Visual Re-Identification Across Large, Distributed Camera Networks

Vildana Sulić Kenk, Rok Mandeljc, Stanislav Kovačič, Matej Kristan, Melita Hajdinjak, Janez Perš

Visual Re-Identification Across Large, Distributed Camera Networks

Vildana Sulić Kenk[a,*], Rok Mandeljc[a], Stanislav Kovačič[a], Matej Kristan[a], Melita Hajdinjak[b], Janez Perš[a]

[a] *Machine Vision Laboratory, Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, SI-1000 Ljubljana, Slovenia*

[b] *Laboratory of Applied Mathematics, Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, SI-1000 Ljubljana, Slovenia*

Abstract

We propose a holistic approach to the problem of re-identification in an environment of distributed smart cameras. We model the re-identification process in a distributed camera network as a distributed multi-class classifier, composed of spatially distributed binary classifiers. We treat the problem of re-identification as an open-world problem, and address novelty detection and forgetting. As there are many tradeoffs in design and operation of such a system, we propose a set of evaluation measures to be used in addition to the recognition performance. The proposed concept is illustrated and evaluated on a new many-camera surveillance dataset and SAIVT-SoftBio dataset.

Keywords: Re-identification, Distributed sensors, Smart cameras, Visual-sensor networks, Surveillance

[*] Corresponding author:
Phone: +386 1 4768 876
Email address: vildana.sulic@fe.uni-lj.si (Vildana Sulić Kenk)

Preprint submitted to Image and Vision Computing *January 5, 2015*

1. Introduction

The increasing demand for security leads to a growing need for surveillance in many environments [1]. This includes installations of vast closed circuit TV (CCTV) systems; at the time of writing, London Underground has more than 12 000, and a typical casino in Las Vegas has more than 2000 surveillance cameras. Since visual sensors generate large amount of data, scalability becomes important, which gives rise to solutions based on distributed architectures – distributed camera networks. Computer-vision-based methods in camera networks are useful for different tasks, such as object detection and tracking, recognition of problematic or unlawful behavior, and re-identification of objects of interest. In this paper, we focus on the problem of re-identification [2, 3, 4, 5, 6], which is the process of finding correspondences between images of an object, acquired at different moments in possibly different camera views.

1.1. Challenges in distributed re-identification

Visual sensor networks (VSNs) may provide relatively large amount of computing and storage resources, but these are typically , both spatially and topologically distant, and the computational capability of an individual node may be low to reduce per-node cost or preserve energy [7]. Consequently, random access to a distant resource may be prohibitively expensive in terms of required network bandwidth, especially in wireless multi-hop networks. While this may be trivially alleviated by replicating all data and processing across all nodes, this defeats the purpose of a distributed architecture and does not solve the polynomially-increasing communication burden. In a truly distributed system, both re-identification and learning are *expensive opera-*

tions; the input data appears randomly at multiple nodes, thus requiring constant exchange with all other nodes, which may or may not have relevant information about object's identity.

1.2. Our contribution

We present a holistic approach towards object re-identification in distributed camera networks, which specifically addresses the issues of distributed environments. Specifically, we claim the following contributions:

- Formalization of object re-identification problem in a distributed environment.
- Treatment of re-identification as an open-world problem, with novelty detection and forgetting.
- A set of performance measures that specifically address issues in open-world distributed surveillance.
- Reproducible experiments on a many-camera surveillance dataset ("Dana36", [8]), 8-camera SAIVT-SoftBio dataset [9] and publicly available experimental source code[1]. The code reproduces all results and graphs from this paper. Researchers are encouraged to use it for rapid evaluation of their descriptors or datasets, evaluation of parameter influence and learning and forgetting strategies.

The remainder of this paper is organized as follows. After the overview of related work in Section 2, we explain the concept of re-identification in

[1] The full source code can be downloaded from: `http://vision.fe.uni-lj.si/research/reid/`

large, distributed camera networks in Section 3. The core of the proposed re-identification mechanism and the experimental methods we used are presented in Section 4, followed by experiments and results in Section 5. Section 6 concludes the paper.

2. Related work

The task of identifying an object based on its previous appearance in some other part of the camera network is called re-identification. In this respect, we can think about re-identification as form of large-scale tracking [10], which is comprised of several distinct challenges. Therefore, we address these separately.

Representation. The most frequently studied problem in re-identification is representation of object's appearance. We do not aim to improve the state-of-the-art in this respect, however, since object description is necessary part of any re-identification system, we present the work done so far for the sake of completeness.

Several approaches model whole body appearance, and have recently been compared by Doretto et al. [10]. Overall appearance is commonly modelled by color or brightness histograms, as for example in [11, 12, 13]. Spatial information can be added by representing appearances in joint color spatial spaces [14]. One of the popular approaches is a mixture of color features and texture features [2, 15, 16]. Other representations include spatio-temporal appearance modelling, such as [17] or spatial and appearance context modelling, such as [18]. Authors in [14] train a multi-class classifier for recognizing people using low-level feature, i.e., color and height histogram. In some approaches, as for example in [19], primitive features such as color, height and body aspect ratio are used in combination with simple threshold-based classification. There is a group of approaches that strives to normalize object appearance across multiple cameras, to improve the performance of appearance descriptors [20, 21].

Several approaches use training data to learn a holistic representation based on different low-level features, for example in [22] based on the bag-of-features representations, or in [23] based on Haar-like features and dominant color descriptors. Parts-based approaches are used as well. Part identification and correspondence can be carried out in several ways. One is to use interest point operators such as SURF [24] as in [25] or in [26] and SIFT [27], for example in [28].

Several authors identify body parts by other means. Bak et al. [29] propose an approach for person re-identification using spatial covariance regions [30] of human body parts, which are detected by using Histogram of Oriented Gradients (HOG, [31]). An approach proposed by Faranzena et al. [32] is based on a pondered extraction of local features that encoded different information: chromatic information, structural information through uniformly colored regions, and the nature of recurrent informative (in an entropy sense) patches. Recently, authors in [4] proposed a novel multiple-shot approach, which builds a specific human signature model based on Mean Riemannian Covariance (MRC) patches extracted from tracks of a particular individual. Authors in [33] evaluate different features, trying to find the most suitable ones for person re-identification. They conclude that despite recent advances, person re-identification using local features remains challenging, which might be due to existing descriptors describing mainly shape and texture.

There seems to be a consensus in scientific community that a person re-identification is a difficult problem and despite the best efforts from computer vision researchers, some claim that it remains largely unsolved [34]. Recently, topic models started to appear as a representation of choice in surveillance and re-identification tasks. Such models are usually based on the Latent Dirichlet Allocation (LDA, [35]), see for example [22]. When used for human appearance representation, LDA does not provide topics with obvious, humanly-understandable meaning. Therefore, Liu et al. [16] devised a semi-supervised method for topic generation that yields topics which can be easily interpreted.

Distributed surveillance systems. Further challenges arise from the need for *distributed representation*, which is especially important to guarantee efficient computation in large-scale networks.

As shown by recent work [26, 22, 23, 28, 36, 37], the community is increasingly aware of constraints in distributed systems. The multi-stage approach proposed by Jüngling et al. [28] provides local extraction of features on camera nodes, thus allowing the lower stages of re-identification to be performed by transmitting extracted features rather than images. Nevertheless, the approach builds its efficiency mainly on compact feature representation that is suitable for transmission and storage in distributed system, and does not provide a specific solution for efficient feature distribution in a large distributed camera system. In the system envisioned by Presti et al. [22], each node individually and autonomously processes the data acquired by its own camera. Communication among nodes enables knowledge sharing and is performed whenever an object leaves a camera's field of view. During the initialization phase, each node detects people and trains a LDA [35] model. These appearance models are propagated across the network and used both to describe incoming objects and to establish correspondences, but it is unclear how the underlying topic model is propagated. Authors claim that the knowledge of the camera network topology is not needed, but they only demonstrate results on data obtained from two cameras – a test case in which efficient feature distribution is obviously not an issue.

The issue of efficient feature propagation in large camera networks has been specifically addressed in our previous work [38]. We have shown that by using hierarchical encoding of features, it is possible to substantially decrease the amount of data transmitted across the network. However, such reduction is limited to *matching*, which is known in surveillance terminology as *matching to the gallery set* [15].

Novelty detection. An important concept in surveillance and person re-identification is the *novelty detection* [39]. Despite being a classic task in computer vision that had been previously addressed, e.g., [40, 41], novelty detection in surveillance received only limited attention, and was to the best of our knowledge used mainly in tasks such as detection of anomalies [42, 43, 44], detection of new classes of objects [45] or detection of unusual pedestrian behavior [46].

Evaluation and datasets. A large amount of work on pedestrian detection, tracking and activity analysis has been done in the framework of the successive PETS workshops. However, to the best of our knowledge, there are only few datasets that are specifically designed for identification and re-identification of pedestrians: the VIPeR dataset [2], the GRID dataset [47], the Person Reidentification dataset [3], 3DPeS [48] dataset, SAIVT-SoftBio [9] dataset, CUHKO02 [49] dataset, and our recent Dana36 dataset [8]. The first three provide only small number of images from one or two cameras, while 3DPeS contains video sequences for 200 people in a 8-camera multi-view setting, but provides bounding boxes for only about 1 200 frames (a subset named 3DPeS ReId Snap). SAIVT-SoftBio consists of image sequences of 150 people, with average 400 frames per person observed with 8 cameras, but the observed persons pass a particular camera view only once. CUHK02 contains images of 1 816 persons, but their identity is observed pairwise regarding the camera views, not on a global scale. The last one, Dana36 dataset, provides 23 683 images from 36 different camera views.

CAVIAR dataset[2] and iLids dataset[3] are not primarily intended for evaluation of re-identification but may be used for this purpose as well. Due to lack of well-annotated, many-camera datasets, it is not surprising that most of the previously-mentioned work [22, 3, 10, 28, 16] has been done on datasets that include up to five cameras. This is a relatively small number, which does not exhibit problems that are specific to large-scale camera systems.

Our aim is to address *those* problems, which in our opinion have so far received insufficient attention. They include re-identification across topologically distant nodes, novelty detection, and objective evaluation in truly distributed surveillance scenarios. Our work is in several aspects most closely related to [36, 28, 22], and in some aspects, extends the work of [38]. Contrary to most of the related work, a) we focus on systems with many cameras (e.g., 36 in our dataset), b) we assume communication constraints, c) we assume

[2] `http://homepages.inf.ed.ac.uk/rbf/CAVIAR/`

[3] `http://www.homeoffice.gov.uk/science-research/hosdb/i-lids/`

that the people re-identification in realistic setting has an *open world* nature, with unknown number of true identities, and d) we assume that pre-training of a such system is either infeasible or impractical.

3. Re-identification in large, distributed camera networks

Object re-identification in camera network essentially requires obtaining object correspondences between any pair of possibly distant camera nodes. In this respect, it would be advantageous to have a single *central processing server node* that aggregates information from all cameras. The main characteristic of such fully-centralized architecture is the ability of the processing node to locally access any piece of stored information. This is the setting that is *implicitly assumed*, but usually not *explicitly stated* in most of the research on surveillance re-identification. Therefore, fully-centralized architecture is spatially constrained, and state-of-the-art classification and recognition algorithms do not scale well with the growing network size due to non-zero communication cost. Additionally, methods that *assume* closed-world nature of the re-identification, cause the system to be severely *temporally constrained.* This is true for essentially any discriminative method that requires training-testing approach and is evaluated by cross-validation.

A realistic, distributed camera network cannot rely on these constraints. A distributed system in a realistic setting is forced to perform re-identification from just a few visual samples, perhaps even from a single previously-obtained image or tracklet, and has to decide on-the-fly whether a sample represents novel identity or not. Even if the best known learning and classification algorithms are run locally or on a group of locally-clustered nodes, they cannot readily use negative samples from distant nodes without intensive and prohibitively expensive communication across the network.

Under such circumstances, obtaining even a *basic feature correspondence between two distant nodes* becomes a non-trivial task, whose complexity and the associated communication cost increase polynomially with the network size. To keep even this basic problem manageable, one can use optimized algorithms for routing of queries across the network, such as hierarchical scheme for feature distribution (HFD) and basic object matching [38] or algorithm for grouping cameras into neighbourhoods [36]. If the capacity of the network allows, simple flooding can be used as well. In the rest of the paper we assume that the functionality of obtaining simple correspondence between the two distant nodes is available in the analyzed network.

4. Methods

In the proposed distributed method for object re-identification, we assume that image features have already been extracted from an image into a feature vector. Without any loss of a generality, features could be extracted from a set of images, or a video sequence, but in the rest of the paper, we use the term "image", which should be interpreted in a broad sense. We use a color histogram descriptor with some minor modifications, as described in Section 4.4. The descriptor is basic enough to allow quick and efficient demonstration of our framework and the effects that appear in camera network in a realistic setting. However, for practical applications, more sophisticated descriptors could be used, such as [20, 36].

4.1. The algorithm for distributed re-identification

We formulate the re-identification problem as follows: given a new image of a previously-seen person, the re-identification system has to be able to determine that person's identity. This is achieved by comparing the new image to an image set that contains examples for each known person. Such examples are called *gallery images* [2, 15] and the process is called *gallery matching.*

From a perspective of an external observer, the whole camera network behaves like a *multi-class* classifier. However, internally, this multi-class classifier consists of a number of binary classifiers that are distributed across a network, as illustrated by Figures 1 and 2.

4.1.1. Classification rule

Assuming that feature vectors $\mathbf{x_i}$ have been already extracted from the corresponding images, we define a set of gallery feature vectors, which represent unique identities of objects that are known to our system as

$$X_{gallery} = \{\mathbf{x}_i | i = 1, \ldots, L\}, \tag{1}$$

where L is the number of known identities. The situation is shown in Figure 1. When the system observes a new feature vector $\mathbf{x}$, it performs classification by comparing $\mathbf{x}$ to each sample from $X_{gallery}$. In terms of object classification, we are dealing with the set Ω of L binary classifiers, $\Omega = \{\omega_i | i = 1, \ldots, L\}$, each providing a binary decision $y_i \in \{-1, 1\}$ whether $\mathbf{x}$ belongs to the class i or not. The binary decision of classifier ω_i is based on the value of its gallery vector $\mathbf{x}_i$:

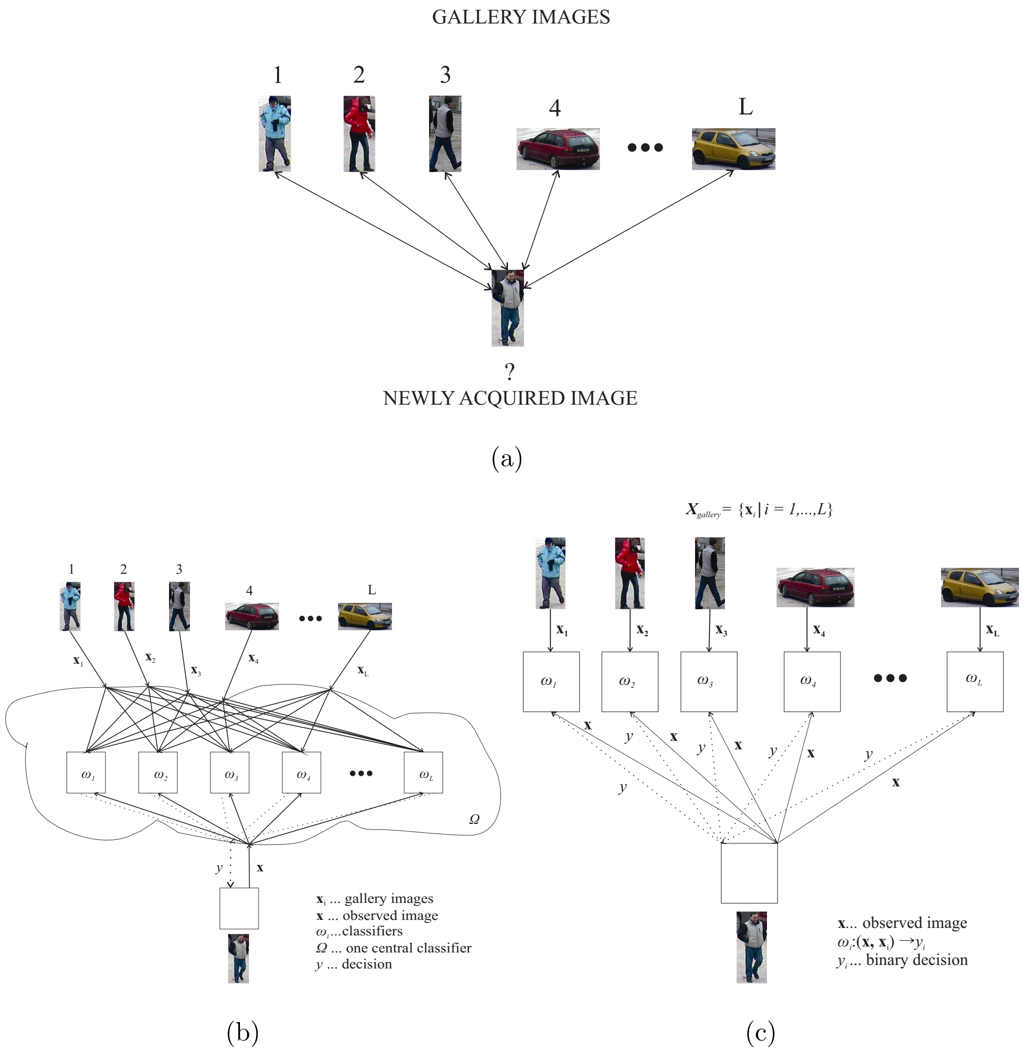


Figure 1: (a) Gallery matching – newly acquired image is compared to a set of gallery images with known identities to obtain identity of a person or object; (b) Centralized structure: one central classifier has access to all learning data. (c) A step towards distributed classification: each class, represented by a single gallery image is handled by a separate, single class classifier.

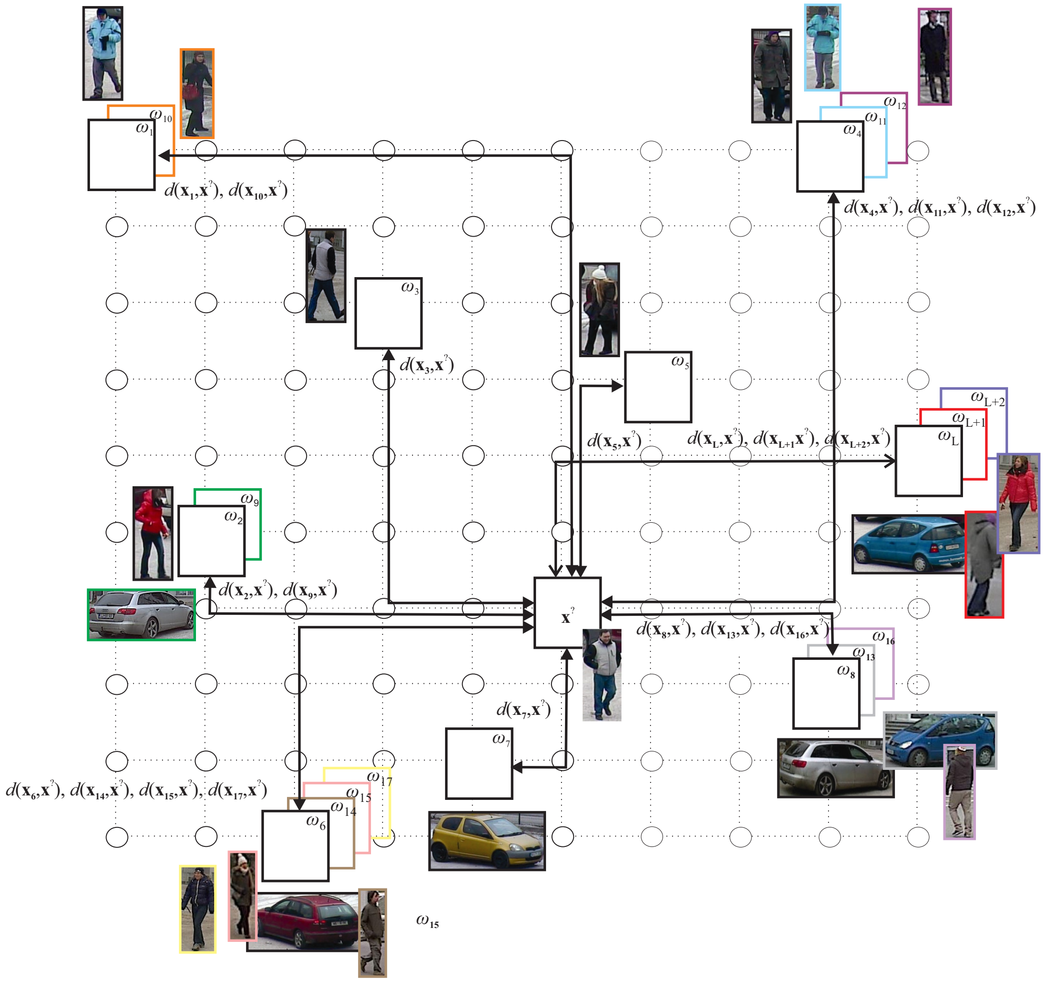
$$\omega_i : (\mathbf{x}, \mathbf{x}_i) \mapsto y_i. \tag{2}$$

In the simplest case, the classifier ω_i calculates the distance $d_i = d(\mathbf{x}_i, \mathbf{x})$ and applies the threshold T:

$$y_i = \begin{cases} +1 & d(\mathbf{x}_i, \mathbf{x}) \leq T \\ -1 & \text{otherwise.} \end{cases} \tag{3}$$

In a distributed system, objects are seen by different nodes on different occasions. Therefore, gallery images are scattered across the network and the cost of transmitting them to the single location is high. Effectively, this means that the classifiers ω_i are distributed across the network, as shown in Figure 2. Since we formulate the classification task as gallery matching using L classifiers and a global threshold T, such structure can exist in a distributed form, as for example in [38]. In a distributed setting, the communication between the nodes is costly, and therefore the distributed classifiers ω_i cannot efficiently compete for a best match. Consequently, the system may produce multiple positive answers, without the individual classifiers ω_i being aware of that. However, from the system perspective, the recipient of this information can aggregate the positive results from individual classifiers – transmission of distances that are below the threshold T across the network incurs only marginally higher communication costs than simply reporting the occurrence of the match. The recipient may then select the best match by comparing the multiple received distances, as follows:

$$y = \underset{i}{\operatorname{argmin}}\{d(\mathbf{x}_i, \mathbf{x})\} | d(\mathbf{x}_i, \mathbf{x}) \in \Delta_T \tag{4}$$



ω_{10}
ω_1
$d(\mathbf{x}_1,\mathbf{x}^?)$, $d(\mathbf{x}_{10},\mathbf{x}^?)$
ω_{12}
ω_{11}
ω_4
$d(\mathbf{x}_4,\mathbf{x}^?)$, $d(\mathbf{x}_{11},\mathbf{x}^?)$, $d(\mathbf{x}_{12},\mathbf{x}^?)$
ω_3
$d(\mathbf{x}_3,\mathbf{x}^?)$
ω_5
$d(\mathbf{x}_5,\mathbf{x}^?)$
ω_{L+2}
ω_{L+1}
ω_L
$d(\mathbf{x}_L,\mathbf{x}^?)$, $d(\mathbf{x}_{L+1}\mathbf{x}^?)$, $d(\mathbf{x}_{L+2},\mathbf{x}^?)$
ω_9
ω_2
$d(\mathbf{x}_2,\mathbf{x}^?)$, $d(\mathbf{x}_9,\mathbf{x}^?)$
$\mathbf{x}^?$
$d(\mathbf{x}_8,\mathbf{x}^?)$, $d(\mathbf{x}_{13},\mathbf{x}^?)$, $d(\mathbf{x}_{16},\mathbf{x}^?)$
ω_{16}
ω_{13}
ω_8
$d(\mathbf{x}_7,\mathbf{x}^?)$
ω_7
$d(\mathbf{x}_6,\mathbf{x}^?)$, $d(\mathbf{x}_{14},\mathbf{x}^?)$, $d(\mathbf{x}_{15},\mathbf{x}^?)$, $d(\mathbf{x}_{17},\mathbf{x}^?)$
ω_{17}
ω_{15}
ω_{14}
ω_6
ω_{15}

Figure 2: In a distributed system, gallery images and the corresponding classifiers are scattered across the network. Each classification still requires traversal of data across many network nodes (shown as circles), but since the classifiers are independent, the complexity of the problem is manageable. As illustrated, each node may contain several classifiers. Multiple classifiers may represent single true identity (for example, ω_1, ω_{11} and ω_2, ω_{L+2}) – a phenomenon called *identity fragmentation*, caused by false negative responses from ω_1 and ω_2.

where y is the final object label and Δ_T is the set of distances below the threshold T, as received from all the individual classifiers ω_i that reported the match. This formulation is similar to the operation of the *one-versus-all* multi-class classifier, but the distributed nature precludes centralized learning due to the communication cost.

4.1.2. Online learning and novelty detection

A typical surveillance environment is highly dynamic, and any pre-training is thus of a limited value. This aspect does not appear if cross-validation on a closed dataset is used – such approach *implicitly assumes* that a system can be successfully pre-trained and that the obtained knowledge never expires. Conversely, we assume that a re-identification system does not have any prior knowledge about objects – initially its gallery is empty. Therefore, online learning is a critical component of such a system. If the learning is *unsupervised* – a desirable property for *fully automated operation* – then the novelty detection is needed as well. We implement novelty detection as a complement of classification rule (3):

$$y^{novelty} = \begin{cases} +1 & d(\mathbf{x}_i, \mathbf{x}) > T \mid \forall i = 1, \ldots, L \\ -1 & \text{otherwise}, \end{cases} \tag{5}$$

where $y^{novelty} = +1$ signals that $\mathbf{x}$ is sufficiently distant from *all* of the gallery vectors in $X_{gallery}$ that it can be regarded as describing a novel object. Effectively, at this point, a new classifier ω_{L+1} is created from the novel feature vector $\mathbf{x}$ at the camera node that observed the object.

Such implementation of novelty detection has its own drawbacks – if the amount of false positives according to rule (5) is greater than zero (a realistic assumption), such scheme may result in a continuously increasing gallery and a continuously increasing number of the corresponding classifiers ω_i across the network. Depending on the circumstances, this number may be far larger than the true number of unique object identities ($L \gg N_{ID}$). This leads to *identity fragmentation* – since we are forced to assume that each classifier ω_i represents a unique object, a single object observed by the system may end up being recognized as several different entities.

The number of unique objects encountered in realistic surveillance environments is essentially unbounded. Consequently, the amount of knowledge accumulated during the operation of the system may be overwhelming. On the other hand, knowledge may become obsolete after a certain period of time after the last observation. People may change their clothing or appearance, or they may simply leave the observed area. Therefore, some kind of systematic *forgetting* needs to be implemented.

4.1.3. Forgetting

Conceptually, forgetting addresses the general problem of limited resource management. In our case, the resources are limited by the maximum number of classifiers ω_i and the storage capacity required to store the associated gallery feature vectors $\mathbf{x}_i$. Situations of similar nature have been already dealt with in computer science, e.g., cache management and page replacement algorithms.

When managing the number of classifiers ω_i, one should be aware of the following two basic factors:

- **Aging.** Probability that the data will be needed decreases with the time that has passed from the last observation of an object.

- **Limited resources.** In unfavorable conditions, some data must be discarded due to the lack of resources.

Accordingly, we define two parameters for our forgetting scheme. During the the run, each classifier ω_i is associated with its *time-to-live* counter τ_i. The parameter τ_{max} determines the maximum value of τ_i, at which classifier ω_i and its gallery feature vector $\mathbf{x}_i$ expire. Whenever the classifier ω_i provides a positive answer, the counter τ_i is reset to its initial value. This way, we prevent accumulation of outdated information, but retain the information that was recently used. On the other hand, each node can contain only the limited number of classifiers. Therefore, node j is associated with the counter of stored classifiers, λ_j. Hence, the second parameter is the maximum number λ_{max} of classifiers ω_i per node, which constrains the memory and processing resources used by our method, and reduces identity fragmentation.

Depending on those parameters and input data, the system operates between the two operating points: *limited lifespan*, where classifiers ω_i are discarded mainly due to their age and *limited capacity*, where classifiers ω_i are discarded mainly due to the appearance of freshly-learned ones.

4.2. Mapping onto HFD

The hierarchical feature-distribution scheme (HFD, [38, 50]) solves very narrow, yet fundamental problem in distributed camera networks: how to efficiently obtain correspondence between the acquired feature vector with unknown identity on one side, and the number of distant and topologically distributed feature vectors with known identities on the other. Using HFD, each node in the network has the ability to query the whole network for the objects that are similar to the observed object.

The classification rule (3) can be implemented either in a centralized or in a distributed system. It directly corresponds to classification approach as defined by HFD and therefore needs no additional modifications.

The classification rule for novelty detection (5) uses simply an inverted logic of the classification rule (3). Therefore, it maps onto HFD without modifications.

Given the classification rule (3), HFD can be viewed as an efficiently managed structure of many simple binary classifiers, distributed among the nodes. Those classifiers forward the query packets based on their own classification results, until the query reaches the node with authoritative classification answer (the actual classifier ω_i). A series of "routing classifiers" correspond to a single ω_i. Therefore, routing classifiers can share attributes (such as τ_i) with ω_i and follow its fate – dying when ω_i is removed from the classifier set due to forgetting.

4.3. The dataset

Recently, we published a dataset "Dana36"[4] [8]. It is intended for evaluation of object matching and recognition methods in surveillance scenarios. The dataset consists of 23 683 images depicting 15 persons and 9 vehicles. The dataset was acquired from 36 stationary camera views using a variety of surveillance cameras. 27 cameras observed the persons and vehicles in an outdoor environment, while the remaining 9 observed the same persons indoors. Due to the large number of camera views, the dataset is especially suitable for research on large-scale distributed camera networks in surveil-

[4] `http://vision.fe.uni-lj.si/research/dana36/`

lance scenarios. Instances of different objects are shown in Figure 3. In this work, we use color-histogram-based descriptor. The re-identification problem on the whole dataset is a difficult one due to large variations between cameras (resolution, location, vantage point, indoor, outdoor and mixed lighting) and similarity between many of the objects and persons. In this situation, it makes sense to either merge visually similar classes, or to retain only the classes that exhibit obvious visual difference. We chose the latter option, selecting 13 most visually-distinctive ones: persons labelled 1, 3, 5, 8, 9, 11, 12, 15 and cars labelled 16, 17, 18, 22 and 24 (referred to as 13-object subset Dana36$_{13}$ and consisting of $N_{Dana36,13}$ =13 483 images).

Dana36$_{13}$ dataset was complemented with the use of SAIVT-SoftBio [9] dataset, which provides image sequences of 152 persons, but only 8 camera views. However, for each observation, it provides the image sequence and corresponding bounding boxes. Since it provides the information about the temporal sequence of observations, it allowed us to simulate the exact, realistic sequence of events.

4.4. Object descriptor and distance measure

In our experiments, we use a *segmented color histogram*: a cropped image of an object is divided into 25 overlapping rectangular segments, and a set of 25 color histograms $H_k(k = 1, ..., 25)$ is computed. We use a three-dimensional RGB histograms with 4×4×4 bins, resulting in 1600-dimensional feature vector (64×25). To compare two sets of image features $\mathbf{x}_i$ and $\mathbf{x}_j$, we compute the distance $d(\mathbf{x}_i, \mathbf{x}_j)$ as the average distance across all image segments:

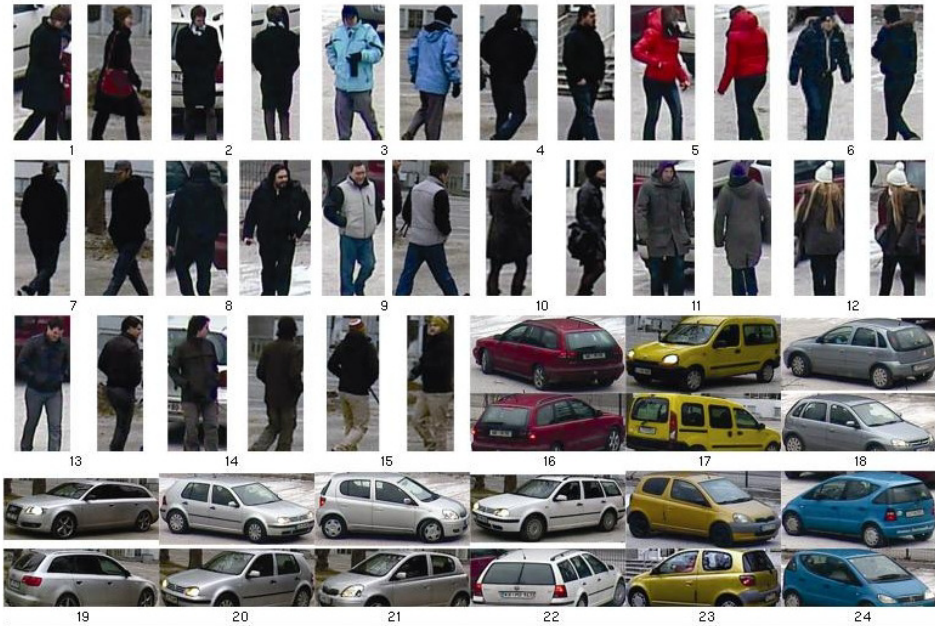


Figure 3: Objects from the "Dana36" dataset [8], two images per person and two images per car are shown.

$$d(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{25} \sum_{k=1}^{25} d_k(\mathbf{x}_i, \mathbf{x}_j) \tag{6}$$

The distance $d_k(\mathbf{x}_i, \mathbf{x}_j)$ between two segment histograms $H_{i,k}$ and $H_{j,k}$ is the Hellinger distance. The range of the distance measure $d(\mathbf{x}_i, \mathbf{x}_j)$ is between zero (complete similarity) and one (complete dissimilarity). The exact implementation of the descriptor is available as part of our source code download [5]. The descriptor has been adapted to work on image sequences as well; in that case, histograms are obtained by counting the pixel values inside bounding boxes across multiple images from the sequence.

[5] http://vision.fe.uni-lj.si/research/reid/

Using the Dana36 and SAIVT-SoftBio, the descriptor performs sufficiently well to illustrate our approach, and in combination with the threshold T it allows simple adjustment of the operating point of the classifier (the ratio of false and true positives of the binary classifiers ω_i). Therefore, we can easily observe our proposed approach at different operating points of the classifier set Ω.

4.5. Evaluation measures

To evaluate the proposed method for re-identification in distributed camera networks, we define a set of measures that are relevant to distributed re-identification.

4.5.1. Recognition performance

Since we deal with multi-class classification, we first evaluate multiclass recognition performance as seen from the recipient of the multiple distances scores (4). For this purpose, we obtain the confusion matrix, and calculate the multi-class accuracy ($Acc_{multiclass}$) as the ratio of results on the main diagonal of the confusion matrix vs. the number of all results.

However, $Acc_{multiclass}$ does not present the whole picture. In evaluation of the each new sample, all classifiers from the current classifier set Ω are consulted; this could be done in an optimized way, as shown for example in [38], or with simple flooding of the unknown sample across the network. In either case, a querying node receives none, one or more responses from the other network nodes, and the number of replies affects the network traffic. This aspect is mostly irrelevant in centralized implementation, however, in distributed setting, it is not. Therefore, we keep track of overall statistics from the binary classifiers, by observing the numbers of false positives (FP)

and true positives (TP), false negatives (FN), true negatives (TN) and total number of tested samples (M) across all classes (micro-averaging [51]). We declare a result of a classification of the vector $\mathbf{x}$ using the classifier ω_i as a true positive if the true identity of $\mathbf{x}$ corresponds to the identity represented by ω_i. Similarly, we count the number of FP, TN and FN. In our case, M denotes a number of individual tests done on all classifiers Ω, $\{\omega_i; i = 1, \ldots, L\}$. Finally, we calculate standard classification measures [52] – due to the effect they have on the network traffic, we primarily observe false positive rate (FPR) and true positive rate (recall or TPR).

The classification rates alone do not show the whole picture regarding the performance of the network. Therefore we keep track of the few additional values. U is the number of *unknowns*, or the objects that yield no positive answer from any classifier ω_i. We also observe the number of learned classifiers (corresponding to the number of gallery images – L) and the number of the unique object identities observed by the system (N_{ID}). Finally, we keep tally of the unique object identities *that are represented by the existing gallery* or classifier set ($N_{ID}(\mathit{\Omega})$).

From these counts we define three additional measures: identity fragmentation (F_{ID}), unknown rate (UR) and classifier coverage (C):

$$\begin{aligned} F_{ID} &= \frac{L - N_{ID}(\mathit{\Omega})}{N_{ID}} && (7) \\ UR &= \frac{U}{M} && (8) \\ C &= \frac{N_{ID}(\mathit{\Omega})}{N_{ID}} && (9) \end{aligned}$$

Identity fragmentation occurs due to the presence of false negatives: if all classifiers ω_i give negative result on a previously-seen object, this triggers unnecessary learning of this object's identity. If the same objects are learned more than once, the correct response of such system may result in several distinct labels. Naturally, an ideal system would have $F_{ID} = 0$, along with high recognition rates.

Second measure that is related to the same phenomenon is the unknown rate, UR. When all of the classifiers ω_i give negative result for an input sample, the system has to conclude that the object is *unknown* (and proceeds with learning). Therefore, in the absence of forgetting, the unknown rate is related to increase in the total number of learned classifiers, L – a greater unknown rate UR causes faster learning. At the beginning, an ideal system would have unknown rate $UR = 1$. If such ideal system was faced with the problem of closed nature (a finite number of object identities in input data) UR would then approach zero. In theory, unknown rate UR increases if we force the system to start forgetting accumulated knowledge.

The third measure is associated with the opposite phenomenon, which occurs due to non-zero FPR – occasionally, a system will observe a new object, but fail to recognize it as such, falsely assigning it to one of already known identities. If such behavior is consistent for a particular object, its identity will never be learned and the system will always produce erroneous response when encountering that object. In that case, such system will have classifier coverage $C < 1$. On the other hand, an ideal system would consistently have $C = 1$.

4.5.2. An illustrative example

The re-identification problem, as addressed in this paper, has a dynamic and open-world nature. To further illustrate the need for the proposed measures, we provide a step-by-step example, which address three hypothetic, yet realistic scenarios. Note that in these scenarios we assume a specific made up sequence of classifier decisions, to illustrate as many aspects of the proposed measures as possible.

The first scenario is a general one, depicted in detail in Figure 4, with its final outcome shown in Figure 5 (a). The surveillance system starts its observation with an empty set of classifiers $\Omega = \emptyset$ (not shown in Figure 4). The values of M and L increase with the number of test samples and the number of classifiers, respectively, so we do not explicitly track their progress. The sequence is started by a sample with identity (true class value) of 1. Since the classifier set Ω is empty, this sample is considered to be unknown, therefore the number of unknown samples U increases. The classifier ω_1 is created, and the number of identities represented by the classifier set, $N_{ID}(\Omega)$ is incremented to 1. In terms of classification, this sample does not influence TP,FP,TN or FN. The outcome of this step is shown in Figure 4 (a).

Suppose that the next sample has the true class value of 5, but is incorrectly classified as 1. Therefore, FP is incremented (Figure 4 (b)). The next sample has the true class value of 2, and after a negative classification result by the only classifier ω_1, it is declared as unknown. Therefore, a new classifier ω_2 is created, and $N_{ID}(\Omega)$ is incremented to 2. U and TN are incremented as well, since the sample was unknown, and the ω_1 yielded the correct result (Figure 4 (c)).

The next sample has true class value of 5 and is (incorrectly) positively

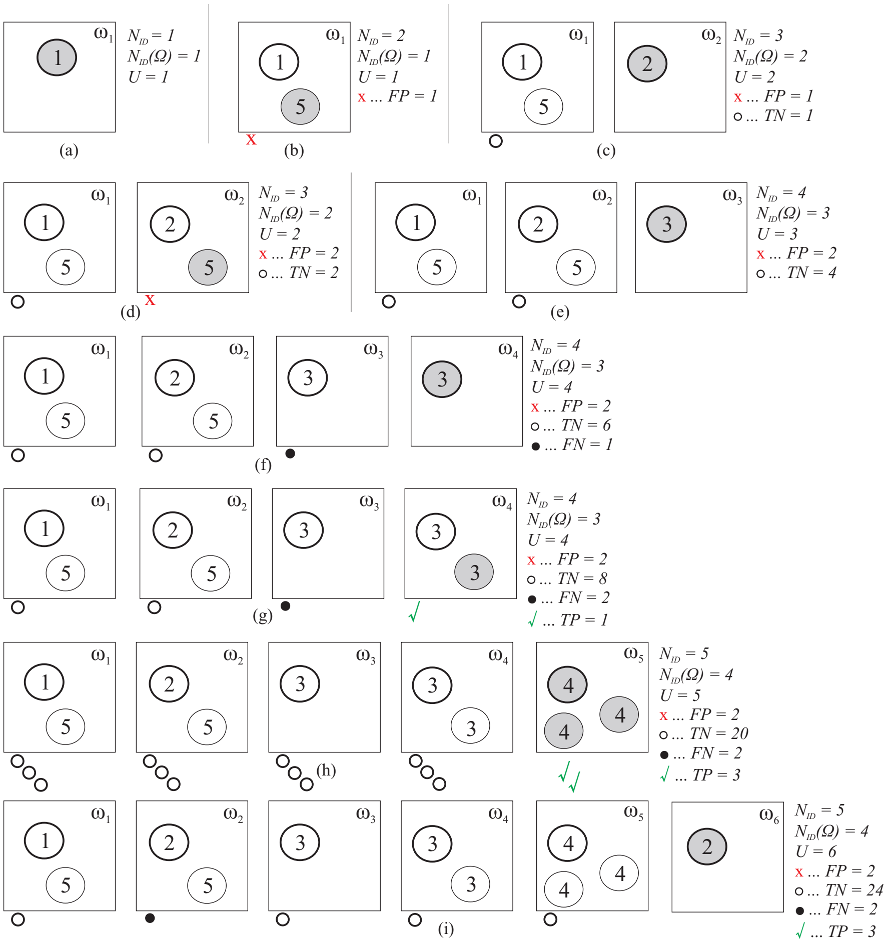


Figure 4: Step-by-step illustration of the first scenario. The situation after the last step is depicted in the Figure 5 (a). Steps of the example are denoted (a) through (i). Each of the boxes at each step represents the state of a single classifier ω_i. Gray circles represent newly arrived samples at the each step. Circles with thicker border represent gallery samples, which created each new classifier ω_i. White circles with thin border represent samples from the previous steps, which have been already classified. The numbers in circles denote the true identity of the samples. Note that "true" identity that the classifier represents is not determined by its index, but with the identity of its gallery sample. Recognition results at each step are denoted with the set of four symbols, that correspond to FP, FN, TP and TN. The statistics on the right hand side are cumulative. The high number of TN illustrates the fact that each classifier is forced to provide a decision about all the newly arrived samples.

recognized by ω_2 (FP is incremented) and (correctly) recognized as negative by ω_1 (TN is incremented). The situation is depicted in Figure 4 (d). The next sample has true class value of 3 (depicted in Figure 4 (e)) and is correctly declared as unknown and yields the new classifier ω_3 (U, $N_{ID}(\Omega)$ and TN are incremented accordingly). The next sample has the true class value of 3 and is correctly rejected by ω_1 and ω_2 (TN is incremented accordingly), but incorrectly rejected by ω_3, therefore a new classifier ω_4 is created. In this case, FN and U are incremented, but $N_{ID}(\Omega)$ is not – even with the classifier ω_4, the system represents only three unique classes, effectively, ω_4 is redundant and contributes to identity fragmentation. This case is shown in Figure 4 (f). A new sample, with true class value of 3 is correctly rejected by ω_1 and ω_2 (TN is incremented), incorrectly rejected by ω_3 (FP is incremented) and correctly recognized (positively classified) by ω_4 and therefore, TP is incremented (Figure 4 (g)). The next three samples have true identity of 4, the first one creates a new classifier ω_5 and the two that follow are correctly classified by ω_5 and rejected by other classifiers. TN, U, $N_{ID}(\Omega)$, and TP are incremented accordingly. The situation is shown in Figure 4 (h).

The next two samples have true class value of 2. The first one is incorrectly rejected by ω_2 and correctly rejected by other classifiers, creating ω_6, incrementing U, TN and FN, but not $N_{ID}(\Omega)$ and resulting in situation shown in Figure 4 (i). The next one is (correctly) recognized by ω_6 and rejected by classifiers ω_1, ω_3, ω_4, ω_5, but (incorrectly) rejected by ω_2 – the outcome is shown in Figure 5 (a). In this case, TP, TN and FN are incremented. Note that the final value of $L = 6$ (we have 6 classifiers), $N_{ID}(\Omega) = 4$ (these classifiers model four unique classes), and $N_{ID} = 5$ (we have shown the system samples from five unique classes). The outcome along with the final values of the evaluation measures are shown in Figure 5 (a).

Figure 5 (b) and Figure 5 (c) show results of degenerate scenarios. In scenario (b) all samples are below the threshold, and therefore, after the first sample creates ω_1, all the others are accepted as being in the same class, either correctly or incorrectly. Note that in this case, the poor performance is shown through a low value of classifier coverage, C – the system does not model the majority of the classes, resulting in low recognition performance (high FPR). In the scenario (c), four samples with a common identity are shown to the system, and are recognized by none of the classifiers created along the way. This results in a high number of false negatives (FN), but more importantly, it also results in a very high identity fragmentation F_{ID}.

4.6. Experimental methodology

We performed experiments in the following manner. First, a matrix of pairwise feature distances $d(\mathbf{x}_i, \mathbf{x}_j), i, j \in 1, \dots, N$ for each of the datasets was computed using a segmented color histogram descriptor from Section 4.4. In the case of Dana36, features were pre-calculated from single images, cropped by the bounding box, and in the case of SAIVT-SoftBio, features were pre-calculated from image sequences, which were cropped according to the provided bounding boxes.

Binary classification threshold T, common to all future classifiers ω_i, is chosen. The system is initialized with an empty set of classifiers $\Omega = \emptyset$. Then a sample-by-sample test run is performed, as shown in Algorithm 1.

Note that all statistics are gathered in each step, therefore measures, such as FPR, TPR and others, become functions of step t, $FPR = FPR(t)$, $TPR = TPR(t)$, etc. In the case of Dana36, our test run contains a random

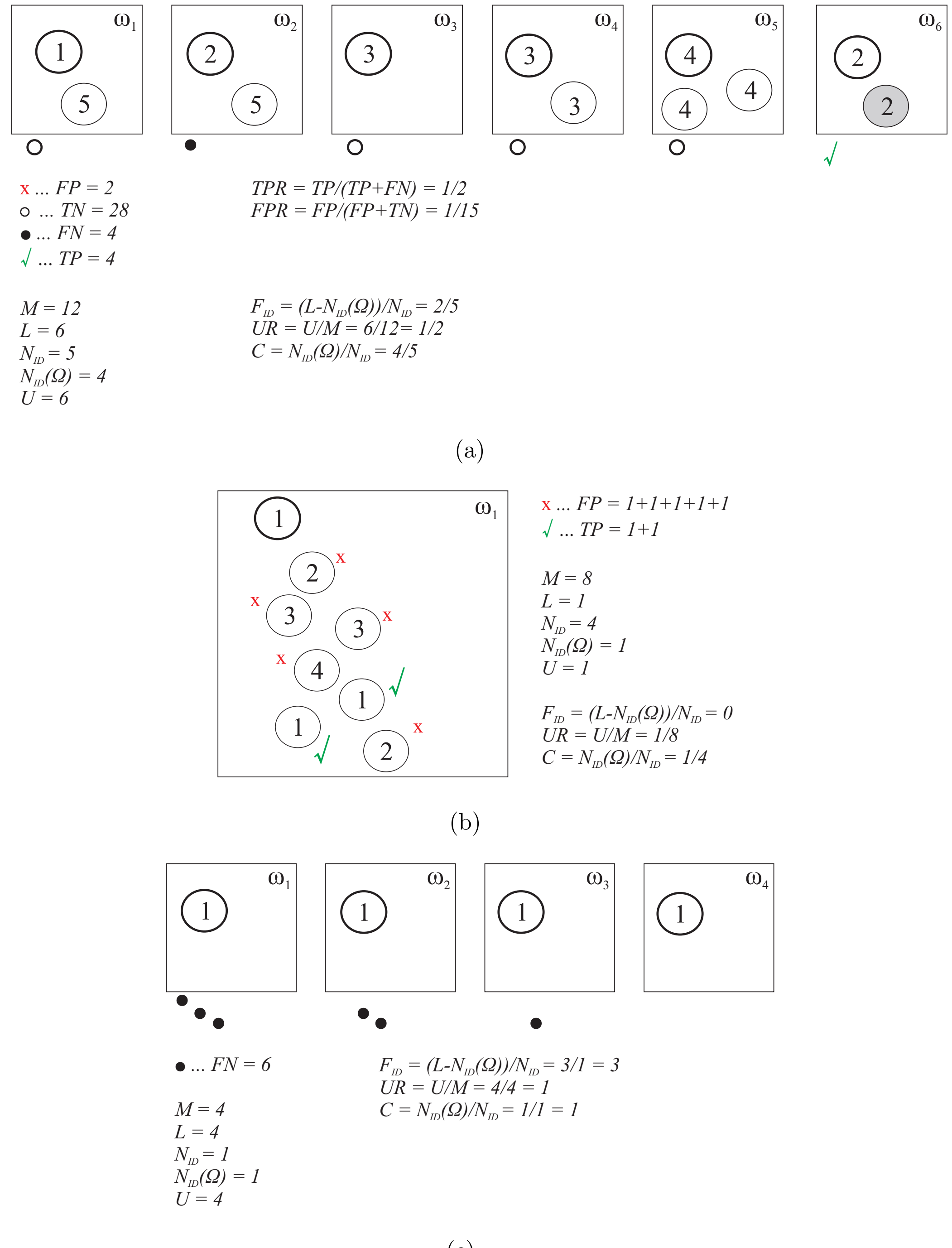


Figure 5: Final outcomes of the three example scenarios, which illustrate the behaviour of the proposed performance measures. (a) general scenario, (b) and (c) degenerate scenarios.

component (sampling of feature vectors $\mathbf{x}$). Therefore we repeat the test multiple times, and calculate mean and standard deviation for all of the observed measures. Standard deviation is in this context a measure of system stability – large standard deviation would indicate that system performance heavily depends on the actual sequence of input samples. In the case of SAIVT-SoftBio, the sequence of observations, as provided by the dataset itself, was used, and the test was performed without repetition.

4.7. Extension to online updating – multiple gallery samples per classifier

So far, we always assumed that there is exactly one gallery feature vector $\mathbf{x}_i$ per classifier ω_i. However, this concept can be extended with the possibility that each classifier contains multiple gallery vectors $\mathbf{x}_{ik}$ that model variations in the class it represents ($k = 1, \ldots, K_i$, K_i is the number of vectors in the classifier ω_i). There are many possibilities of implementing such functionality, however, to stay within the constraints of the distributed camera network, the behaviour of the classifier ω_i towards the network must remain exactly the same.

Classification. Internally, classifier ω_i calculates multiple (that is, K_i) distances $d_{ik} = d(\mathbf{x}_{ik}, \mathbf{x})$ to the observed vector $\mathbf{x}$, one for each of the stored vectors $\mathbf{x}_{ik}$, and d_i is assigned the smallest of the distances d_{ik}. Then, classification rule (3) is applied to the obtained d_i and the decision whether $\mathbf{x}$ is positive or negative is made.

Online updating. If a result of classification is positive, the distance d_i is checked against two learning thresholds, T_{inner} and T_{outer}. If it lies between them, $T_{inner} < d_i < T_{outer}$, then the newly recognized sample $\mathbf{x}$ is added to the classifier ω_i's gallery and K_i is incremented by one. T_{inner} prevents the

Algorithm 1 : Evaluation (full run)

```
Input: A set of all available samples $\mathcal{X}$
Input: Maximum lifespan of a classifier ($\tau_{max}$)
Input: Maximum node capacity $\lambda_{max}$
Output: Evaluation measures
Initialize empty set of classifiers $\Omega = \emptyset$.
Initialize $M \leftarrow 0; L \leftarrow 0; \lambda_j \leftarrow 0; U \leftarrow 0; N_{ID} \leftarrow 0; N_{ID}(\Omega) \leftarrow 0$.
Initialize $FP \leftarrow 0; FN \leftarrow 0; TP \leftarrow 0; TN \leftarrow 0$.
for each step do
    $M \leftarrow M + 1$
    Randomly draw $\mathbf{x} \in \mathcal{X}$ without repetition ($Dana36_{13}$) or select next $\mathbf{x}$ (SAIVT-SoftBio).
    Node number $j$ is determined by drawn $\mathbf{x}$.
    if $\mathbf{x}$ represents truly novel (previously unseen) identity then
        $N_{ID} \leftarrow N_{ID} + 1$
    end if
    $matched \leftarrow 0$
    $multclass\_dist \leftarrow \infty$; $multclass\_id \leftarrow \varnothing$
    for all $\omega_i$, $i \in \{1, \ldots, L\}$ do
        Obtain distance $d(\mathbf{x}, \mathbf{x}_i)$.
        Obtain decision $y_i$ according to the classification rule (3).
        if $y_i > 0$ then
            $matched \leftarrow 1$
            Refresh: $\tau_i \leftarrow \tau_{max}$
```

```

      **if** sample $\mathbf{x}$ has same identity as classifier $\omega_i$ **then**
        $TP \leftarrow TP + 1$
      **else**
        $FP \leftarrow FP + 1$
      **end if**
      **if** $d(\mathbf{x}, \mathbf{x}_i) < multclass\_dist$ **then**
        $multclass\_id \leftarrow$ identity of classifier $w_i$.
        $multclass\_dist \leftarrow d(\mathbf{x}, \mathbf{x}_i)$.
      **end if**
    **else**
      **if** sample $\mathbf{x}$ has same identity as classifier $\omega_i$ **then**
        $FN \leftarrow FN + 1$
      **else**
        $TN \leftarrow TN + 1$
      **end if**
    **end if**
  **end for** // Multi-class evaluation
  **if** $multiclass\_id \neq \varnothing$ **then**
    Increment appropriate field in confusion matrix, based on $multiclass\_id$ and real identity of sample $\mathbf{x}$.
  **end if**
     // Create new classifier, if necessary
  **if** $matched \neq 1$ **then**
    Create a new classifier $\omega_{L+1}(\mathbf{x})$. Initialize $\tau_{L+1} \leftarrow \tau_{max}$.
```

```

        Increase counter of classifiers for node $j$: $\lambda_j \leftarrow \lambda_j + 1$
        $U \leftarrow U + 1; L \leftarrow L + 1$
        **if** sample $\mathbf{x}$ is truly unknown to the system **then**
            $N_{ID}(\Omega) \leftarrow N_{ID}(\Omega) + 1$
        **end if**
    **end if** // Aging and time-to-live-based pruning
    **for all** $\omega_i$, $i \in \{1, \dots, L\}$ **do**
        $\tau_i \leftarrow \tau_i - 1$
        **if** $\tau_i < 0$ **then**
            Remove the classifier $\omega_i$ from the set $\Omega$.
            **if** $\omega_i$ uniquely represented an identity **then**
                $N_{ID}(\Omega) \leftarrow N_{ID}(\Omega) - 1$
            **end if**
        **end if**
    **end for**
          // Limited-capacity-based pruning
    **if** $\lambda_j > \lambda_{max}$ **then**
        Remove $\omega_i$ from $\Omega$, where $i = \underset{k}{\operatorname{argmin}}\, \tau_k$
        **if** $\omega_i$ uniquely represented an identity **then**
            $N_{ID}(\Omega) \leftarrow N_{ID}(\Omega) - 1$
        **end if**
    **end if**
**end for**
```

learning of the samples that are very close to existing samples in the ω_i's gallery and would waste resources; on the other hand, T_{outer} determines the maximum allowed degree of adaptation of the classifier in a single step.

Forgetting. Instead per-classifier, the forgetting is re-formulated to operate on per-vector basis – when all of the classifier's gallery feature vectors are forgotten, the classifier itself is removed from the classifier set.

Since the behaviour of each single classifier towards the network remains the same, all evaluation measures remain valid even for a case with multiple gallery samples.

5. Experiments and results

Experiments have been designed to examine the following:

- The behavior of the proposed approach during the test run. In particular, we are interested in the dynamics of the evaluation measures.
- The stability of the proposed method, expressed as standard deviation of evaluation measures among the multiple test runs.
- The effects of parameter variation – we varied T, τ_{max} and T_{outer}, one at a time, with all other parameters fixed.

Unless specified otherwise, the experiments were performed with the following settings: based on our preliminary research, we set the threshold T to $T_{13} = 0.5$. At this threshold, false positive rate on Dana36 was estimated to be below 20 %. Online updating of samples was disabled ($T_{inner} = T_{outer} = 1$), except in the last experiment, when the influence of T_{outer} was examined.

Each *run* consisted of 2000 steps on Dana36 and 788[6] steps on SAIVT-SoftBio. Our implicit assumption is that new images or image sequences arrive in constant time intervals, therefore, in the rest of the paper, we equate the step number with time. To estimate standard deviation of observed measures, each test on Dana36 consisted of 10 runs – in this case, each sample is drawn uniformly without replacement. When comparing multiple tests (e.g., to determine the influence of parameters), a sequence of pseudo-random numbers is restarted between the tests, to provide consistent results.

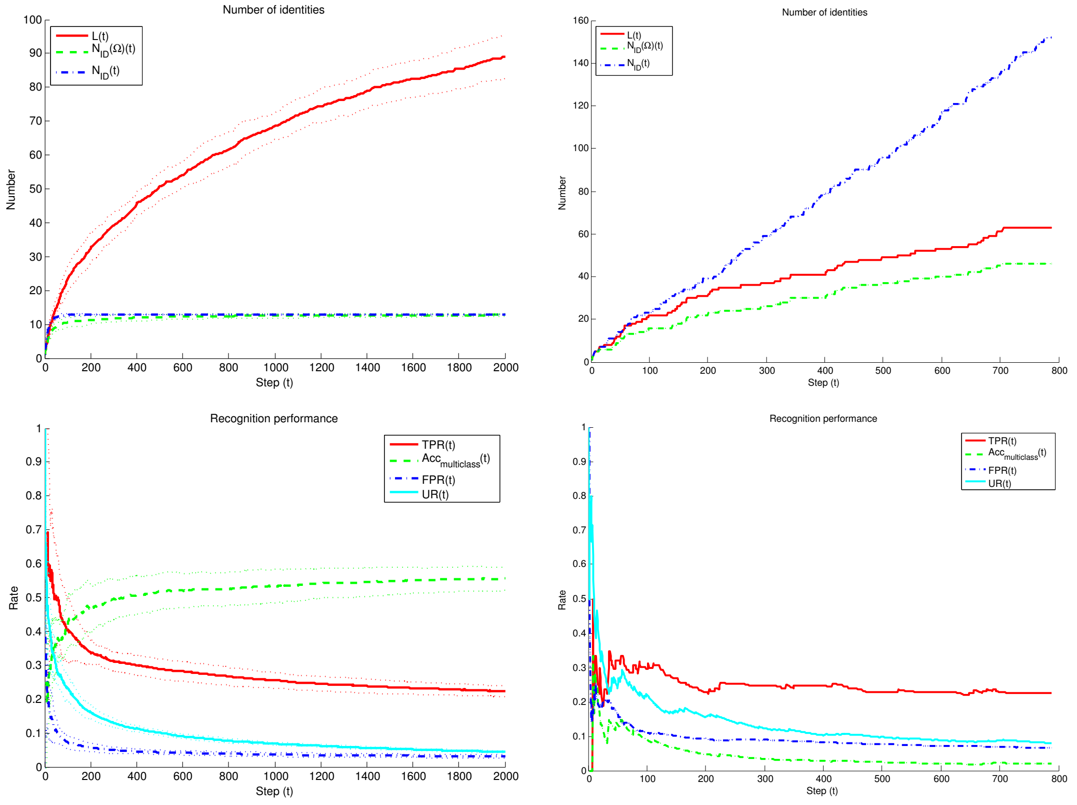
5.1. Performance without forgetting

In this case, τ_{max} and λ_{max} are set to infinity, which means that no classifiers expired during the test run. This simulates the system that retains all the acquired information.

The results for both subsets are shown in Figure 6 and summarized in Table 1.

As seen in Figure 6, at the beginning of the run, there is intensive learning, indicated by high values of unknown rate UR, and steeper slope for number of classifiers L. Later, the increase in L is more gradual. The number of *unique* identities represented by classifier set Ω, $N_{ID}(\mathit{\Omega})$ rises, but with the parameters selected, it does not reach the true number of identities in the observed data, N_{ID}. This also results in the final classifier coverage C being below one. For both datasets, after the initial instability, TPR slowly falls until the end of experiment. The multi-class accuracy measure $Acc_{multiclass}$ for SAIVT-SoftBio may seem low at first glance, but one should remember

[6] This is the actual number of samples in the SAIVT Soft-Bio dataset.



Dana36$_{13}$ SAIVT-SoftBio

Figure 6: Performance (number of identities and recognition performance) as the function of time (t), for the proposed approach *without* forgetting, for the two datasets. Dotted lines show standard deviation for Dana36$_{13}$.

that recognizing identities of 152 persons is a very difficult problem – a random classifier would yield $Acc_{multiclass}$ of only $1/152 = 0.0066$.

5.2. Performance with forgetting

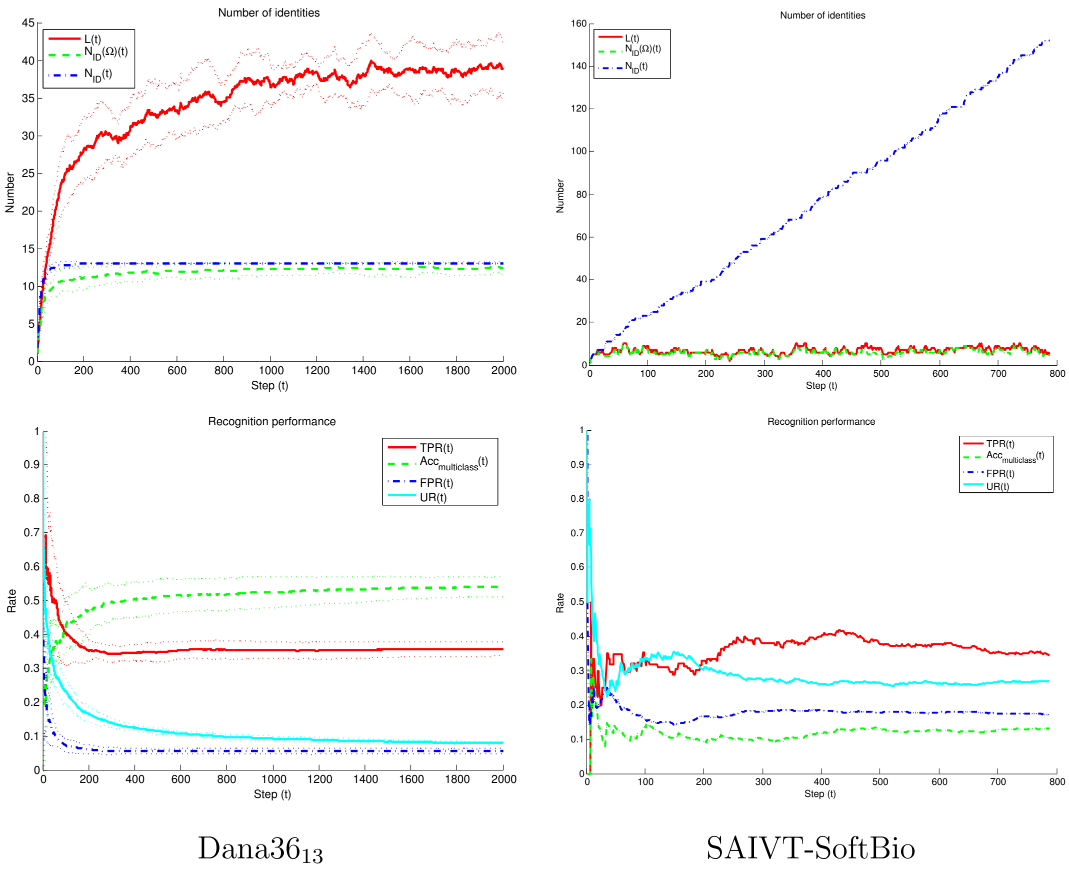
In the next two experiments, we examined the performance of a system with forgetting. Two extreme operating points were selected. In the first one, we enforced the *limited lifespan*, with $\tau_{max} = 100$ steps for Dana36 and $\tau_{max} = 10$ for SAIVT-SoftBio. In the second part of the experiment, we enforced *limited capacity*, with $\lambda_{max} = 2$ classifiers per node for both datasets.

5.2.1. Limited lifespan

Results for this case are shown in Figure 7 and in Table 1. In comparison to the approach without forgetting, the graphs show that the limited-lifespan-based forgetting scheme introduces some instability into the number of classifiers L on Dana36 dataset. At the beginning, the system learns quickly and levels off as the classifiers start to expire. Due to quick learning at the beginning, several classifiers expire approximately at the same time. This effect is visible as lower rise and then drop-off in L. The other effect of such forgetting scheme is well visible in results for SAIVT-SoftBio – much smaller number of classifiers L, and correspondingly smaller classifier coverage (C), but also significant drop in identity fragmentation (F_{ID}). This shows an important tradeoff of such forgetting scheme – identity fragmentation can be decreased, but at the cost of lower coverage and possibly less stable L. Finally, multi-class accuracy $Acc_{multiclass}$ for SAIVT-SoftBio actually rises and displays slightly upwards trend due to benefits of forgetting on such realistic scenario. The system quickly forgets people which are not seen at any later moment, thus improving its odds at classifying people that actually appear.

5.2.2. Limited capacity

Results for limited capacity are shown in Figure 8. Compared to limited lifespan, it can be seen that in this operating point there is no instability in L for Dana36, but increase in multi-class accuracy $Acc_{multiclass}$ for SAIVT-SoftBio is still visible. The final results are shown in Table 1.

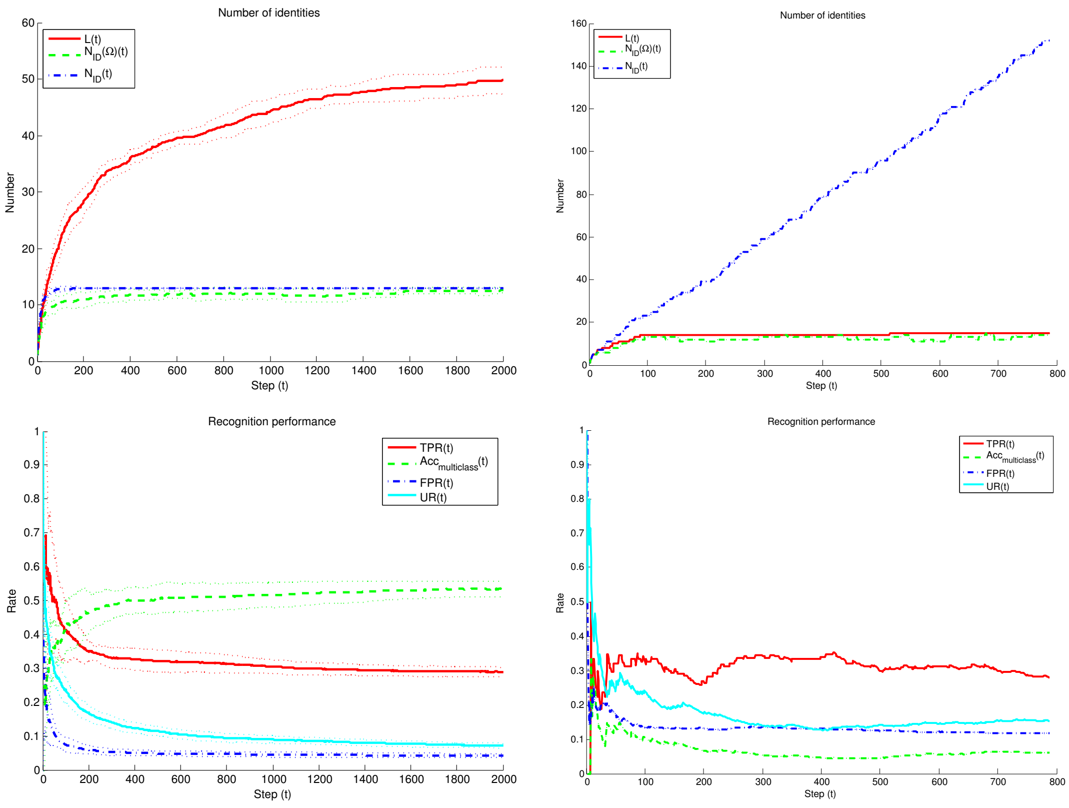


Dana36_{13} SAIVT-SoftBio

Figure 7: Number of identities and recognition performance as a function of time (t), for the proposed approach *with* forgetting, based on the *limited lifespan*, for the two datasets. Dotted lines show standard deviation for Dana36_{13}.

5.3. Parameter influence

Previous experiments were performed with fixed parameter values, chosen to illustrate the dynamics of the proposed system during its operation. Nevertheless, parameters have significant influence on the performance and stability. We explored the influence of classification threshold T, which controls the operating point of classifier set Ω, the influence of maximum time-



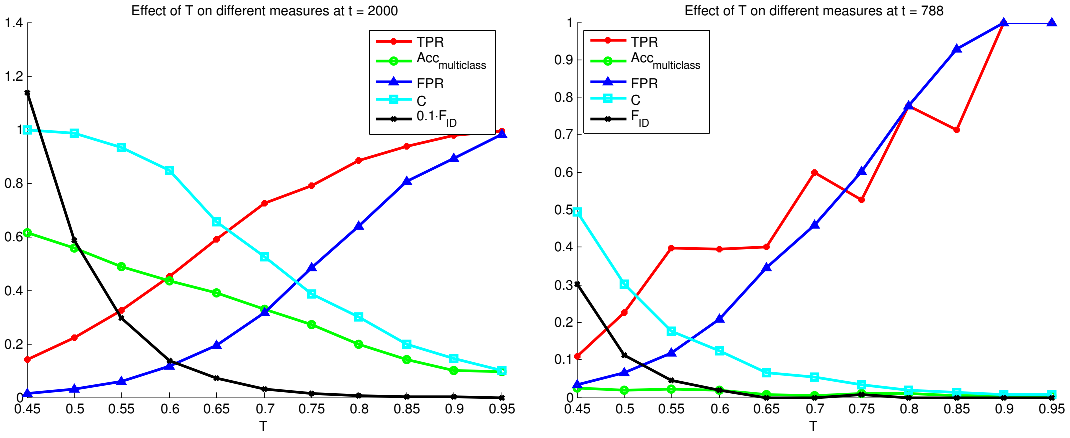
Dana36$_{13}$ SAIVT-SoftBio

Figure 8: Performance (number of identities and recognition performance) as the function of time (t), for the proposed approach *with* forgetting, based on the *limited capacity*, for the two datasets. Dotted lines show standard deviation for Dana36$_{13}$.

to-live parameter τ_{max} when limited lifespan is enforced, and the influence of learning threshold T_{outer}, which controls the degree of updating. We varied only single parameter and fixed the rest to values from the beginning of the Section 5. When examining the influence of T and T_{outer}, forgetting was disabled ($\tau_{max} = \infty$, $\lambda_{max} = \infty$). Results for T and τ_{max} are shown in Figures 9 and 10, respectively.

Table 1: Mean values and standard deviations of evaluation measures at the end of each run ($t = 2000$ for Dana36 and $t = 788$ for SAIVT-SB). LL denotes *limited lifespan*, and LC denotes *limited capacity*. TPR is the true positive rate, $Acc_{multiclass}$ is the multi-class accuracy, C is classifier coverage and F_{ID} is identity fragmentation. Note that a random classifier would yield $Acc_{multiclass}$ of only $1/152 = 0.0066$ on SAIVT-SoftBio, therefore, $Acc_{multiclass}$ of 0.13 represents significant improvement.

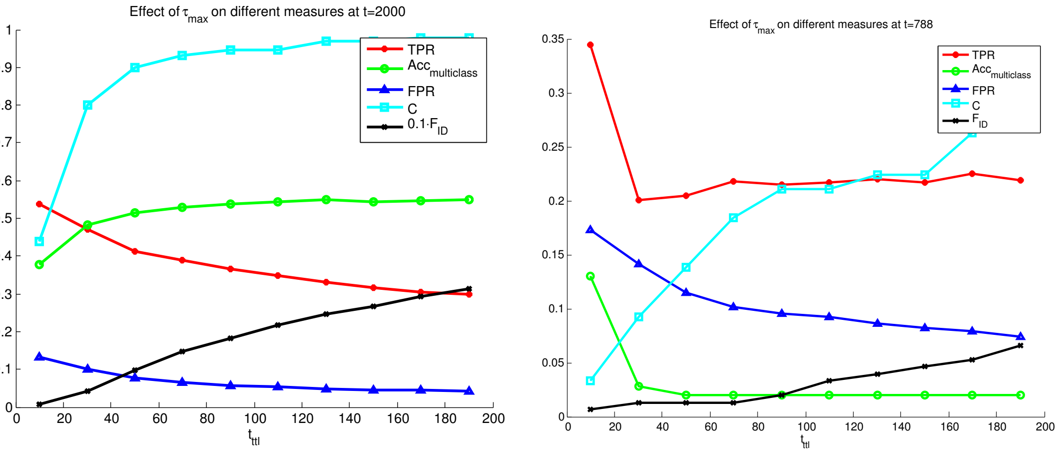
Cases	Datasets	Evaluation measures				
		TPR	$Precision$	$Acc_{multiclass}$	C	F_{ID}
No forgetting	Dana36$_{13}$	0.22 ± 0.02	0.39 ± 0.03	0.56 ± 0.03	0.98 ± 0.03	5.87 ± 0.52
	SAIVT-SoftBio	0.23	0.01	0.02	0.30	0.11
Forgetting – LL	Dana36$_{13}$	0.36 ± 0.02	0.37 ± 0.03	0.54 ± 0.03	0.95 ± 0.06	2.05 ± 0.24
	SAIVT-SoftBio	0.34	0.11	0.13	0.03	0.01
Forgetting – LC	Dana36$_{13}$	0.29 ± 0.01	0.38 ± 0.03	0.53 ± 0.02	0.97 ± 0.04	2.86 ± 0.18
	SAIVT-SoftBio	0.28	0.05	0.06	0.09	0.01



Dana36$_{13}$ SAIVT-SoftBio

Figure 9: Influence of the classification threshold T on the system performance for both datasets. Identity fragmentation F_{ID} is drawn in different scales.

It can be seen that T and τ_{max} influence the behavior of the system in similar ways, despite the significant differences in nature of the data (Dana36 is image-based and samples were drawn randomly, SAIVT-SoftBio



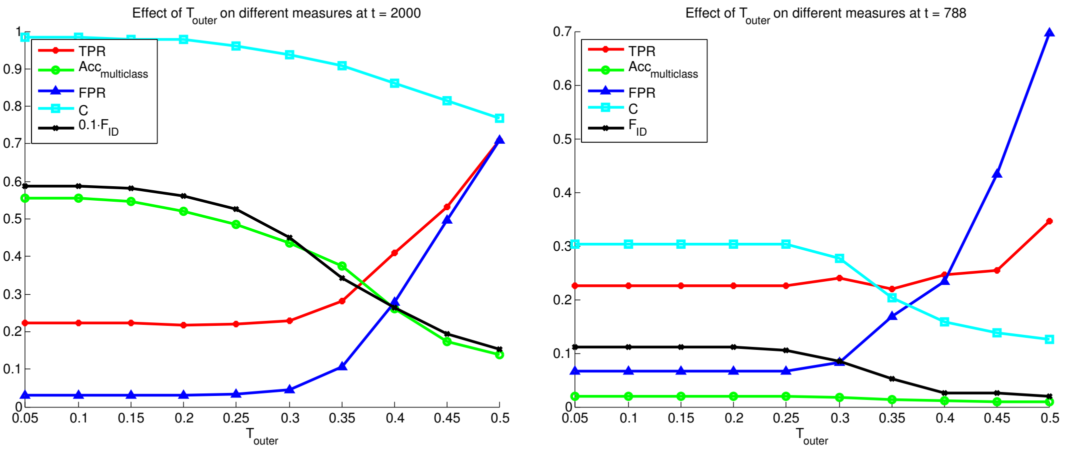
Dana36_{13} SAIVT-SoftBio

Figure 10: Influence of the forgetting parameter τ_{max} on the system performance for both datasets. Identity fragmentation F_{ID} is drawn in different scales.

is image-sequence-based with predetermined sequence of events). Increasing the threshold T increases the number of true and false positives, while the classifier coverage C drops, along with identity fragmentation F_{ID}. This further confirms that there is tradeoff between higher C and lower F_{ID}. Similarly, lower τ_{max} reduces identity fragmentation, but also lowers the classifier coverage C. It is also obvious that both T and τ_{max} influence the multiclass accuracy $Acc_{multiclass}$ – in the case of SAIVT-SoftBio dataset, intensive forgetting (low τ_{max}) actually increases $Acc_{multiclass}$.

In the last batch of experiments, shown in Figure 11, we examined the influence of online updating threshold T_{outer}. Based on our preliminary experiments, the value of T_{inner} was set to a value of 0.05, and the value of T_{outer} was varied between 0.05 and 0.5 (the latter is the chosen value of the classification threshold T), resulting in more (high T_{outer}) or less (low T_{outer})

aggressive updating.



Dana36_{13}

SAIVT-SoftBio

Figure 11: Influence of the learning threshold T_{outer} on the system performance for both datasets. The point $T_{outer} = T_{inner} = 0.05$ is equivalent to online updating disabled. Identity fragmentation F_{ID} is drawn in different scales.

It can be seen that in this case, the accuracy $Acc_{multiclass}$ does not improve with online updating – it even decreases with Dana36_{13}. However, sufficiently aggressive updating does reduce identity fragmentation F_{ID}, which is a tangible benefit. With online updating, individual classifiers are able to model variations in data, therefore fewer new classifiers are created.

All presented results clearly demonstrate that in a distributed re-identification system that works on the realistic, open-world problem, there are many interdependencies among various aspects of the system performance – the recognition performance alone does not tell the whole story.

6. Conclusion

In this paper, we addressed re-identification problem in large, distributed camera networks – a topic that has been under-represented in the research so far. We formalized object re-identification problem in a distributed environment, and analyzed it as an open-world problem. We documented the obstacles that are *inherent* to truly distributed surveillance systems. These preclude the direct use of state-of-the-art algorithms, and demand that functionality of the system is built using only the limited resources available in a distributed environment. Assuming only this basic functionality, we built a scheme for distributed re-identification, which provides on-line learning, forgetting and novelty detection – critical components for addressing *open world problems.* Performance analysis in such distributed system requires more than just observing classification performance. Therefore, we proposed a set of measures geared towards distributed surveillance. We demonstrated that in such a system, we deal with multiple tradeoffs – even in the case of the simplest classification algorithm with a single parameter, the operating point of the classifier set influences several aspects of the system, not just classification rates. This interdependence may radically alter the overall performance of the re-identification system.

It should be noted that the presented method for constructing distributed re-identification system is generic. Even though we limited ourselves to color histograms, the method could be applied to more sophisticated cases where complex features are extracted from images or image sequences, or even from locally-connected multi-view camera systems, which can still represent single node in a large, distributed camera network.

7. Acknowledgements

This work was supported by research programs P2-0095 and P2-0098, research project J2-4284 and the research grant 1000-10-310118, all by Slovenian Research Agency. We would like to thank the SAIVT Research Labs at Queensland University of Technology (QUT) for freely supplying us with the SAIVT-SoftBio database for our research.

References

[1] M. Valera, S. A. Velastin, Intelligent distributed surveillance systems: a review, IEE Proceedings - Vision, Image and Signal Processing 152 (2005) 192–204.

[2] D. Gray, H. Tao, Viewpoint invariant pedestrian recognition with an ensemble of localized features, in: European Conference on Computer Vision (ECCV'08), pp. 262–275.

[3] M. Hirzer, C. Beleznai, P. Roth, H. Bischof, Person re-identification by descriptive and discriminative classification, in: Proceedings of Scandinavian Conference on Image Analysis (SCIA'11), pp. 91–102.

[4] S. Bak, E. Corvee, F. Bremond, M. Thonnat, Boosted human re-identification using riemannian manifolds, Image and Vision Computing 30 (2012) 443–452.

[5] X. Wang, Intelligent multi-camera video surveillance: A review, Pattern Recognition Letters 34 (2013) 3–19.

[6] R. Vezzani, D. Baltieri, R. Cucchiara, People reidentification in surveillance and forensics: A survey, ACM Comput. Surv. 46 (2013) 29:1–29:37.

[7] B. Murovec, J. Perš, R. Mandeljc, V. Sulić Kenk, S. Kovačič, Towards commoditized smart-camera design, Journal of Systems Architecture 59 (2013) 847–858.

[8] J. Perš, V. Sulić, R. Mandeljc, M. Kristan, S. Kovačič, Dana36: A multi-camera image dataset for object identification in surveillance scenarios, in: 9th IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS'12), pp. 64–69.

[9] A. Bialkowski, S. Denman, P. Lucey, S. Sridharan, C. B. Fookes, A database for person re-identification in multi-camera surveillance networks, in: Digital Image Computing : Techniques and Applications (DICTA 2012), IEEE, 2012, pp. 1–8.

[10] G. Doretto, T. Sebastian, P. Tu, J. Rittscher, Appearance-based person reidentification in camera networks: problem overview and current approaches, Journal of Ambient Intelligence and Humanized Computing 2 (2011) 127–151.

[11] O. Javed, K. Shafique, M. Shah, Appearance modeling for tracking in multiple non-overlapping cameras, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), volume 2, pp. 26–33.

[12] B. Prosser, S. Gong, T. Xiang, Multi-camera matching under illumination change over time, in: ECCV Workshop on Multi-camera and

Multi-modal Sensor Fusion Algorithms and Applications (ECCVW'08), pp. 1–12.

[13] M. Andriluka, S. Roth, B. Schiele, People-tracking-by-detection and people-detection-by-tracking, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08), pp. 1–8.

[14] Z. Lin, L. S. Davis, Learning pairwise dissimilarity profiles for appearance recognition in visual surveillance, in: International Symposium of Visual Computing, pp. 23–34.

[15] W. Zheng, S. Gong, T. Xiang, Transfer re-identification: From person to set-based verification, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12).

[16] X. Liu, M. Song, Q. Zhao, D. Tao, C. Chen, J. Bu, Attribute-restricted latent topic model for person re-identification, Pattern recognition 45 (2012) 4204–4213.

[17] N. Gheissari, T. Sebastian, R. Hartley, Person reidentification using spatiotemporal appearance, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), volume 2, pp. 1528–1535.

[18] X. Wang, G. Doretto, T. Sebastian, J. Rittscher, P. Tu, Shape and appearance context modeling, in: IEEE 11th International Conference on Computer Vision (ICCV'07), pp. 1–8.

[19] A. K. Park, U.and Jain, I. Kitahara, K. Kogure, N. Hagita, Vise: Visual search engine using multiple networked cameras, in: International Con-

ference on Pattern Recognition (ICPR), IEEE Computer Society, 2006, pp. 1204–1207.

[20] C. Siebler, K. Bernardin, R. Stiefelhagen, Adaptive color transformation for person re-identification in camera networks, in: Proceedings of the Fourth ACM/IEEE International Conference on Distributed Smart Cameras, ICDSC '10, pp. 199–205.

[21] C.-T. Chu, J.-N. Hwang, K.-M. Lan, S.-Z. Wang, Tracking across multiple cameras with overlapping views based on brightness and tangent transfer functions, in: Distributed Smart Cameras (ICDSC), 2011 Fifth ACM/IEEE International Conference on, pp. 1–6.

[22] L. Lo Presti, S. Sclaroff, M. La Cascia, Object matching in distributed video surveillance systems by lda-based appearance descriptors, in: Proceedings of International Conference on Image Analysis and Processing (ICIAP'09), pp. 1–11.

[23] S. Bak, E. Corvee, F. Bremond, M. Thonnat, Person re-identification using haar-based and dcd-based signature, in: Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS'10), pp. 1–8.

[24] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, Surf: Speeded up robust features, Computer Vision and Image Understanding 110 (2008) 346–359.

[25] O. Hamdoun, F. Moutarde, B. Stanciulescu, B. Steux, Person re-identification in multi-camera system by signature based on interest

point descriptors collected on short video sequences, in: Proceedings of the IEEE Conference on Distributed Smart Cameras (ICDSC'08), pp. 1–6.

[26] I. O. de Oliveira, J. L. de Souza Pio, People reidentification in a camera network, in: Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC'09), pp. 461–466.

[27] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2004) 91–110.

[28] K. Jungling, C. Bodensteiner, M. Arens, Person re-identification in multi-camera networks, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'11), pp. 55–61.

[29] S. Bak, E. Corvee, F. Bremond, M. Thonnat, Person re-identification using spatial covariance regions of human body parts, in: Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS'10), pp. 435–440.

[30] F. Tuzel, O. amd Porikli, P. Meer, Region covariance: A fast descriptor for detection and classification, in: European Conference on Computer Vision (ECCV'06), pp. 589–600.

[31] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), volume 1, pp. 886–893.

[32] M. Farenzena, L. Bazzani, A. Perina, V. Murino, M. Cristani, Person re-identification by symmetry-driven accumulation of local features, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10), pp. 2360–2367.

[33] M. Bauml, R. Stiefelhagen, Evaluation of local features for person re-identification in image sequences, in: 8th IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS'11), pp. 291 –296.

[34] W.-S. Zheng, S. Gong, T. Xiang, Person re-identification by probabilistic relative distance comparison, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11), pp. 649–656.

[35] D. Blei, A. Ng, M. Jordan, Latent dirichlet allocation, Journal of Machine Learning Research (2003) 993—-1022.

[36] N. Martinel, C. Micheloni, C. Piciarelli, Distributed signature fusion for person re-identification, in: Distributed Smart Cameras (ICDSC), 2012 Sixth International Conference on, pp. 1–6.

[37] V. Jelaca, A. Pizurica, J. O. Nino-Castaneda, A. Frias-Velazquez, W. Philips, Vehicle matching in smart camera networks using image projection profiles at multiple instances, Image and Vision Computing (2013) –. Accepted manuscript.

[38] V. Sulić, J. Perš, M. Kristan, S. Kovačič, Efficient feature distribution for object matching in visual-sensor networks, IEEE Transactions on Circuits and Systems for Video Technology 21 (2011) 903–916.

[39] M. Markou, S. Singh, Novelty detection: a review–part 1: statistical approaches, Signal Processing 83 (2003) 2481–2497.

[40] M. Breitenstein, H. Grabner, L. Van Gool, Hunting nessie – real-time abnormality detection from webcams, in: ICCV'09 WS on Visual Surveillance.

[41] S.-P. Yong, J. D. Deng, M. K. Purvis, Novelty detection in wildlife scenes through semantic context modelling, Pattern Recognition 45 (2012) 3439 – 3450.

[42] P. Angelov, P. Sadeghi-Tehran, R. Ramezani, An approach to automatic real-time novelty detection, object identification, and tracking in video streams based on recursive density estimation and evolving takagi–sugeno fuzzy systems, International Journal of Intelligent Systems 26 (2011) 189–205.

[43] F. Tung, J. S. Zelek, D. A. Clausi, Goal-based trajectory analysis for unusual behaviour detection in intelligent surveillance, Image and Vision Computing 29 (2011) 230–240.

[44] P. Sadeghi-Tehran, P. Angelov, A real-time approach for novelty detection and trajectories analysis for anomaly recognition in video surveillance systems, in: IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), 2012, pp. 108 –113.

[45] C. P. Diehl, J. B. Hampshire II, Real-time object classification and novelty detection for collaborative video surveillance, in: International Joint Conference on Neural Networks 2002, pp. 2620–2625.

[46] J. Owens, A. Hunter, E. Fletcher, Novelty detection in video surveillance using hierarchical neural networks, in: International Conference on Artificial Neural Networks (ICANN'02), pp. 1249–1254.

[47] C. C. Loy, T. Xiang, S. Gong, Multi-camera activity correlation analysis, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009), pp. 1988–1995.

[48] D. Baltieri, R. Vezzani, R. Cucchiara, 3dpes: 3d people dataset for surveillance and forensics, in: Proceedings of the 1st International ACM Workshop on Multimedia access to 3D Human Objects, pp. 59–64.

[49] L. Wei, X. Wang, Locally aligned feature transforms across views, in: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pp. 3594–3601.

[50] V. Sulić, J. Perš, M. Kristan, S. Kovačič, Efficient Feature Distribution in Visual Sensor Networks – An Example, Technical Report, Machine Vision Laboratory, Faculty of Electrical Engineering, University of Ljubljana, 2012. http://vision.fe.uni-lj.si/docs/janezp/techReportExample.pdf.

[51] C. D. Manning, H. Schutze, Foundations of Statistical Natural Language Processing, MIT Press, p. 577.

[52] T. Fawcett, An introduction to ROC analysis, Pattern Recognition Letters 27 (2006) 861–874.