

# Multi-class Classification on Riemannian Manifolds for Video Surveillance

Diego Tosato<sup>1</sup>, Michela Farenzena<sup>1</sup>, Marco Cristani<sup>1,2</sup>,  
Mauro Spera<sup>1</sup>, and Vittorio Murino<sup>1,2</sup>

<sup>1</sup> Dipartimento di Informatica, University of Verona, Italy

<sup>2</sup> Istituto Italiano di Tecnologia (IIT), Genova, Italy

**Abstract.** In video surveillance, classification of visual data can be very hard, due to the scarce resolution and the noise characterizing the sensors' data. In this paper, we propose a novel feature, the ARray of CO-variances (ARCO), and a multi-class classification framework operating on Riemannian manifolds. ARCO is composed by a structure of covariance matrices of image features, able to extract information from data at prohibitive low resolutions. The proposed classification framework consists in instantiating a new multi-class boosting method, working on the manifold  $Sym_d^+$  of symmetric positive definite  $d \times d$  (covariance) matrices. As practical applications, we consider different surveillance tasks, such as head pose classification and pedestrian detection, providing novel state-of-the-art performances on standard datasets.

## 1 Introduction

An important goal of automated video surveillance is to design algorithms that can characterize different objects of interest (OIs), especially when immersed in a cluttered background and captured at low resolution. The detection (e.g., of faces or pedestrians) and the classification (e.g., of facial poses) are among the most studied applications. In the multi-faceted plethora of approaches in the literature (see [1,2,3] for extensive reviews), boosting-based techniques play a primary role [4,5,6,7,8,9,10,11,12,13,14]: boosting [15,16,17] is a remarkable, highly customizable way to create strong and fast classifiers, employing various features fed into diverse architectures, with specific policies.

Among the different features considered for boosting in object classification (see [18] for an updated list), covariance features [19] have been exploited as powerful descriptors of pedestrians [11,12,13], and their effectiveness has been explicitly investigated in a comparative study [14]. When injected in boosting systems [11,12,13,14], covariances provide strong detection performances. They encapsulate the high intra-class variances (due to pose and view changes of the OI), they are in general stable in presence of noise, and provide an elegant way to fuse multiple low-level features, as they intrinsically exploit possible inter-features' dependencies. Moreover, thanks to the integral image representation, they can be calculated in a very efficient way.

Since covariance matrices lie in the Riemannian manifold of symmetric positive definite matrices  $Sym_d^+$ , their usage in a boosting framework requires a careful treatment. In [11], the input covariance features are projected into the tangent space at particular points of the manifold, where an Euclidean metric can be instantiated, and the Logitboost framework can be applied.

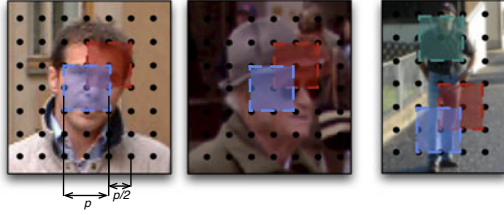
In this paper, we propose two main contributions. First, we present a novel kind of feature, *i.e.*, the *ARray of COvariances* (ARCO), able to describe visual objects at prohibitive low resolutions (up to  $5 \times 5$  pixels): it marries the dense descriptors philosophy, adopted for example in [20], with the expressivity of the covariance information. Second, we show how such features can be embedded in a multi-classification framework by boosting, extending [11] to the multi-class case. We experimentally show that  $Sym_d^+$  has non positive curvature and in the areas where the curvature is almost flat the Euclidean metric on the tangent space at any point on the manifold is a good approximation of the Riemannian metric. Therefore, unlike [11], we map all the data in a unique tangent space, and we perform all the computations on this (Euclidean) space where a typical multi-class LogitBoost algorithm can be applied.

The experimental trials show how we outperform the current methods in two important applications for surveillance like head pose classification and pedestrian detection, without adopting complex boosting schemes such as Floatboosting for pyramids [9], decision trees [6], VectorBoosting for width-first-search trees [7], or Probabilistic Boosting Networks [21]. We fix novel state-of-the-art performances on standard databases. This encourages the embedding of our Riemannian framework in the above quoted boosting schemes. We stress also the capability of dealing with compelling image resolutions, promoting the use of ARCOs for heterogeneous applications, especially in the surveillance field.

The rest of the paper is organized as follows. Sec. 2 describes the proposed ARCO feature, and Sec. 3 depicts the proposed multi-class framework. Sec. 4 shows the experimental results on several surveillance applications, and finally we draw our conclusions in Sec. 5.

## 2 ARCO: ARray of COvariance Matrices

The proposed classification framework has been specifically designed to deal with low resolution images, typical of a video surveillance scenario. In such conditions, the number of features that can be extracted are relatively small, and quite unreliable. This is very challenging in problems like, for instance, head pose classification, in which the details are crucial to distinguish the different object classes. Moreover, the classifier must cope with objects (pedestrians, heads) views in a variety of light conditions. Our solution is based on two main concepts: 1) the organization of the image into a grid of uniformly spaced and overlapping patches (Fig. 1); 2) the use of covariance matrices of image features as patch descriptors, which are classified by multi-class LogitBoost on Riemannian manifolds. In a few words, each patch classifier votes for a class, and the final classification result is the class voted by the majority of them.



**Fig. 1.** Array of Covariance matrices (ARCO) feature. The image is organized as a grid of uniformly spaced and overlapping patches. On each patch, a multi-class classifier is estimated.

In [11], where the use of covariance matrix descriptors is tailored for pedestrian detection, LogitBoost was used for both a greedy estimation of the most discriminative patches among a set of different sizes and positions, and for classifying them, i.e., as feature selection and classification method at the same time. The same reasoning, using boosting for feature selection and classification, has been applied to other approaches in the literature, as for example in [22,23]. Here, instead, a feature selection operation is unfeasible, because low resolution images contain such scarce and noisy information that the result would be unreliable: it is more convenient to use *all* features in a suitable way. Our approach takes inspiration from the literature on dense image descriptors (see [20] as an example). We sample the image  $I$  into uniformly distributed and overlapping patches of the same dimension. Each patch is described by the covariance matrix representation, that encodes the local shape and appearance of the (small) region. We use these patches in a democratic way: we exalt their discriminative power by boosting a strong multi-class classifier, and we collect their classification results.

More formally, given a set of patches  $\{P_i\}_{i=1,\dots,N_P}$ , we learn a multi-class classifier for each patch  $\{F_{P_i}\}_{i=1,\dots,N_P}$  through the multi-class LogitBoost algorithm [17], adapted to work on Riemannian manifolds.

Let  $\Delta_j = \sum_{i=1}^{N_P} (F_{P_i} == j)$  be the number of patches that vote for the class  $j \in \{1, \dots, J\}$ . We assign a class label  $c$  to an image, estimating

$$c = \arg \max_j \{\Delta_j\}, \quad j = 1, \dots, J. \quad (1)$$

In order to increase robustness to local illumination variations, we apply the normalization operator introduced in [11] before applying the multi-class framework.

The ARCO representation has several advantages. First, it allows to take into account different features, inheriting their expressivity, and exploiting possible correlations. In this sense, it is as a compact and powerful integration of features. Second, due to the use of integral images, ARCO is fast to compute, making it suitable for a possible real-time usage. Finally, as a dense representation, it is robust to occlusions. We will prove all the above characteristics during the experimental trials in Sec.4.

### 3 Multi-class Classification on Riemannian Manifolds

Let  $C_1, C_2, \dots, C_J$  be the data classes whose elements (the covariances) live in the Riemannian manifold  $\mathcal{M}$  of  $d \times d$  symmetric positive definite matrices denoted by  $Sym_d^+$ . Let  $\mathcal{S} = \{X_i, y_i\}_{i=1, \dots, N}$  be the set of  $N$  training examples, with  $X_i \in \mathcal{M}$  and label  $y_i \in \{1, \dots, J\}$ . The goal is to produce a function  $F(X_i) : \mathcal{M} \mapsto \{1, \dots, J\}$  as

$$F(X_i) = \arg \max_j \{F_j(X_i)\}, \quad j = 1, \dots, J. \tag{2}$$

$F_j$  is a *single-class* strong classifier, and it is defined, in turn, as a sum of  $L$  weak classifiers  $\{f_{lj}\}_{l=1, \dots, L}$ . These weak classifiers are learned by multi-class LogitBoost.

#### 3.1 Riemannian Geometry on $Sym_d^+$

In this section, we briefly review the geometry of  $Sym_d^+$ , the manifold consisting of all  $d \times d$  symmetric definite positive matrices (covariance matrices), extending the treatment given in [11].

The tangent space  $T_Y$  at any point  $Y \in Sym_d^+$  can be identified with  $Sym_d$ , the (vector) space of  $d \times d$  symmetric matrices.

The mapping of  $X$  on  $T_Y$ , is given by the point-dependent  $\log_Y$  operator:

$$\log_Y(X) = Y^{\frac{1}{2}} \log \left( Y^{-\frac{1}{2}} X Y^{-\frac{1}{2}} \right) Y^{\frac{1}{2}}, \tag{3}$$

inverse to the exponential map.

The (geodesic) distance on  $Sym_d^+$  is defined as

$$d^2(X_1, X_2) = Tr(\log(X_1^{-\frac{1}{2}} X_2 X_1^{-\frac{1}{2}})^2) = \sum_{i=1}^d (\log \xi_i)^2 \tag{4}$$

where the  $\xi_i$ 's are the (positive) eigenvalues of  $X_1^{-\frac{1}{2}} X_2 X_1^{-\frac{1}{2}}$ .

On the tangent space, the Euclidean distance

$$d_{\mathcal{E}}^2(x_1, x_2) = Tr[(x_1 - x_2)^2], \tag{5}$$

with  $x_1 = \log_Y X_1$  and  $x_2 = \log_Y X_2$  for any  $Y \in Sym_d^+$ , is the first-order approximation of Eq. (4).

In [11], a boosting framework on  $Sym_d^+$  for detection (*i.e.*, binary classification) is presented. The idea is to build weak learners by regression over the mappings of the training points on a suitable tangent plane. This tangent plane is defined over the weighted Karcher mean [24] of the positive training data points, such to preserve their local layout on  $Sym_d^+$ . The negative points, instead, (*i.e.*, all but pedestrians) are assumed to be spread on the manifold, thus including them in the mean estimation would bias the result. Once moving from binary to multi-class classification the above considerations do not hold anymore, because we have many “positive” classes, each of them localized in a different part of

the manifold. Therefore, 1) choosing the Karcher mean of one class would privilege that class with respect to the others, 2) the Karcher mean of all classes is inadequate.

A thorough analysis of  $Sym_d^+$  opens a new perspective. First, its *sectional curvature*, the natural generalization of the classical Gaussian curvature for surfaces, is non-positive. Since  $Sym_d^+$  is actually a symmetric space, the following formula holds for computing the sectional curvature  $\kappa_{I_d}$  at  $I_d$  – due to the homogeneity of  $Sym_d^+$  [25], there is no loss of generality – with  $x, y \in Sym_d$  linearly independent:

$$\begin{aligned} \kappa_{I_d}(x, y) &= \frac{\langle R(x, y)x, y \rangle}{\|x\|^2\|y\|^2 - \langle x, y \rangle^2} = \frac{Tr([x, y], x)y)}{Tr(x^2)Tr(y^2) - (Tr(xy))^2} = \\ &= 2 \frac{Tr((xy)^2 - x^2y^2)}{Tr(x^2)Tr(y^2) - (Tr(xy))^2}, \end{aligned} \tag{6}$$

by the cyclical property of the trace. Here,  $[x, y] = xy - yx$  is the matrix commutator, and  $R(x, y) : z \mapsto [[x, y], z]$  is the *Riemann curvature operator* (in the symmetric space framework). It can be shown (for the actual proof, see Appendix in the additional material), that  $\kappa_{I_d}(x, y) \leq 0$ .

Now, an application of Preissmann’s theorem [25] shows that, taking the geodesic triangle with vertices  $I_d, X_1, X_2$ , one gets

$$d_{\mathcal{E}}(\log_{I_d} X_1, \log_{I_d} X_2) \leq d(X_1, X_2) \tag{7}$$

More precisely,

$$d(X_1, X_2) = d_{\mathcal{E}}(\log_{I_d} X_1, \log_{I_d} X_2) + \Xi(\kappa_{I_d}) \tag{8}$$

where  $\Xi(\kappa_{I_d}) \geq 0$  is a function that depends on the sectional curvature. An explicit form for  $\Xi$  cannot be easily derived, but it is evident that if the sectional curvature is “small”, one can replace the “true” distance by the Euclidean one.

Notice that the above remark reconcile the present “classical” approach with the one in [26,27], where the Log-euclidean metric is employed throughout, upon endowing  $Sym^+$  with a Lie group structure.

The reasoning above suggests a practical manoeuvre to check this condition. We randomly pick a representative set of covariance matrices from the datasets under observation and we estimate the sectional curvature (Eq. 6) for each pair, calculating the mean at the end. Experimentally, this mean value results  $-10^{-3}$ , that is far from the standard negative curvature of  $-1$ .

In this conditions, one can choose any point on  $Sym_d^+$  on which to map the dataset, and execute the learning on that (Euclidean) space. In practice, we choose the identity matrix  $I_d$ , as this simplifies the computation. Indeed, Eq. (3) becomes

$$\log_{I_d}(X) = \log(X) = U \log(D)U^T, \tag{9}$$

where  $U \log(D)U^T$  is the eigenvalue decomposition of  $X$ , with  $X$  a generic point in  $Sym_d^+$ ,  $U$  is an orthogonal matrix, and  $\log(D)$  is the diagonal matrix composed by the eigenvalues’ logarithms.

Moreover, the tangent space is the space of symmetric matrices, but there are only  $d(d + 1)/2$  independent coefficients, which are the upper triangular or lower triangular part of the matrix. Thus, by applying the vector operator, an orthonormal coordinate system for the tangent space is defined as

$$\text{vec}_{I_d}(x) = \text{vec}(x) = [x_{1,1} \ x_{1,2} \ \dots \ x_{1,d} \ x_{2,2} \ x_{2,3} \ \dots \ x_{d,d}], \tag{10}$$

where  $x$  is the map of  $X \in \text{Sym}_d^+$  in the tangent space. This operator relates the Riemannian metric on the tangent space to the canonical metric defined in  $\mathbb{R}^m$ , with  $m = d(d + 1)/2$ .

### 3.2 Algorithm Description

Following the considerations above, we map our dataset  $\mathcal{S}$  to the tangent Euclidean space  $T_{I_d}$ , and we perform the classification directly on this space. In this way,  $\mathcal{S}_T = \{\mathbf{x}_i, y_i\}_{i=1, \dots, N}$  is the mapped dataset, with  $\mathbf{x}_i = \text{vec}(\log_{I_d}(X_i))$ .

The essence of a boosting algorithm is an iterative re-weighting system that tends to focus on the most difficult examples in the training set. In the multi-class classification there are  $J$  different sets of weights built from the posterior distribution. Let  $\text{Pr}_j[\mathbf{x}_i]$  be the posterior probability for a training example  $\mathbf{x}_i$  to belong to the  $j$ -th class. It is computed as:

$$\text{Pr}_j[\mathbf{x}_i] = \frac{e^{F_j(\mathbf{x}_i)}}{\sum_{k=1}^J e^{F_k(\mathbf{x}_i)}}, \quad F_j(\mathbf{x}_i) = \sum_{l=1}^L f_{lj}(\mathbf{x}_i), \tag{11}$$

where  $\{f_{lj}\}_{l=1, \dots, L}$  is a class-specific set of weak learners. Each example in the training set  $\mathcal{S}_T$  is associated to a weight that depends on the class considered:

$$w_{ij} = \text{Pr}_j[\mathbf{x}_i](1 - \text{Pr}_j[\mathbf{x}_i]). \tag{12}$$

The core of the learning process is the definition of the inter-class decision boundaries, which is carried out by weak learners. We build weak classifiers  $g_{lj} : T_{I_d} \mapsto \mathbb{R}$  that solve a binary problem, one class against the others, then the multi-class classifiers  $f_{lj} : T_{I_d} \mapsto \mathbb{R}$  derive from their combination.

The binary weak learners  $g_{lj}$  solve a weighted regression problem, whose goodness of fit is measured by the response values  $z_{ij}$ , defined as:

$$z_{ij} = \frac{y_{ij}^* - \text{Pr}_j[\mathbf{x}_i]}{\text{Pr}_j[\mathbf{x}_i](1 - \text{Pr}_j[\mathbf{x}_i])}, \tag{13}$$

where  $y_{ij}^* = (j == y_i)$ . The combination of a set of  $J$  binary weak learners  $g_{lj}$  is provided by the following equation [17]:

$$f_{lj}(\mathbf{x}_i) = \frac{J - 1}{J} \left( g_{lj}(\mathbf{x}_i) - \frac{1}{J} \sum_{k=1}^J g_{lk}(\mathbf{x}_i) \right). \tag{14}$$

Please note that this operation is possible because the  $g_{lk}(\cdot)$ s live in the same domain  $T_{I_d}$ . If the binary classification had been carried out mapping each class

in a different space, similarly to [11], the combination of the results would have been much more complicated and unclear. Working on  $T_{\mathbf{Id}}$  represents an elegant and reasonable solution to the problem.

In the following we explain some details of the algorithm, summed up in pseudo-code here below.

---

**Algorithm 1.** Multi-class LogitBoost on  $\mathcal{M}$

---

**Require:**  $(X_1, y_1), \dots, (X_N, y_N)$  with  $X_i \in \mathcal{M}$  e  $y_i \in \{1, \dots, J\}$

- Map the data points to the tangent space  $T_{I_d}$ , by  $\mathbf{x}_i = (\log_{I_d}(X_i))$
- Start with weights  $w_{ij} = 1/N$  and  $i = 1, \dots, N$ ,  $F_j(\mathbf{x}_i) = 0$  e  $\text{Pr}_j[\mathbf{x}_i] = 1/J \forall j$ .

**for**  $l = 1, 2, \dots, L$  **do**

**for**  $j = 1, 2, \dots, J$  **do**

- Compute the response values (Eq. 12) and weights (Eq. 13).
- Fit the function  $g_{lj}(\mathbf{x}_i) : \mathbb{R}^m \mapsto \mathbb{R}$  by weighted least-square regression of  $z_{ij}$  to  $\mathbf{x}_i$  using weights  $w_{ij}$ .
- Set  $F_j(\mathbf{x}_i) \leftarrow F_j(\mathbf{x}_i) + f_{lj}(\mathbf{x}_i)$  where  $f_{lj}(\mathbf{x}_i)$  is defined in Eq. (14).
- Update  $\text{Pr}_j[\mathbf{x}_i]$  as in Eq. (11).
- Save  $F_j = \{g_{lj}\}$ .

**end for**

**end for**

- Save the ensemble of classifiers  $\{F_1, \dots, F_J\}$ .

---

### 3.3 Algorithm Details

**Binary weak classification strategy.** In boosting, it is possible to use very different types of weak learners. The most common are the decision stumps (or regression stumps), which are piecewise constant regression functions or linear regression functions. The original LogitBoost algorithm adopts linear regression functions as proposed in [17]. In a binary classification task a linear regression can be sufficient to solve the problem, as shown in [11] for pedestrian detection. However, a more powerful weak classification strategy is mandatory for the multi-class classification problem, as evidenced in [21], where piecewise constant functions are used.

After investigating different solutions, we have selected the weighted *regression trees* [28], which are more powerful than global models, like linear or polynomial regressors, where a single predictive formula is supposed to hold over the entire data space, and they have lower computational costs, in both the learning and testing phases. In order to avoid the risk of overtraining of the regression tree, we establish as stopping rule a minimal number  $\tau$  of observations per tree leaf, experimentally estimated (see Sec. 4).

**Stop condition.** It is important to specify a automatic stop criterion for the learning phase. The proposed rule is a composition of two terms. The first term takes into account the accuracy with which the classes are correctly classified:

we set the maximum accuracy  $\tau_{\text{acc}}$  for all the classes. The second term concerns the *learning rate*, which is the difference in accuracy between two consecutive iterations of LogitBoost. If the learning rate is less than  $\tau_{\text{r}}$  for all the classes, we assume that the boosting process has converged to its optimal solution. More formally, the learning process is stopped at the  $l$ -th iteration, when:

$$\text{acc}_l(j) \geq \tau_{\text{acc}} \quad \forall (\text{acc}_l(j) - \text{acc}_{l-1}(j)) \leq \tau_{\text{r}}, \quad \forall j \in \{1, \dots, J\}, \quad (15)$$

where  $\text{acc}_l(j)$  counts the examples of the  $j$ -th class correctly classified at the  $l$ -th iteration. In all the experiments,  $\tau_{\text{acc}}$  is set to 99% and  $\tau_{\text{r}}$  to 1%.

**Multi-class detection.** Our multi-class algorithm can be naturally extended to detection purposes by simply adding a class that contains background examples. It is a very large class, because it is potentially composed by all the possible images that do not contain foreground examples. For this reason, we combine the LogitBoost classifier with a *rejection cascade structure* [4].

Algorithm 1 becomes the learning procedure of each cascade level. The stop condition for a cascade level is given by Eq. (15), except for the background class that is optimized to correctly classify at least the 35% of the examples in this class, as in [11]. In practice, we order the examples in the background (BG) class, according to  $\text{Pr}_{\text{BG}}[\mathbf{x}]$ . Let  $\mathbf{x}_{\text{BG}}$  be the element with  $(0.35N_{\text{BG}})$ -th smallest probability among all the background examples. We set  $th_k = F_{\text{BG}}(\mathbf{x}_{\text{BG}})$ , where  $k$  is the current cascade level.

At the cascade level  $(k+1)$ , the BG class is first pruned using the cascade of  $k$  classifiers, rejecting the samples correctly classified as background. To obtain the desired rejection rate, the classification response for BG is redefined as  $F_{\text{BG}}(\mathbf{x}) = (F_{\text{BG}}(\mathbf{x}) - th_k)$ .

**Computational considerations.** The proposed framework inherits some of the computational characteristics of [11], where the main cost is due to SVD factorization needed for the projection of the covariance matrices on the tangent space (see Eq. 9). In our case, the presence of a unique projection point decreases the number of required SVD factorizations. This means a dramatic reduction of the computational cost in both the learning and testing phase.

## 4 Experiments

In this section, we show different video surveillance applications where our framework applies: head pose classification, pedestrian detection, and head detection + pose classification. In the first two cases, where comparative tests on shared databases are feasible, we outperform the relative best performances in the literature. In the third case, only qualitative results can be appreciated.

### 4.1 Head Pose Classification

We build a multi-class classifier for head pose classification on the 4 Pose Head Database<sup>1</sup>. This dataset contains head images of dimension  $50 \times 50$  (see some

<sup>1</sup> [http://www.eecs.qmul.ac.uk/~orozco/index\\_files/Page558.htm](http://www.eecs.qmul.ac.uk/~orozco/index_files/Page558.htm)



samples in Fig. 2a)) obtained from the i-LIDS dataset<sup>2</sup>. These images come from a real video surveillance scene, mirroring well typical critical conditions: they are noisy, motion-blurred, and at low resolution. The images are divided in 4 foreground (FG) classes: Back (4200 examples), Front (3555 examples), Left (3042 examples), and Right (4554 examples). Moreover, this dataset contains another set of 2216 background (BG) images. We partition the FG dataset in 2 equal parts, using one partition for training and one for testing. We extract from each image  $I$  a set  $\Phi$  of  $d = 12$  features, composed by:

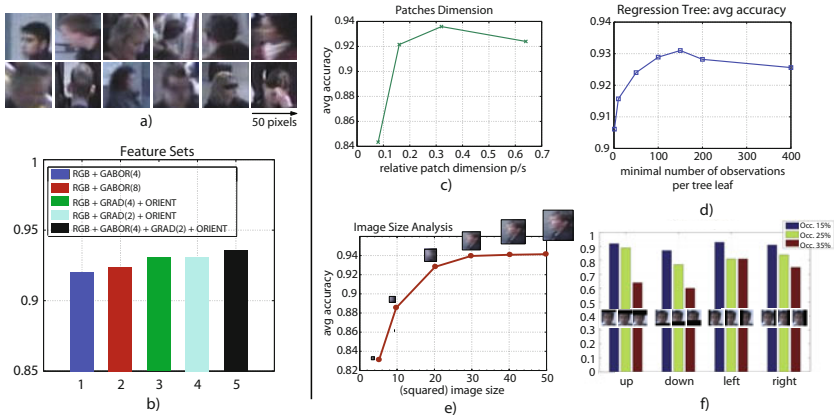
$$\Phi = [X \ Y \ R \ G \ B \ I_x \ I_y \ O \ \text{Gab}_{\{0, \pi/3, \pi/6, 4\pi/3\}}]. \quad (16)$$

$X, Y$  represent the spatial layout in  $I$ , and  $R, G, B$  are the color channels.  $I_x$  and  $I_y$  are the directional derivatives of  $I$ , and  $O$  is the gradient orientation. Finally,  $\text{Gab}$  is a set of 4 maps containing the results of Gabor filtering. We would like to stress that these features are particularly suited for head orientation classification. Apart from the general position ( $X, Y$ ) and shape information ( $I_x, I_y$ ), the covariance of the color channels permits to implicitly detect hair and skin textural properties. This particularly helps in distinguishing frontal from back views. Moreover, Gabor filters emphasize facial details, such as the vertical orientation for the nose, or the horizontal orientation of the mouth, if visible. We tried different combination of these filters, and the best results are obtained with dimension  $2 \times 4$ , sinusoidal frequency 16, and directions  $\mathcal{D} = \{0, \pi/3, \pi/6, 4\pi/3\}$ . In order to give an idea on how the choice of the features affects the system's performances, Fig. 2b depicts the behavior of the system in terms of mean classification accuracy by considering different subsets of  $\Phi$ . Once the features are extracted, we calculate the covariance matrices from all the patches of  $p \times p$  pixels, on a fixed grid of  $p/2$  pixels steps. This means that the patches remain overlapped by half of their size. We vary  $p$ , in order to investigate how the dimension (and thus, the number) of the patches modifies the classification performances. The best performance is obtained with  $p = 0.32s$ , where  $s$  is the image dimension. As visible in Fig. 2c, enlarging the patch dimension to more than this value diminishes the accuracy. This highlights that having a high number of small patches is better than having few large ones. This because with less, large-sized covariances all the image details are mixed together, losing the spatial information.

For each patch, a 4-class classifier is built, as described in Sec. 3.2. The  $\tau$  parameter, that rules the complexity of the regression trees, has been fixed to the optimal value 150 according to the accuracy test in Fig. 2d. It is interesting to note that exceeding this value, the performance drops, which is a sign of overtraining of the system.

A very important result is the ability to maintain a high classification accuracy on extremely low resolution images. Figure 2e shows the performance of our classifier varying the image dimension  $s$  (and changing proportionally the patch parameters, with  $p = \lceil 0.32s \rceil$ ). On a  $5 \times 5$  image we reach an average accuracy above 82%.

<sup>2</sup> <http://scienceandresearch.homeoffice.gov.uk/hosdb/cctv-imaging-technology/i-lids/>



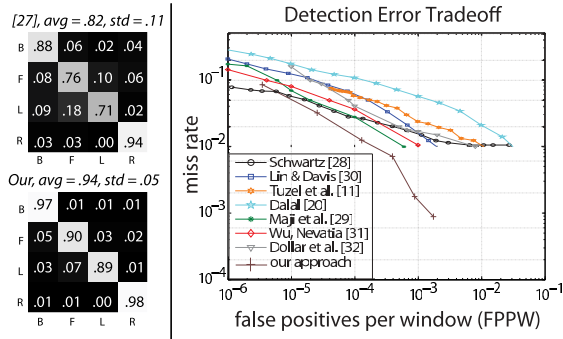
**Fig. 2.** Head pose classification analysis. (a) Some images from the considered dataset. Classification performance in terms of mean classification accuracy varying (b) the feature vector  $\Phi$ , (c) the patch dimensions  $p$ , (d) the regression tree stop criterion (the number  $\tau$  of elements per leaf), (e) the test image dimensions, and (f) considering occlusions of different strength.

Moreover, we test the ability of our classifier to deal with occlusions. Indeed, patch-based classifiers, as part-based classifiers, are naturally able to manage the presence of occlusions. We depict in Figure 2f the robustness to four types of occlusions (left-, right-, top- and bottom-side), in different sizes. As visible, top and bottom occlusions reduce the performances more, because they completely hide meaningful parts of the face.

Last, we compare our method with Orozco et al. [29], the state-of-the-art method for head pose classification for low resolution data. It is a head pose descriptor based on similarity distance maps to mean appearance templates of head images at different poses. All images in this dataset have their related pose descriptors, provided by the authors themselves [29]. The classifier is trained by Support Vector Machines (SVMs) using a polynomial kernel, as done in [29]. The result of the comparison, in terms of confusion matrix, is reported in Fig. 3. The average rate is 93.5% for our model, against 82.3% for Orozco's model.

## 4.2 Pedestrian Detection

We instantiate our framework on the binary problem of pedestrian detection to verify the performance of our approach on a pure detection task. We consider the INRIA Person dataset [20] for testing. It contains 1212 human images for the training part of dimension  $128 \times 64$  and 1133 images for the testing part. We pick a region of interest of  $50 \times 50$  at the center of the pedestrian images, that corresponds to the actual region where the pedestrian is enclosed (all positive examples come with a quite large border). Then, we use the same patch configuration described above (Sec. 4.1), but with a set of features  $\Phi$  more



**Fig. 3.** Confusion matrix for the method proposed in [29] (upper left) and our method (bottom left) for head pose orientation. On the right, DET curve for pedestrian detection, compared with the state-of-the-art methods [11,20,30,31,32,33,34].

suitable for the detection task, i.e. the same proposed in [11]. In Fig. 3c, we compare our framework with [11] and with the methods in [20,30,31,32,33,34]. The performances are evaluated by the Detection Error Tradeoff (DET) curve, that expresses the proportion of true detections against the proportion of false positives, on a log-log scale. The curve is estimated by varying the threshold  $th_k$  in the range  $[-1, 1]$ . We use a rejection cascade of 5 levels in which each level is populated by 10000 background examples. Please, note that augmenting the number of cascade levels to more than 5 does not appreciably increase the accuracy, since the number of covariance features remains fixed (in [11], instead, at each step a new feature is selected). Our detector clearly outperforms the other methods at the state-of-the-art, especially in terms of miss-rate.

### 4.3 Head Pose Detection and Classification

As we are proposing a multi-class framework, we can simply add a background class to the problem at hand, to perform detection along with classification. Here, we show how the system works for the problem of head pose detection and classification.

As first experiment, we consider the 4 head pose classes of the Pose Head dataset used in Sec. 4.1, adding its 2215 background examples. We use the same optimal settings estimated above, and we compare the performance of our approach with [29]. Even though the original paper performs classification only, so the comparison is a bit unfair, their template descriptor is provided for background images as well. We add the background class to the other positive classes, and we compute the classification stage by using SVMs, as described in the paper. The comparison, shown in Fig. 4, shows the ability of our system to naturally deal with this task as well.

On the other hand, the images of this dataset, though challenging for location and scale variations, are all taken from the same scene, with scarce lighting



**Fig. 4.** Confusion matrices for the experiments on head pose detection and classification. In (a) e (b), results for the first experiment on 4 Pose Head dataset (Orozco’s method in (a) [29], our method in (b)). (c) is the result of the second experiment with the more general dataset (see text for details). The other images are examples of detection and classifications in crowded scenes. The arrows indicate the head orientation. In green the correct answers provided by classifier, in red the misclassifications.

variations. Thus, the model we built is not general enough to work with different scenarios. For this reason, we perform a second experiment, building another model, and enriching the training set with new data coming from a different, more general, dataset. We use the head dataset employed in [35], composed by 2736  $20 \times 20$  head images, contained in a ROI of  $32 \times 32$  pixels. This dataset is mostly obtained from the INRIA person dataset, thus the images are taken from many different scenes and with a large variation of illumination conditions. The set of negative examples is composed by different real scenarios and other images containing parts of the body. We organize the data in four classes (plus background) according to heads’ orientation, since the original dataset does not contain such information.

The positive examples from the 4 Pose Head dataset are resized to  $20 \times 20$  pixels, whereas for the other dataset the examples are cropped from the center of the ROI. Half the data are used for training, and the testing set is composed by just the testing set of [35]. Fig. 4c summarizes the detection and classification results. Note that due to the variations in scale and position of the head, the cropped images can contain the head only partially. This is not a problem, though, since our model is robust to partial occlusions, as shown before. Finally, the other images in Fig. 4 show some qualitative results in crowded scenes, obtained with this last classifier.

## 5 Conclusions

In this paper, we face three classic video surveillance applications. We propose the novel general-purpose ARCO descriptor, and we adopt a common theoretical

framework of multi-class classification on Riemannian manifold  $Sym_d^+$ . Two are the advancements. From a practical point of view ARCO can describe faces as well as pedestrians, by including arbitrary features, and exploiting their dependencies via spatially local covariances. From a theoretical point of view, we show that  $Sym_d^+$  has non-positive sectional curvature and that where the curvature is almost flat we can perform multi-class Logitboost projecting the ARCO features on the tangent plane at any point of  $Sym_d^+$ . The experimental section validates the proposed approach, with novel state-of-the-art performances.

## Acknowledgments

This research is funded by the EU-Project FP7 SAMU- RAI, grant FP7-SEC-2007-01 No. 217899.

## References

1. Murphy-Chutorian, E., Trivedi, M.M.: Head pose estimation in computer vision: A survey. *IEEE Trans. PAMI* 31, 607–626 (2009)
2. Enzweiler, M., Gavrilu, D.M.: Monocular pedestrian detection: Survey and experiments. *IEEE Trans. PAMI* 31, 2179–2195 (2009)
3. Yang, M., Kriegman, D., Ahuja, N.: Detecting faces in images: A survey. *IEEE Trans. PAMI* 24, 34–58 (2002)
4. Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple. In: *Proc. CVPR* (2001)
5. Li, S., Zhu, L., Zhang, Z., Blake, A., Zhang, H., Shum, H.: Statistical learning of multi-view face detection. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002. LNCS*, vol. 2353, pp. 67–81. Springer, Heidelberg (2002)
6. Viola, M., Jones, M.J., Viola, P.: Fast multi-view face detection. In: *Proc. CVPR* (2003)
7. Huang, C., Ai, H., Li, Y., Lao, S.: Vector boosting for rotation invariant multi-view face detection. In: *Proc. ICCV*, pp. 446–453 (2005)
8. Wu, B., Ai, H., Huang, C., Lao, S.: Fast rotation invariant multi-view face detection based on real adaboost. In: *FGR*, pp. 79–84 (2004)
9. Li, S., Zhang, Z.: Floatboost learning and statistical face detection. *IEEE Trans. PAMI* 26 (2004)
10. Bar-Hillel, A., Hertz, T., Weinshall, D.: Object class recognition by boosting a part-based model. In: *Proc. CVPR*, pp. 702–709 (2005)
11. Tuzel, O., Porikli, F., Meer, P.: Pedestrian detection via classification on riemannian manifolds. *IEEE Trans. PAMI*, 1713–1727 (2008)
12. Yao, J., Odobez, J.: Fast Human Detection from Videos Using Covariance Features. In: *The Eighth International Workshop on Visual Surveillance* (2008)
13. Wu, B., Nevatia, R.: Optimizing discrimination-efficiency tradeoff in integrating heterogeneous local features for object detection. In: *Proc. CVPR* (2008)
14. Paisitkriangkrai, S., Shen, C., Zhang, J.: Performance evaluation of local features in human classification and detection. *IET-CV* 2, 236–246 (2008)
15. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* 55, 119–139 (1997)

16. Schapire, R., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. *Mach. Learn.* 37, 297–336 (1999)
17. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: A statistical view of boosting. *The Annals of Statistics* 28, 337–374 (2000)
18. Wu, B., Nevatia, R.: Detection and segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses. *IJCV* 82 (2009)
19. Tuzel, O., Porikli, F., Meer, P.: Region covariance: A fast descriptor for detection and classification. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) *ECCV 2006*. LNCS, vol. 3952, pp. 589–600. Springer, Heidelberg (2006)
20. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: *Proc. CVPR*, vol. 1, p. 886 (2005)
21. Zhang, J., Zhou, S., McMillan, L., Comaniciu, D.: Joint real-time object detection and pose estimation using probabilistic boosting network. In: *Proc. CVPR*, vol. 8 (2007)
22. Wu, B., Nevatia, R.: Detection and segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses. In: Sanderson, J.G. (ed.) *A Relational Theory of Computing*. LNCS, vol. 82, pp. 185–204. Springer, Heidelberg (1980)
23. Chen, Y.-T., Chen, C.-S., Hung, Y.-P., Chang, K.-Y.: Multi-Class Multi-Instance Boosting for Part-Based Human Detection. In: *ICCV 2009 Workshops*, pp. 1177–1184 (2009)
24. Karcher, H.: Riemannian Center of Mass and Mollifier Smoothing. *Comm. Pure and Applied Math.* 30, 509–541 (1997)
25. Chavel, I.: *Riemannian Geometry - A modern introduction*. Cambridge University Press, Cambridge (2006)
26. Pennec, X.: Probabilities and statistics on Riemannian manifolds: a geometric approach. Technical report, INRIA (2004)
27. Arsigny, V., Fillard, P., Pennec, X., Ayache, N.: Fast and simple calculus on tensors in the Log-Euclidean framework. In: Duncan, J.S., Gerig, G. (eds.) *MICCAI 2005*. LNCS, vol. 3749, pp. 115–122. Springer, Heidelberg (2005)
28. Breiman, L., Friedman, J., Olshen, R., Stone, C., Breiman, L., Hoeffding, W., Searfing, R., Friedman, J., Hall, O., Buhlmann, P., et al: Classification and Regression Trees. *Ann. Math. Statist.* 19, 293–325
29. Orozco, J., Gong, S., Xiang, T.: Head pose classification in Crowded Scenes. In: *Proc. BMVC* (2009)
30. Schwartz, W., Kembhavi, A., Harwood, D., Davis, L.: Human Detection Using Partial Least Squares Analysis. In: *Proc. ICCV* (2009)
31. Maji, S., Berg, A., Malik, J.: Classification using intersection kernel support vector machines is efficient. In: *Proc. CVPR*, vol. 1, p. 4 (2008)
32. Lin, Z., Davis, L.: A pose-invariant descriptor for human detection and segmentation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part IV*. LNCS, vol. 5305, pp. 423–436. Springer, Heidelberg (2008)
33. Wu, B., Nevatia, R.: Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In: *ICCV* (2005)
34. Dollár, P., Babenko, B., Belongie, S., Perona, P., Tu, Z.: Multiple component learning for object detection. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part II*. LNCS, vol. 5303, pp. 211–224. Springer, Heidelberg (2008)
35. Li, M., Zhang, Z., Huang, K., Tan, T.: Estimating the number of people in crowded scenes by MID based foreground segmentation and head-shoulder detection. In: *Proc. ICPR*, pp. 1–4 (2008)