance can be associated to the introduction of improved representation from the classic Bag of Visual Words (BoW) (Csurka et al. 2004) to the Fisher Vector (FV) (Perronnin et al. 2010). In such approaches, local image features [herein, SIFT (Lowe 1999)] are extracted. These features are encoded in a high dimensional image representation.

In order to evaluate the stability of the framework, we study the impact of different representations in the performance of NEVIL.ubm. We evaluate three encoding approaches: hard quantization, soft quantization (hierarchical bag-of-visterms), and Fisher method. Classical BoW computes a *spatial histogram (hard quantization)* of visual words constituting the baseline representation. Recent methods replace hard quantization with methods that retain more information. This can be done in two ways: (1) soft quantization or in other words, expressing the representation as combination of visual words (e.g. Teixeira and Corte-Real 2009), and (2) expressing the representation as the difference between the features and visual words (e.g. FV) (Chatfield et al. 2011).

In order to extract the hard quantized representation, we used a dictionary with 8000 visual words in classic BoW that provides 96,000 features for each frame. Following the approach in Teixeira and Corte-Real (2009), a hierarchical bag-of-visterms method is applied to represent the tracked objects, resulting in a descriptor vector of size 11,110 for each bounding box (soft quantized representation). In Fisher encoding, visual words are represented by means of a GMM, and the average first and second order differences of image descriptor and the visual words are recorded as global representation. We use a GMM with $k = 256$, resulting in a vector size of 327,680 for each bounding box. We used the implementation provided in Chatfield et al. (2011) to extract hard quantized and Fisher Vector features.

To avoid the curse of dimensionality, Principle Component Analysis (PCA) is applied to the full set of features as a pre-processing step. The number of features in each stream is reduced to 85 features for hard quantization and 350 dimensions for both soft quantization and FV.

5.3 Baseline methods

We compared our proposed NEVIL.ubm framework with two groups of baseline approaches: (1) unwise methods, in where the query is blindly requested. (2) wise approaches that select queries meticulously.

Unwise strategy

In such methods (e.g. Random strategy), queries are blindly chosen. The Random strategy (Zliobaite et al. 2014) labels the incoming batches randomly instead of wisely deciding which batches are more informative. Constrained by budget, batches are sent for annotation.

Wise methods

To the best of our knowledge, there is no approach [except NEVIL (Khoshrou et al. 2014b) and NEVIL.g (Khoshrou et al. 2014a)] that can be used in our learning setting. We stress that the methods in the literature fail to classify uneven parallel streams.

NEVIL (Khoshrou et al. 2014b) trains a classifier (employing discriminative approaches) per time slot. The classifiers are kept in an ensemble and participate in the final decision using a weighted sum strategy. If the decision is not reliable enough, the batch will be sent to an oracle for annotation.

NEVIL.g (Khoshrou et al. 2014a) employs a class-based ensemble of GMMs. A computational model is built for individual classes available in a given time slot. Unreliable batches are chosen for manual labeling.

5.4 Evaluation criteria

Active learning aims to achieve high accuracy using as little annotation effort as possible. Thus, a trade-off between accuracy and proportion of labelled data can be considered as one of the most informative measures.

Accuracy

In a classical classification problem the disparity between real and predicted labels explains how accurately the system works. However, in our scenario the labels do not carry any semantic meaning (it is not a person recognition problem). The same person should have the same label in different batches, whichever the label. One is just interested that, whatever label is used to represent a person, it is correctly transported to the next batches. The labels are therefore permutable and just define a partition of the set of all batches according to which label was assigned to it. As such, when evaluating the performance of our framework we are just comparing the partition of the set of batches as defined by the reference labelling with the partition obtained by the NEVIL labelling. We adopted a generic partition-distance method for assessing set partitions, initially proposed for assessing spatial segmentations of images and videos (Cardoso and Corte-Real 2005; Kuhn 1955). Thus, the accuracy of the system is formulated as:

$$Accuracy = \frac{N - Cost}{N} \tag{16}$$

where N denotes the total number of batches, and $Cost$ refers to the cost, yielded by the assignment problem.

Annotation

Assume *MLB* and *TB* denote the manually labelled batches and all the batches available during a period (includes one or more time slots), respectively. The *Annotation Effort* is formulated as:

$$Annotation\ effort = \frac{\#MLB}{\#TB} \tag{17}$$

It is expected that the accuracy increases with the increase of the annotation effort.

Area under the learning curve (ALC)

ALC (Cawley 2011) is a standard metric in active learning research that combines *accuracy* and *annotation effort* into a single measurement. The rationale behind the use of such metric is that there is not a single budget level that everyone agrees is the reasonable cost for a particular problem. Hence, ALC, which provides an average of accuracy over various budget levels, seems to be a more informative metric. Herein, the learning curve is the set of accuracy plotted as a function of their respective annotation effort, a, $Accuracy = f(a)$. The ALC is obtained by:

$$ALC = \int_0^1 f(a)da \tag{18}$$

6 Results

Firstly, multiple tests were run to determine the optimal batch size for each dataset to be explored. The batch size was varied between 1 and 50 % of the size of the shortest stream available in each dataset. Experiments were repeated for 50 equally spaced values in that range. The optimal batch size varies and is influenced by the characteristics of the streams present in each dataset. Optimal batch sizes have been observed to range between 25 and 35 for video streams. In order to explore the properties of the proposed framework, we evaluated it on multiple datasets covering various possible scenarios in a multi-camera surveillance system.

6.1 Results on synthetic data sets

Table 2 provides a summary of ALC of baseline approaches as well as NEVIL.ubm on multiple synthetic datasets. We plot the accuracy of a given strategy as a function of annotation effort in Fig. 6. Results show that NEVIL.ubm outperforms all the other techniques, specially in more complex scenarios: *Scenario III* and *Scenario IV*. All the experiments were repeated ten times to smooth initialization variability. Results demonstrate that the new framework (NEVIL.ubm) outperforms the baseline methods, providing over 90 % accuracy with <10 % annotation for all the datasets.

6.2 Results on real video streams

We compared NEVIL.ubm against baseline methods on multiple real video data, where various lengths and number of streams from different classes are present. Results are provided in Table 3. We note that the results showed significant improvement in favour of wise strategies

Table 2 Assessment on synthetic datasets

Datasets	ALC			
	Random	NEVIL	NEVIL.g	NEVIL.ubm
Scenario I	0.544	0.976	0.990	0.990
Scenario II	0.532	0.943	0.980	0.986
Scenario III	0.613	0.882	0.886	0.983
Scenario IV	0.523	0.883	0.972	0.973

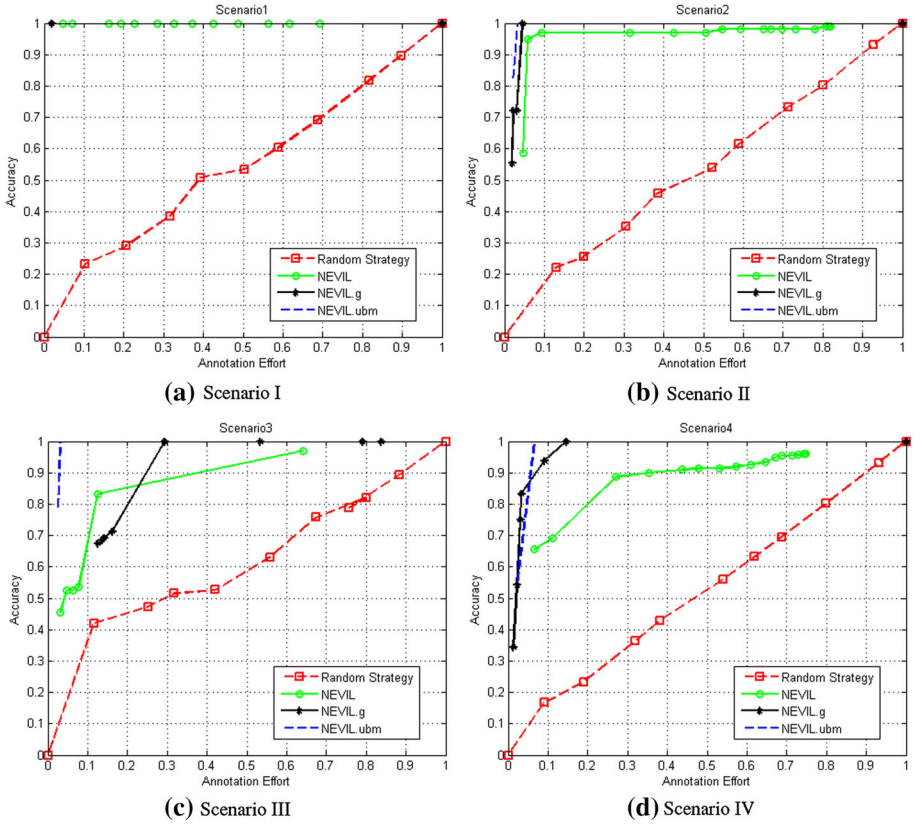


Fig. 6 Performance of baseline methods as well as NEVIL.ubm on synthetic datasets (Accuracy against Annotation effort). The signs *red squre*, *green circle*, *black asterisk*, *blue dash* denote the results of Random sampling, NEVIL, NEVIL.g, and NEVIL.ubm, respectively. **a** Scenario I. **b** Scenario II. **c** Scenario III. **d** Scenario IV (Color figure online)

in where queries are carefully chosen (Random strategy occupies the lowest place in the table). We observe that NEVIL.g and NEVIL.ubm are both significantly better than NEVIL. NEVIL was based on a discriminative learning of the models, being unable to detect novel classes. Therefore, it requires more user input for the same performance. Moreover, the learning of a multiclass classifier at each timeslot using only the subset of objects present in that timeslot is likely to induce false high likelihoods for the more recent classes. NEVIL.ubm has the highest ALC (except for "OneShopOneWait1" and "OneLeaveShopReenter2") and the best

Table 3 Comparison of NEVIL.ubm with baseline methods on real-world datasets

Methods	Datasets									Mean rank
	Reenter2	Reenter1	Wait1	Front	Path1	Enter2	Enter1	PETS2009	SAIVT	
Random strategy	0.69	0.63	0.59	0.68	0.62	0.66	0.51	0.56	0.57	**4**
NEVIL	0.76	0.90	0.84	0.79	0.74	0.84	0.78	0.68	0.82	**3**
NEVIL.g	0.94	0.89	0.90	0.88	0.85	0.91	0.81	0.71	0.88	**2**
NEVIL.ubm	0.93	0.96	0.88	0.93	0.86	0.95	0.87	0.79	0.92	**1**

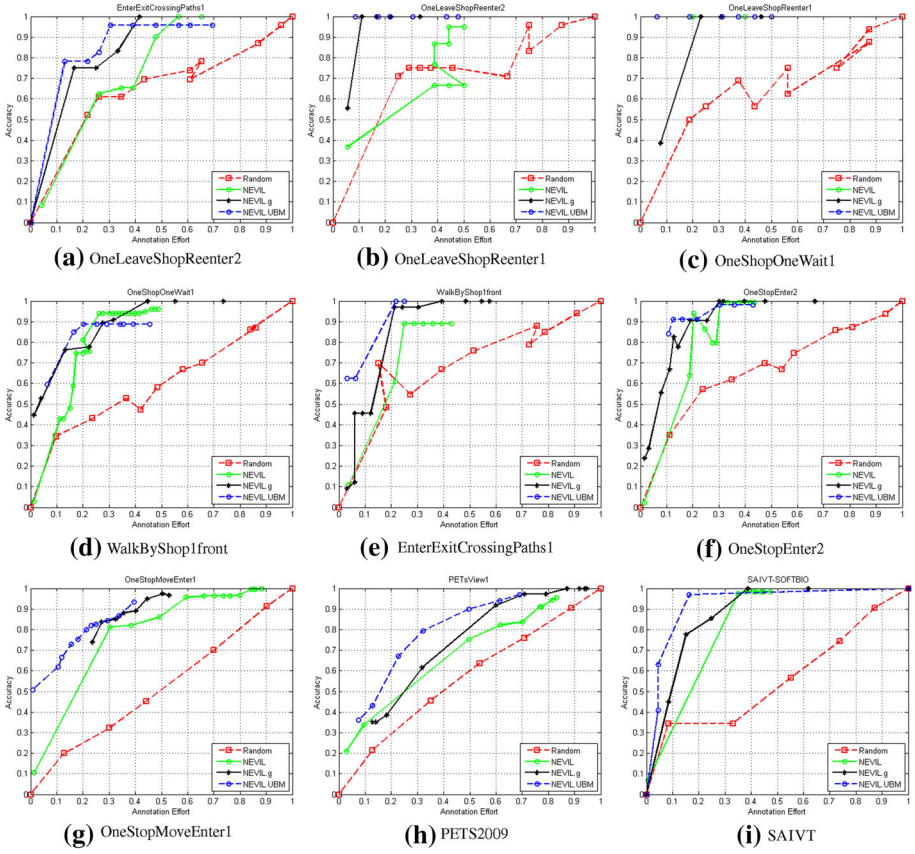


Fig. 7 Comparison of the performance of NEVIL, NEVIL.g, NEVIL.ubm on real-world datasets (Accuracy against Annotation effort). The signs *red squre*, *green circle*, *black asterisk*, *blue dash* denote the results of Random sampling, NEVIL, NEVIL.g, and NEVIL.ubm, respectively. **a** OneLeaveShopReenter2. **b** OneLeaveShopReenter1. **c** OneShopOneWait1. **d** WalkByShop1front. **e** EnterExitCrossingPaths1. **f** OneStopEnter2. **g** OneStopMoveEnter1 **h** PETS2009 (i) SAIVT (Color figure online)

mean rank over all the experiments. Figure 7 depicts the accuracy of various methods against the amount of queries placed on the operator.

Although there is not a single operating point in the Learning Curve suitable for all the applications, similarly to Culver et al. (2006), we chose the point obtained by labelling

20 % of batches for a more detailed analysis. Given that budget, we obtain 100 % for four scenarios (OneLeaveShopReenter2, OneLeaveShopReenter1, OneStopEnter2, and WalkByShop1front). For more complex scenarios, such as OneStopMoveEnter1 (in where 42 streams from 14 classes are available) 80 % of batches are correctly classified, showing a clear improvement over prior approaches. All the results are obtained using the most confident class as batch confidence measure and the median as the combination rule.

We presented multiple combination rules including sum, product and median and various confidence measure in Sects. 4.1.1 and 4.1.2, respectively. Any of the rules and measures can be applied in the framework. Table 4 provides a summary of the ALC measure for each setting on all datasets along with the mean of ALC rank averaged over all the experiments. We omit the margin measure results as it has shown results almost equal to the modified margin. The table shows that settings in where sum rule have been applied for combining the information occupy the two of top three spots (first and third). It is not surprising, since sum rule outperformed the product rule when complex data is present (Alexandre et al. 2001; Kittler et al. 1998). The results indicate that the most confident class as batch confidence measure selects more informative batches than modified margin, as settings employing the former have better mean rank. Based on the average rank, we conclude that the arithmetic mean as combination rule and the most confident as selection criterion represents the optimal design.

Timeline generation

Figure 8 shows an example of automatically labelled streams of "OneEnterExitCrossingPath1" and the respective ground truth (Fig. 8a). The framework assigns labels to the batches. It is desirable to assign the same identifier to all the streams of an individual object, however labels do not carry any semantic information (a name corresponds to a unique number in results). Figure 8b shows the output of the framework when 7 % of batches are labelled. The framework fails to identify the second class. Figure 8c can be considered as a successful case, since all objects are correctly identified. The main difference to the groundtruth is the miss identification of a stream. As the second object made a brief appearance in the scene, and he is heavily occluded, the stream experiences an abrupt drift.

6.3 Stability

In many classical object recognition problems, the representation plays an important role in the performance of the system. Our scenario as a pseudo object classification is not an exception. In here we analyse the impact of the representation in the performance of NEVIL.ubm. We compare the performance reached with the three descriptors introduced in Sect. 5.2. Table 5 lists FV as the top rank representation, attaining the lowest mean rank. We observe that the performance of NEVIL.ubm does not change much with the representation, presenting a good stability.

6.4 Memory

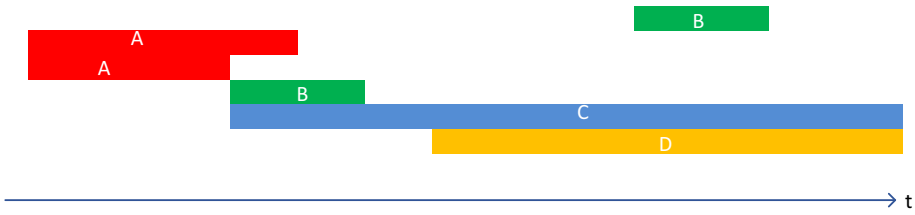
Decisions made by models inside ensembles are combined in respect to time (ℓ). Models are incrementally forgotten, to give emphasis to models built from more recent data.

To evaluate the impact of the forgetting factor (α), we kept batch size constant letting α vary. Results are plotted in Fig. 9. We observe that based on the datasets characteristics, exploiting previous models could have different impacts on the final results; for scenarios in where data drifts abruptly and re-reappearance of classes is not present (e.g scenario2,

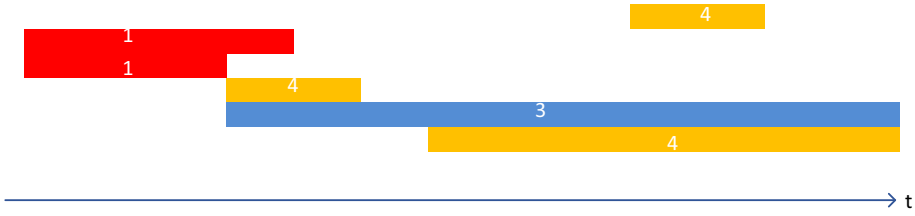
Table 4 Multiple settings of NEVIL.ubm on real-world datasets

Confidence measure	Combination method	Datasets									Mean rank
		Reenter1	Reenter2	front	Paths1	Enter2	Wait1	Enter1	PETS2009	SAIVT	
Most confident class	Median	**0.96**	0.93	0.93	0.86	0.95	0.88	0.87	0.79	0.88	**3**
	Prod	0.93	**0.97**	0.931	0.85	0.92	0.901	0.89	0.81	0.89	**4**
	Sum	0.96	**0.97**	0.90	**0.87**	0.95	0.90	**0.90**	**0.85**	**0.92**	**1**
Modified margin	Median	**0.96**	0.91	0.95	0.87	0.93	0.91	0.87	0.71	0.86	**5**
	Prod	0.937	**0.97**	**0.97**	0.84	0.95	0.89	0.87	0.75	0.87	**6**
	Sum	0.962	0.95	0.93	0.85	**0.96**	**0.92**	0.89	0.75	**0.92**	**2**

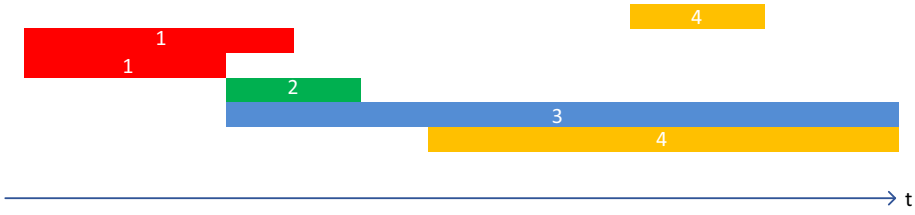
The best results (highest ALC) are given in bold



(a) Groundtruth



(b) Output of the framework using 7% labelling



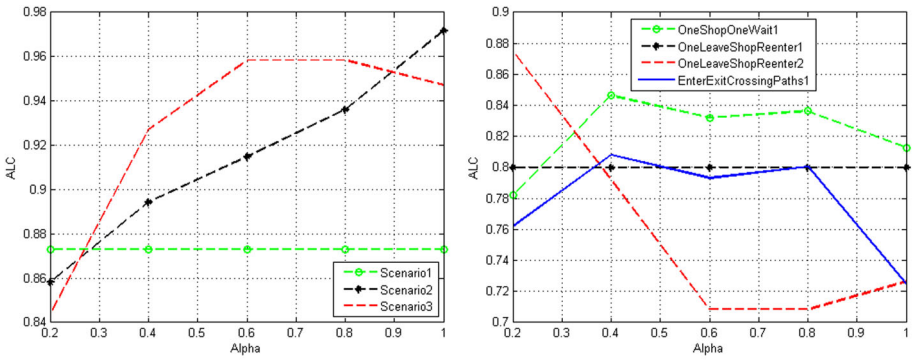
(c) Output of the framework using 21% labelling

Fig. 8 Streams of "OneEnterExitCrossingPath1", groundtruth and timeline outputted by the framework using different amount of labelling. **a** Groundtruth. **b** Output of the framework using 7 % labelling. **c** Output of the framework using 21 % labelling

Table 5 The ALC obtained with multiple descriptors

	Hard quantization	Bag of vistream	Fisher kernel
OneLeaveShopReenter2	0.97 (1)	0.95 (2)	0.95 (3)
OneLeaveShopReenter1	0.93 (3)	0.96 (2)	0.97 (1)
OneShopOneWait1	0.91 (2)	0.86 (3)	0.92 (1)
WalkByShop1front	0.79 (3)	0.93 (1)	0.89 (2)
EnterExitCrossingPaths1	0.81 (2)	0.86 (1)	0.76 (3)
OneStopEnter2	0.95 (2)	0.91 (3)	0.97 (1)
OneStopMoveEnter1	0.77 (3)	0.87 (1)	0.85 (2)
PETS2009	0.76 (3)	0.79 (1)	0.79 (2)
SAIVT	0.79 (3)	0.90 (2)	0.92 (1)
Mean rank	3	2	1

The rank of the descriptors in a given dataset is presented next to the ALC between parentheses



(a) Synthetic datasets **(b)** Multiple video clips

Fig. 9 Effect of forgetting factor (α) in ALC for various synthetic as well as real-world datasets. **a** Synthetic datasets. **b** Multiple video clips

OneShopReenter2), keeping the last model is enough. However, in a real world surveillance system, people may re-enter the scene after a while (which is the case for all our video clips except OneShopReenter2). Furthermore, the appearance of objects may (it is very likely) drift in time, but the drift is not strictly abrupt (which is the case in scenario II , in where data is generated from a completely different distribution). In such scenarios, the framework definitely gets advantage from proper choice of α. Through this proper range the choice of α is not critical (see Fig. 9b, when $\alpha \in [0.4, 0.8]$).

6.5 Time efficiency

Since NEVIL.ubm was developed in MATLAB without any efficiency concerns, a straightforward assessment of the time efficiency is not adequate. Nevertheless, some comments on the running time are in order. The analysis time grows naturally with the complexity of the dataset; the OneStopMoveEnter1 dataset was therefore the slowest to process. Although the time to process a batch grows with the batch size, since the time spanned by the batch also grows, the overall processing rate is not much affected by the batch size. Finally, ignoring the time to build the UBM model (done before the deployment of the system) the NEVIL.ubm framework was able to process in between real time and twice as fast the video streams, for a framerate of 25 fps (running in an Intel Core i7 at 3.2 GHz).

7 Conclusion

The typical learning settings already studied in the literature are not necessarily the most interesting for practical applications, since they may not represent well the information that is available.

In this paper, we present a learning setting yet unexplored in the literature but with wide relevance, especially in video surveillance. After formalizing the learning problem, we propose a class-based ensemble framework for the classification of parallel visual data streams. NEVIL.ubm framework is intended to learn from uneven parallel streams in non-stationary environments in where both concept drift and concept evolution are available.

The framework receives directly the tracked sequences outputted by the tracking system and maintains a global object identity common to all the cameras in the system. It adopts

a UBM-normalized strategy in a class-based ensemble, where an individual ensemble (so called micro-ensemble) is trained for every single class. The outputs of the individual models are combined in individual ME using a weighted voting, where the weights are dynamically updated with respect to the classifiers' time of design. Since accumulating the labelling error can severely damage this kind of on-line learning approach, we allow the system to interact wisely with an operator, to help it stay on track. The framework has shown promising performance with a fairly little human collaboration and can be applied in an on-line process.

There is still room for improvement in this framework: exploiting video specific descriptors (e.g. Bak et al. 2012; Wang et al. 2011), controlling the complexity of the ensemble, exploiting smarter novelty detection approaches, and expanding the experimental work to a large scale real-world dataset, all constitute our future work.

Acknowledgments The authors would like to thank Fundação para a Ciência e a Tecnologia (FCT)-Portugal for financing this work through the grant SFRH/BD/80013/2011.

References

Ackermann, M. R., Lammersen, C., Märtens, M., Raupach, C., Sohler, C., & Swierkot, K. (2012). StreamKM++: A clustering algorithms for data streams. *Journal of Experimental Algorithmics*, *17*, 173–187.

Al-Khateeb, T. M., Masud, M. M., Khan, L., & Thuraisingham, B. (2012). Cloud guided stream classification using class-based ensemble. In *Proceedings of the 2012 IEEE fifth international conference on cloud computing, IEEE Computer Society, Washington, DC, USA CLOUD '12* (pp. 694–701).

Alexandre, L. A., Campilho, A. C., & Kamel, M. (2001). On combining classifiers using sum and product rules. *Pattern Recognition Letters*, *22*(12), 1283–1289.

Bak, S., Charpiat, G., Corvée, E., Brémond, F., & Thonnat, M. (2012). Learning to match appearances by correlations in a covariance metric space. In *Computer vision—ECCV 2012—12th European conference on computer vision, Florence, Italy* (pp. 806–820). 7–13 Oct 2012, Proceedings, Part III. doi:10.1007/978-3-642-33712-3_58.

Balcan, M., Blum, A., Choi, P. P., Lafferty, J., Pantano, B., Rwebangira, M. R.,& Zhu, X. (2005). Person identification in webcam images:an application of semi-supervised learning. In *International conference on machine learning workshop on learning from partially classified training data* (pp. 1–9).

Berclaz, J., Fleuret, F., Türetken, E., & Fua, P. (2011). Multiple object tracking using k-shortest paths optimization. *IEEE transactions on pattern analysis and machine intelligence*, *33*(9), 1806–1819.

Beringer, J., & Hüllermeier, E. (2006). Online clustering of parallel data streams. *Data Knowledge Engineering*, *58*(2), 180–204.

Bialkowski, A., Denman, S., Sridharan, S., Fookes, C., & Lucey, P. (2012). A database for person re-identification in multi-camera surveillance networks. In *2012 international conference on digital image computing techniques and applications, DICTA 2012, Fremantle, Australia* (pp. 1–8). 3–5 Dec 2012. doi:10.1109/DICTA.2012.6411689.

Capo, R., Dyer, K. B., & Polikar, R. (2013). Active learning in nonstationary environments. In *IJCNN* (pp. 1–8).

Cardoso, J. S., & Corte-Real, L. (2005). Toward a generic evaluation of image segmentation. *IEEE Transactions on Image Processing*, *14*, 1773–1782. doi:10.1109/TIP.2005.854491.

CAVIAR Project Consortium. (2001). Caviar dataset. http://homepages.inf.ed.ac.uk/rbf/CAVIAR/.

Cawley, G. C. (2011). Baseline methods for active learning. (pp. 47–57).

Chang, T. H., & Gong, S. (2001). Tracking multiple people with a multi-camera system. In *IEEE workshop on multi-object tracking*.

Chatfield, K., Lempitsky, V. S., Vedaldi, A., & Zisserman, A. (2011). The devil is in the details: An evaluation of recent feature encoding methods. In *BMVC* (pp. 1–12).

Chatfield, K., Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*.

Chen, L., Zou, L., & Tu, L. (2012). A clustering algorithm for multiple data streams based on spectral component similarity. *Information Sciences*, *183*(1), 35–47.

Chen, Y. (2009). Clustering parallel data streams. Data mining and knowledge discovery in real life applications I-Tech education and publishing.

Csurka, G., Dance, C. R., Fan, L., Willamowski, J., & Bray, C. (2004). *Visual categorization with bags of keypoints* (pp. 1–22). In *In workshop on statistical learning in computer vision, ECCV*.

Culver, M., Deng, K., & Scott, S. D. (2006). Active learning to maximize area under the ROC curve. *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006), 18–22 December 2006* (pp. 149–158). China: Hong Kong.

Dick, A. R., Brooks, M. J. (2003). Issues in automated visual surveillance. In *Proceedings VIIth digital image* (pp. 195–204).

Ditzler, G., & Polikar, R. (2011). Semi-supervised learning in nonstationary environments. In *IJCNN* (pp. 2741–2748).

Dyer, K. B., Capo, R., & Polikar, R. (2014). Compose: A semisupervised learning framework for initially labeled nonstationary streaming data. *IEEE transactions on neural networks and learning systems*, *25*(1), 12–26.

Elwell, R., & Polikar, R. (2011). Incremental learning of concept drift in nonstationary environments. *IEEE Transactions on Neural Networks*, *22*(10), 1517–1531.

Farenzena, M., Bazzani, L., Perina, A., Murino, V., & Cristani, M. (2010). Person re-identification by symmetry-driven accumulation of local features. In *The twenty-third IEEE conference on computer vision and pattern recognition, CVPR 2010, San Francisco, CA, USA* (pp. 2360–2367). 13–18 June 2010. doi:10.1109/CVPR.2010.5539926.

Gama, J., Sebastião, R., & Rodrigues, P. P. (2013). On evaluating stream learning algorithms. *Machine Learning*, *90*(3), 317–346.

Hamid, R., Kumar, R. K., Hodgins, J. K., & Essa, I. A. (2014). A visualization framework for team sports captured using multiple static cameras. *Computer Vision and Image Understanding*, *118*, 171–183.

Huang, T. (2014). Surveillance video: The biggest big data. *Computing Now*, *7*(2), http://bit.ly/1penrSO.

Javed, O. (2005). Appearance modeling for tracking in multiple non-overlapping cameras. In *In IEEE international conference on computer vision and pattern recognition* (pp. 26–33).

Javed, O., & Shah, M. (2008). *Automated multi-camera surveillance: Algorithms and practice, the international series in video computing* (Vol. 10). Newyork: Springer.

Keogh, E., & Kasetty, S. (2003). On the need for time series data mining benchmarks: A survey and empirical demonstration. *Data Mining Knowledge Discovery*, *7*(4), 349–371.

Khoshrou, S., Cardoso, J. S., & Teixeira, L. F. (2014a). Active learning of video streams in a multi-camera scenario. In *22nd international conference on pattern recognition*.

Khoshrou, S., Cardoso, J. S., & Teixeira, L. F. (2014b) *Active mining of parallel video streams*. CoRR abs/1405.3382.

Kittler, J., Hatef, M., Duin, R. P. W., & Matas, J. (1998). On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(3), 226–239.

Kolter, J. Z., & Maloof, M. A. (2007). Dynamic weighted majority: An ensemble method for drifting concepts. *Journal of Machine Learning Research*, *8*, 2755–2790.

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, *2*(1–2), 83–97.

Kuo, C. H., Huang, C., & Nevatia, R. (2010). Inter-camera association of multi-target tracks by on-line learned appearance affinity models. In *Proceedings of the 11th European conference on computer vision: Part I, ECCV'10* (pp. 383–396).

Lim, S. N., Davis, L. S., & Elgammal, A. (2003). A scalable image-based multi-camera visual surveillance system. In *Proceedings of the IEEE conference on advanced video and signal based surveillance, IEEE Computer Society, Washington, DC, USA, AVSS '03* (pp. 205–212).

Lowe, D.G. (1999) Object recognition from local scale-invariant features. In *Proceedings of the international conference on computer vision, ICCV '99* (pp. 1150–1157).

Markou, M., & Singh, S. (2003). Novelty detection: A review-part 1: Statistical approaches. *Signal Processing*, *83*(12), 2481–2497.

Masud, M. M., Gao, J., Khan, L., Han, J., & Thuraisingham, B. M. (2010). Classification and novel class detection in data streams with active mining. In *PAKDD* (2) (pp. 311–324).

Masud, M. M., Gao, J., Khan, L., Han, J., & Thuraisingham, B. M. (2011). Classification and novel class detection in concept-drifting data streams under time constraints. *IEEE Transactions on Knowledge Data Engineering*, *23*(6), 859–874.

Matei, B. C., Sawhney, H. S., & Samarasekera, S. (2011). Vehicle tracking across nonoverlapping cameras using joint kinematic and appearance features. In *CVPR '11* (pp. 3465–3472).

Nguyen, G. H., Bouzerdoum, A., & Phung, S. L. (2009). *Learning pattern classification tasks with imbalanced data sets*. Intech. doi:10.5772/7544.

Pagano, C. C., Granger, E., Sabourin, R., Marcialis, G. L., & Roli, F. (2014). Adaptive ensembles for face recognition in changing video surveillance environments. *Information Sciences*, *286*, 75–101. doi:10.1016/j.ins.2014.07.005.

Perronnin, F., Sánchez, J., & Mensink, T. (2010). Improving the fisher kernel for large-scale image classification. In *Proceedings of the 11th European conference on computer vision: Part IV* (pp. 143–156). Berlin: Springer, ECCV'10. http://dl.acm.org/citation.cfm?id=1888089.1888101.

Pflugfelder, R., & Bischof, H. (2010). Localization and trajectory reconstruction in surveillance cameras with nonoverlapping views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(4), 709–721.

Povey, D., Chu, S. M., Varadarajan, B. (2008). Universal background model based speech recognition. In *IEEE international conference on acoustics, speech and signal processing* (pp. 4561–4564).

Reynolds, D. (2008). Gaussian mixture models. In *Encyclopedia of Biometric Recognition* (pp. 12–17).

Reynolds, D., Quatieri, T., & Dunn, R. (2000). Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, *10*(1), 19–41.

Reynolds, D. A. (2002). An overview of automatic speaker recognition technology. In *Acoustics, speech, and signal processing (ICASSP), 2002 IEEE international conference on* (Vol. 4, pp. IV–4072). IEEE.

Rodrigues, P. P., Gama, J., & Pedroso, J. P. (2008). Hierarchical clustering of time-series data streams. *IEEE Transaction on Knowledge Data Engineering*, *20*(5), 615–627.

Schacter, D., Gilbert, D. T., & Wegner, D. M. (2011). Psychology (2nd ed.). New York: Worth. http://www.amazon.com/Psychology-Daniel-L-Schacter/dp/1429237198/ref=sr_1_1?s=books&ie=UTF8&qid=1313937150&sr=1-1

Settles, B. (2009). Active learning literature survey. Technical Report 1648, University of Wisconsin–Madison.

Shan, Y., Sawhney, H. S., & Kumar. R. (2005). Unsupervised learning of discriminative edge measures for vehicle matching between non-overlapping cameras. In *CVPR* (1) (pp. 894–901).

Shinoda, K., & Inoue, N. (2013). Reusing speech techniques for video semantic indexing. *Signal Processing Magazine, IEEE*, *30*(2), 118–122.

Song, Y., sheng Hua, X., rong Dai, L., & Wang, M. (2005). Semi-automatic video annotation based on active learning with multiple complementary predictors. In *Proceedings of ACM SIGMM international workshop on multimedia information retrieval* (pp. 97–104).

Teichman, A., & Thrun, S. (2011). Tracking-based semi-supervised learning. In *Proceedings of robotics: Science and systems, Los Angeles, CA, USA*.

Teixeira, L. F., & Corte-Real, L. (2009). Video object matching across multiple independent views using local descriptors and adaptive learning. *Pattern Recognition Letters*, *30*(2), 157–167.

Teixeira, L. F., Carvalho, P., Cardoso, J. S., & Corte-Real, L. (2012). Automatic description of object appearances in a wide-area surveillance scenario. In *19th IEEE international conference on image processing* (pp. 1609–1612).

Vezzani, R., Baltieri, D., & Cucchiara, R. (2013). People reidentification in surveillance and forensics: A survey. *ACM Computing Surveys*, *46*(2), 29. doi:10.1145/2543581.2543596.

Wang, H., Klaser, A., Schmid, C., & Liu, C. L. (2011). Action recognition by dense trajectories. In *Proceedings of the 2011 IEEE conference on computer vision and pattern recognition, IEEE computer society, Washington, DC, USA, CVPR '11* (pp. 3169–3176). doi:10.1109/CVPR.2011.5995407.

Wang, M., Hua, X. S., Mei, T., Hong, R., Qi, G., Song, Y., et al. (2009). Semi-supervised kernel density estimation for video annotation. *Computer Vision and Image Understanding*, *113*(3), 384–396.

Wang, X. (2013a). Intelligent multi-camera video surveillance: A review. *Pattern Recognition Letters*, *34*(1), 3–19.

Wang, X. (2013b). Intelligent multi-camera video surveillance: A review. *Pattern Recognition Letters*, *34*(1), 3–19. doi:10.1016/j.patrec.2012.07.005.

Xiong, Z., Zheng, T., Song, Z., Soong, F., & Wu, W. (2006). A tree-based kernel selection approach to efficient gaussian mixture model-universal background model based speaker identification. *Speech Communication*, *48*(10), 1273–1282.

Xu, X .S., Jiang, Y., Xue, X., & Zhou, Z. H. (2012). Semi-supervised multi-instance multi-label learning for video annotation task. In *Proceedings of the 20th ACM international conference on multimedia, ACM, New York, NY, USA, MM '12* (pp. 737–740).

Zheng, W. S., Gong, S., & Xiang, T. (2011). Person re-identification by probabilistic relative distance comparison. In *CVPR* (pp. 649–656).

Zliobaite, I., Bifet, A., Pfahringer, B., & Holmes, G. (2011). Active learning with evolving streaming data. In D. Gunopulos, T. Hofmann, D. Malerba, & M. Vazirgiannis (Eds.), ECML/PKDD (3), Springer, Lecture Notes in Computer Science, (Vol. 6913, pp. 597–612).

Zliobaite, I., Bifet, A., Pfahringer, B., & Holmes, G. (2014). Active learning with drifting streaming data. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(1), 27–39.