Designing a Boosted Classifier on Riemannian Manifolds

Fatih Porikli, Oncel Tuzel, and Peter Meer

Abstract It is not trivial to build a classifier where the domain is the space of symmetric positive definite matrices such as nonsingular region covariance descriptors lying on a Riemannian manifold. This chapter describes a boosted classification approach that incorporates the apriori knowledge of the geometry of the Riemannian space. The presented classifier incorporated into a rejection cascade and applied to single image human detection task. Results on INRIA and DaimlerChrysler pedestrian datasets are reported.

1 Introduction

Detecting and locating different types of objects in visual data is one of the fundamental tasks in computer vision. Object detection be considered as a classification problem where each candidate image region is evaluated by a learned classifier for being from the specific object class or not. This can be accomplished by generative and discriminative learning [6, 25]; two of the major paradigms for solving prediction problems in machine learning, each offering distinct advantages.

In generative approaches [7, 29], one models conditional densities of object and non-object classes, and parameters are typically estimated using a likelihood-based criterion. In discriminative approaches, one directly models the mapping from inputs to outputs (often via a prediction function); parameters are estimated by optimizing objectives related to various loss functions. Discriminative approaches have

Fatih Porikli
ANU / NICTA, Canberra, Australia, e-mail: fatih.porikli@anu.edu.au

Oncel Tuzel
MERL, Cambridge, USA, e-mail: oncel@merl.com

Peter Meer
Rutgers University, USA e-mail: meer@cronos.rutgers.edu

shown better performance given enough data, as they are better tailored to the prediction task and appear more robust to model mismatches. Most of the leading approaches in object detection can be categorized as discriminative models such as neural networks (NNs) [12], support vector machines (SVMs) [3] or boosting [24], convolutional neural nets (CNNs) [1, 14]. These methods became increasingly popular since they can cope with high dimensional state spaces and are able to select relevant descriptors among a large set. In [22] NNs, and in [17] SVMs were utilized as a single strong classifier for detection of various categories of objects. The NNs and SVMs were also utilized for intermediate representations [5, 16] for final object detectors. In [28], multiple weak classifiers trained using AdaBoost were combined to form a rejection cascade.

In this chapter, we apply local object descriptors, namely region covariance descriptors, to human detection problem. Region covariance features were first introduced in [26] for matching and texture classification problems, and later were extended to many applications from tracking [20], event detection [11] and video classification successfully [30]. We represent a human with several covariance descriptors of overlapping image regions where the best descriptors are determined with a greedy feature selection algorithm combined with boosting. A region covariance descriptor is a covariance matrix that measures of how much pixel-wise variables, such as spatial location, intensity, color, derivatives, pixel-wise filter responses, etc., change together within the given image region. The space of these covariance matrices does not form a vector space. For example, it is not closed under multiplication with negative scalars. Instead, they constitute a connected Riemannian manifold. More specifically, nonsingular covariance matrices form a symmetric positive definite manifold that has Riemannian geometry.

It is not possible to use classical machine learning techniques to design the classifiers in this space. Consider a simple linear classifier that makes a classification decision by dividing the Euclidean space based on the value of a linear combination of input coefficients. For example, the simplest form a linear classifier in $\mathbb{R}^2$, which is a point and a direction vector in $\mathbb{R}^2$, define a line which separates $\mathbb{R}^2$ into two. A function that divides the manifold is rather a complicated notion compared with the Euclidean space. For example, if we consider the image of the lines on the 2-torus, the curves never divide the manifold into two. Typical approaches map such manifolds to higher dimensional Euclidean spaces, which corresponds to flattening of the manifold. They map the points on the manifold to a tangent space where traditional learning techniques can be used for classification. A tangent space is an Euclidean space relative to a point. Processing a manifold through a single tangent space is restrictive, as only distances to the original point are true geodesic distances. Distances between arbitrary points on the tangent space do not represent true geodesic distances. In general, there is no single tangent space mapping that globally preserves the distances between the points on the manifold. Therefore, a classifier trained on a single tangent space or flattened space does not reflect the global structure of the data points. As a remedy, we take advantage of the boosting framework that consist of iteratively learning weak learners in different tangent spaces to obtain a strong classifier. After a weak learner is added, the training data

is reweighted. Misclassified examples are set to gain weight and correctly classified examples to lose weight. Thus, consecutive weak learners focus more on the examples that previous weak learners misclassified. To improve speed, we further structure multiple strong classifiers into a final rejection cascade such that if any previous strong classifier rejects a hypothesis, then it is considered a negative example. This provides an efficient algorithm due to sparse feature selection, besides only a few classifiers are evaluated at most of the regions due to the cascade structure. A previous version of the classification method presented in this book chapter has been published in [27].

For completeness, we present an overview of Riemannian geometry focusing on the space of symmetric positive definite matrices in the next Section 2. We explain how to learn a boosted classifier on a Riemannian manifold in Section 3. Then, we describe the covariance descriptors in Section 4 and their application to human detection in Section 5 with experimental evaluations in Section 6.

2 Riemannian Manifolds

We refer to points on a manifold with capital letters $X \in M$, whereas symmetric positive definite matrices with capital bold letters $\mathbf{X} \in Sym_d^+$ and points on a tangent space with small bold letters $\mathbf{x} \in T_X$. The matrix norms are computed by the Frobenius norm $\|\mathbf{X}\|^2 = \text{trace}(\mathbf{X}\mathbf{X}^T)$, and the vector norms are the ℓ_2 norm.

2.1 Manifolds

A manifold M is a topological space which is locally similar to an Euclidean space. Every point on the manifold has a neighborhood for which there exists a homeomorphic mapping the neighborhood to $\mathbb{R}^m$. Technically, a manifold M of dimension d is a connected Hausdorff space for which every point has a neighborhood that is homeomorphic to an open subset of $\mathbb{R}^d$.

A differentiable manifold M^c is a topological manifold equipped with an equivalence class of atlas whose transition maps are c-times continuously differentiable. In case all the transition maps of a differentiable manifold are smooth, i.e. all its partial derivatives exist, then it is a smooth manifold M^∞.

For differentiable manifolds, it is possible to define the derivatives of the curves on the manifold and attach to every point X on the manifold a tangent space T_X, a real vector space that intuitively contains the possible directions in which one can tangentially pass through X. In other words, the derivatives at a point X on the manifold lies in a vector space T_X, which is the tangent space at that point. The tangent space T_X is the set of all tangent vectors at X. The tangent space is a vector space, thereby it is closed under addition and scalar multiplication.

2.2 Riemannian Geometry

A Riemannian manifold (M,g) is a differentiable manifold in which each tangent space has an inner product g metric, which varies smoothly from point to point. It is possible to define different metrics on the same manifold to obtain different Riemannian manifolds. In practice, this metric is chosen by requiring it to be invariant to some class of geometric transformations. The inner product g induces a norm for the tangent vectors on the tangent space $\|\mathbf{x}\|_X^2 = <\mathbf{x},\mathbf{x}>_X = g(\mathbf{x})$.

The minimum length curve connecting two points on the manifold is called the geodesic, and the distance between the points $d(X,Y)$ is given by the length of this curve. On a Riemannian manifold, a geodesic is a smooth curve that locally joins their points along the shortest path. Suppose $\gamma(r) : [r_0, r_1] \mapsto M$ be a smooth curve on M. The length of the curve $L(\gamma)$ is defined as

$$L(\gamma) = \int_{r_0}^{r_1} \|\gamma'(r)\| dr. \tag{1}$$

A smooth curve is called geodesic if and only if its velocity vector is constant along the curve $\|\gamma'(r)\| = const$. Suppose X and Y be two points on M. The distance between the points $d(X,Y)$, is the infimum of the length of the curves, such that, $\gamma(r_0) = X$ and $\gamma(r_1) = Y$. For each tangent vector $\mathbf{x} \in T_X$, there exists a unique geodesic γ starting at $\gamma(0) = X$ having initial velocity $\gamma'(0) = \mathbf{x}$. All the shortest length curves between the points are geodesics but not vice-versa. However, for nearby points the definition of geodesic and the shortest length curve coincide.

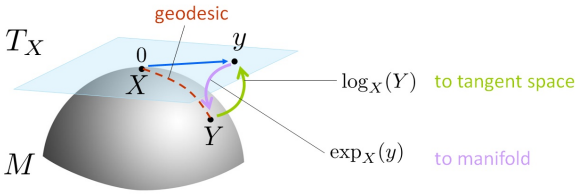

Fig. 1 Illustration of a manifold M and the corresponding tangent space T_X at X for point Y.

The exponential map, $\exp_X : T_X \mapsto M$, maps the vector $\mathbf{y}$ in the tangent space to the point reached by the geodesic after unit time $\exp_X(y) = 1$. Since the velocity along the geodesic is constant, the length of the geodesic is given by the norm of the initial velocity $d(X, \exp_X(y)) = \|y\|_X$. An illustration is shown in Figure 1. Under the exponential map, the image of the zero tangent vector is the point itself $\exp_X(0) = X$. For each point on the manifold, the exponential map is a diffeomorphism (one-to-one, onto and continuously differentiable mapping in both directions) from a neighborhood of the origin of the tangent space T_X onto a neighborhood of the point X.

In general, the exponential map $\exp_X$ is onto but only one-to-one in a neighborhood of X. Therefore, the inverse mapping $\log_X : X \mapsto T_X$ is uniquely defined only

around a small neighborhood of the point X. If for any $Y \in M$, there exists several $\mathbf{y} \in T_X$ such that $Y = \exp_X(\mathbf{y})$, then $\log_X(Y)$ is given by the tangent vector with the smallest norm. Notice that both operators are point dependent.

From the definition of geodesic and the exponential map, the distance between the points on manifold can be computed by

$$d(X,Y) = d(X,\exp_X(y)) = <\log_X(Y), \log_X(Y)>_X = \|\log_X(Y)\|_X = \|\mathbf{y}\|_X. \quad (2)$$

2.3 Space of Symmetric Positive Definite Matrices

The $d \times d$ dimensional nonsingular covariance matrices, i.e. region covariance descriptors, are symmetric positive definite Sym_d^+, and can be formulated as a connected Riemannian manifold. The set of symmetric positive definite matrices is not a multiplicative group. However, an affine invariant Riemannian metric on the tangent space of Sym_d^+ is given by [18]

$$<\mathbf{y},\mathbf{z}>_{\mathbf{X}} = \text{trace}\left(\mathbf{X}^{-\frac{1}{2}}\mathbf{y}\mathbf{X}^{-1}\mathbf{z}\mathbf{X}^{-\frac{1}{2}}\right). \quad (3)$$

The exponential map associated to the Riemannian metric

$$\exp_{\mathbf{X}}(\mathbf{y}) = \mathbf{X}^{\frac{1}{2}} \exp\left(\mathbf{X}^{-\frac{1}{2}}\mathbf{y}\mathbf{X}^{-\frac{1}{2}}\right)\mathbf{X}^{\frac{1}{2}} \quad (4)$$

is a global diffeomorphism. Therefore, the logarithm is uniquely defined at all the points on the manifold

$$\log_{\mathbf{X}}(\mathbf{Y}) = \mathbf{X}^{\frac{1}{2}} \log\left(\mathbf{X}^{-\frac{1}{2}}\mathbf{Y}\mathbf{X}^{-\frac{1}{2}}\right)\mathbf{X}^{\frac{1}{2}}. \quad (5)$$

Above, the exp and log are the ordinary matrix exponential and logarithm operators. Not to be confused, $\exp_{\mathbf{X}}$ and $\log_{\mathbf{X}}$ are manifold specific point dependent operators, i.e. $\mathbf{X} \in Sym_d^+$.

For symmetric matrices, these ordinary matrix exponential and logarithm operators can be computed easily. Let $\Sigma = \text{UDU}^T$ be the eigenvalue decomposition of a symmetric matrix. The exponential series is

$$\exp(\Sigma) = \sum_{k=0}^{\infty} \frac{\Sigma^k}{k!} = \text{U}\exp(\text{D})\text{U}^T \quad (6)$$

where $\exp(\text{D})$ is the diagonal matrix of the eigenvalue exponentials. Similarly, the logarithm is given by

$$\log(\Sigma) = \sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k}(\Sigma - \text{I})^k = \text{U}\log(\text{D})\text{U}^T. \quad (7)$$

The exponential operator is always defined, whereas the logarithms only exist for symmetric matrices with positive eigenvalues, Sym_d^+. From the definition of the geodesic given in the previous section, the distance between two points on Sym_d^+ is measured by substituting (5) into (3)

$$\begin{aligned} d^2(\mathbf{X},\mathbf{Y}) &= < \log_{\mathbf{X}}(\mathbf{Y}), \log_{\mathbf{X}}(\mathbf{Y}) >_{\mathbf{X}} \\ &= \text{trace}\left(\log^2(\mathbf{X}^{-\frac{1}{2}}\mathbf{Y}\mathbf{X}^{-\frac{1}{2}})\right). \end{aligned} \tag{8}$$

An equivalent form of the affine invariant distance metric was first given in [9], in terms of joint eigenvalues of $\mathbf{X}$ and $\mathbf{Y}$ as

$$d(\mathbf{X},\mathbf{Y}) = \left(\sum_{k=1}^{d} (\ln \lambda_k(\mathbf{X},\mathbf{Y}))^2\right)^{\frac{1}{2}} \tag{9}$$

where $\lambda_k(\mathbf{X},\mathbf{Y})$ are the generalized eigenvalues of $\mathbf{X}$ and $\mathbf{Y}$, computed from

$$\lambda_k \mathbf{X}\mathbf{v}_k - \mathbf{Y}\mathbf{v}_k = 0 \quad k = 1 \ldots d \tag{10}$$

and $\mathbf{v}_k$ are the generalized eigenvectors. This distance measure satisfies the metric axioms, positivity, symmetry, triangle inequality, for positive definite symmetric matrices.

2.4 Vectorized Representation for the Tangent Space of Sym_d^+

The tangent space of Sym_d^+ is the space of $d \times d$ symmetric matrices and both the manifold and the tangent spaces are $d(d+1)/2$ dimensional. There are only $d(d+1)/2$ independent coefficients which are the upper triangular or the lower triangular part of the matrix. The off-diagonal entries are counted twice during norm computation.

For classification, we prefer a minimal representation of the points in the tangent space. We define an orthonormal coordinate system for the tangent space with the vector operation. The orthonormal coordinates of a tangent vector $\mathbf{y}$ in the tangent space at point $\mathbf{X}$ is given by the vector operator

$$\text{vec}_{\mathbf{X}}(\mathbf{y}) = \text{vec}_{\mathbf{I}}(\mathbf{X}^{-\frac{1}{2}}\mathbf{y}\mathbf{X}^{-\frac{1}{2}}) \tag{11}$$

where $\mathbf{I}$ is the identity matrix and the vector operator at identity is defined as

$$\text{vec}_{\mathbf{I}}(\mathbf{y}) = [y_{1,1}\ \sqrt{2}y_{1,2}\ \sqrt{2}y_{1,3}\ \ldots\ y_{2,2}\ \sqrt{2}y_{2,3}\ \ldots\ y_{d,d}]^T. \tag{12}$$

Notice that, the tangent vector $\mathbf{y}$ is a symmetric matrix and with the vector operator $\text{vec}_{\mathbf{X}}(\mathbf{y})$ we get the orthonormal coordinates of $\mathbf{y}$ which is in $\mathbb{R}^d$. The vector operator relates the Riemannian metric (3) on the tangent space to the canonical metric

defined as

$$< \mathbf{y}, \mathbf{y} >_{\mathbf{X}} = \|\mathrm{vec}_{\mathbf{X}}(\mathbf{y})\|_2^2. \tag{13}$$

2.5 Mean of the Points on Sym_d^+

Let $\{\mathbf{X}_i\}_{i=1\ldots N}$ be a set of symmetric positive definite matrices on Riemannian manifold M. Similar to Euclidean spaces, the Riemannian center of mass [13], is the point on M which minimizes the sum of squared Riemannian distances

$$\mu = \arg\min_{\mathbf{X}\in M} \sum_{i=1}^{N} d^2(\mathbf{X}_i, \mathbf{X}) \tag{14}$$

where in our case d^2 is the distance metric (8). In general, the Riemannian mean for a set of points is not necessarily unique. This can be easily verified by considering two points at antipodal positions on a sphere, where the error function is minimal for any point lying on the equator. However, it is shown in several studies that the mean is unique and the gradient descent algorithm is convergent for Sym_d^+ [8] [15] [18].

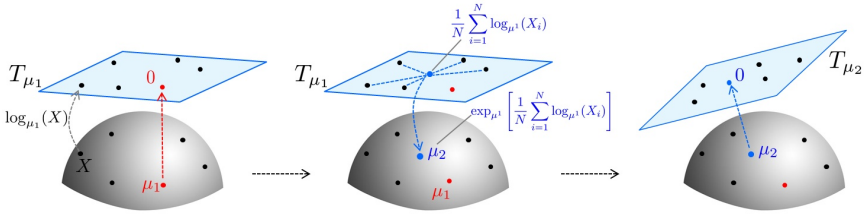

Fig. 2 Illustration of iterative mean computation by mapping back and forth to tangent space.

Differentiating the error function with respect to $\mathbf{X}$, we see that mean is the solution to the nonlinear matrix equation

$$\sum_{i=1}^{N} \log_{\mathbf{X}}(\mathbf{X}_i) = 0 \tag{15}$$

which gives the following gradient descent procedure [18]

$$\mu^{t+1} = \exp_{\mu^t}\left[\frac{1}{N}\sum_{i=1}^{N}\log_{\mu^t}(\mathbf{X}_i)\right]. \tag{16}$$

The method iterates by computing first order approximations to the mean on the tangent space. The weighted mean computation is similar to (16). We replace in-

side of the exponential, the mean of the tangent vectors, with the weighted mean $\frac{1}{\sum_{i=1}^{N} w_i} \sum_{i=1}^{N} w_i \log_{\mu^t}(\mathbf{X}_i)$ as shown in Figure 2.

3 Classification on Riemannian Manifolds

We use a boosted classification approach that consist of iteratively learning weak learners in different tangent spaces to obtain a strong classifier. After a weak learner is added, the training samples are reweighted such that the weights of the misclassified examples are increased and the weights of the correctly classified examples are increased with respect to a logistic regression rule. Boosting enables future learners focus more on the examples that previous weak learners misclassified.

Furthermore, we adopt a rejection cascade structure such that if any previous strong classifier rejects a hypothesis, then it is considered a negative example. This provides an efficient algorithm as majority of hypotheses in a test image are negatives that are dismissed early in the cascade.

Let $\{(\mathbf{X}_i, l_i)\}_{i=1\ldots N}$ be the training set with class labels, where $l_i \in \{0,1\}$. We aim to learn a strong classifier $F(\mathbf{X}) : M \mapsto \{0,1\}$, which divides the manifold into two based on the training set of the labeled items.

3.1 Local Maps and Weak Classifiers

We describe an incremental approach by training several weak classifiers on the tangent spaces, and combining them through boosting. We start by defining mappings from neighborhoods on the manifold to the Euclidean space, similar to coordinate charts. Our maps are the logarithm maps, $\log_{\mathbf{X}}$, that map the neighborhood of points $\mathbf{X} \in M$ to the tangent spaces $T_{\mathbf{X}}$. Since this mapping is a homeomorphism around the neighborhood of the point, the structure of the manifold is locally preserved. The tangent space is a vector space, and we use standard machine learning techniques to learn the classifiers on this space.

For classification task, the approximations to the Riemannian distances computed on the ambient space should be as close to the true distances as possible. Since we approximate the distances (3) on the tangent space $T_{\mathbf{X}}$,

$$d^2(\mathbf{Y},\mathbf{Z}) \approx \|\mathrm{vec}_{\mathbf{X}}(\log_{\mathbf{X}}(\mathbf{Z})) - \mathrm{vec}_{\mathbf{X}}(\log_{\mathbf{X}}(\mathbf{Y}))\|_2^2 \tag{17}$$

is a first order approximation. The approximation error can be expressed in terms of the pairwise distances computed on the manifold and the tangent space

$$\varepsilon = \sum_{i=1}^{N} \sum_{j=1}^{N} \left(d(\mathbf{X}_i,\mathbf{X}_j) - \left\| \mathrm{vec}_{\mathbf{X}}(\log_{\mathbf{X}}(\mathbf{X}_i)) - \mathrm{vec}_{\mathbf{X}}(\log_{\mathbf{X}}(\mathbf{X}_j)) \right\|_2 \right)^2 \tag{18}$$

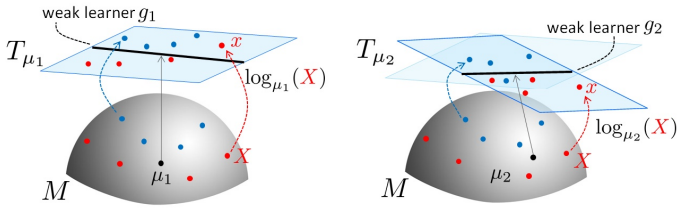

Fig. 3 Two iterations of boosting on a Riemannian manifold. The manifold is depicted with the surface of the sphere and the plane is the tangent space at the mean. The samples are projected to tangent spaces at means via $\log_{\mu}$. The weak learners g are learned on the tangent spaces T_{μ}. Left: sample $\mathbf{X}$ is misclassified therefore its weight increases. In the second iteration of boosting (right), the weighted mean moves towards $\mathbf{X}$.

which is equal to

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\left\|\log\left(\mathbf{X}_i^{-\frac{1}{2}}\mathbf{X}_j\mathbf{X}_i^{-\frac{1}{2}}\right)\right\|_F - \left\|\log\left(\mathbf{X}^{-\frac{1}{2}}\mathbf{X}_i\mathbf{X}^{-\frac{1}{2}}\right) - \log\left(\mathbf{X}^{-\frac{1}{2}}\mathbf{X}_j\mathbf{X}^{-\frac{1}{2}}\right)\right\|_F\right)^2 \tag{19}$$

for the space of symmetric positive definite matrices using (5) and (13).

The classifiers can be learned on the tangent space at any point $\mathbf{X}$ on the manifold. Best approximation, which preserves the pairwise distances is achieved at the minimum of ε. The error can be minimized with respect to $\mathbf{X}$ which gives the best tangent space to learn the classifier.

Since the mean of the points (14) is the minimizer of the sum of squared distances from the points in the set and the mapping preserves the structure of the manifold locally, it is also a good candidate for the minimizer of the error function (19). However, for this a theoretical proof does not exist. For some special cases it can be easily verified that the mean is the minimizer. Such a case arises when all the points lie on a geodesic curve, where the approximation error is zero for any point lying on the curve. Since mean also lies on the geodesic curve, the approximation is perfect. Nevertheless, for a general set of points, we only have empirical validation based on simulations. We generated random points on Sym_d^+, many times with varying d. The approximation errors were measured on the tangent spaces at any of the points $T_{\mathbf{X}_{i=1\ldots N}}$ and at the mean $T_{\mathbf{X}_\mu}$. In our simulations, the errors computed on the tangent spaces at the means were significantly lower than any other choice and counter examples were not observed. The simulations were also repeated for weighted sets of points, where the minimizers of the weighted approximation errors were achieved at the weighted means of the points.

Therefore, the weak learners are learned on the tangent space at the mean of the points. At each iteration, we compute the weighted mean of the points through (16), where the weights are adjusted through boosting. Then, we map the points to the tangent space at the weighted mean and learn a weak classifier on this vector space. Since the weights of the samples which are misclassified during the earlier stages of boosting increase, the weighted mean moves towards these points and more accurate

classifiers are learned for these points. The process is illustrated in Figure 3. To evaluate a test example, the sample is projected to the tangent spaces at the computed weighted means, and the weak learners are evaluated. The approximation error is minimized by averaging over several weak learners.

3.2 LogitBoost on Riemannian Manifolds

Consider the binary classification problem with labels $l_i \in \{0,1\}$ on vector spaces. The probability of $\mathbf{x}$ being in class 1 is represented by

$$p(\mathbf{x}) = \frac{e^{F(\mathbf{x})}}{e^{F(\mathbf{x})} + e^{-F(\mathbf{x})}} \qquad F(\mathbf{x}) = \frac{1}{2}\sum_{k=1}^{K} f_k(\mathbf{x}). \tag{20}$$

The LogitBoost algorithm learns the set of regression functions $\{f_k(\mathbf{x})\}_{k=1\ldots K}$ (weak learners) by minimizing the negative binomial log-likelihood of the data $(l, p(\mathbf{x}))$

$$-\sum_{i=1}^{N} [l_i \log(p(\mathbf{x}_i)) + (1-l_i)\log(1-p(\mathbf{x}_i))] \tag{21}$$

through Newton iterations. At the core of the algorithm, LogitBoost fits a weighted least square regression, $f_k(\mathbf{x})$ of training points $\mathbf{x}_i \in \mathbb{R}^d$ to response values $z_i \in \mathbb{R}$ with weights w_i where

$$z_i = \frac{l_i - p(\mathbf{x}_i)}{p(\mathbf{x}_i)(1-p(\mathbf{x}_i))} \qquad w_i = p(\mathbf{x}_i)(1-p(\mathbf{x}_i)). \tag{22}$$

The LogitBoost algorithm [10] on Riemannian manifolds is similar to the original LogitBoost, except a few differences at the level of weak learners. In our case, the domain of the weak learners are in M such that $f_k(\mathbf{X}) : M \mapsto \mathbb{R}$. Following the discussion of the previous section, we learn the regression functions on the tangent space at the weighted mean of the points. We define the weak learners as

$$f_k(\mathbf{X}) = g_k(\mathrm{vec}_{\mu_k}(\log_{\mu_k}(\mathbf{X}))) \tag{23}$$

and learn the functions $g_k(\mathbf{x}) : \mathbb{R}^d \mapsto \mathbb{R}$ and the weighted mean of the points $\mu_k \in M$. Notice that the mapping vec_{μ_k} (11), gives the orthonormal coordinates of the tangent vectors in T_{μ_k}.

The algorithm is presented in Figure 4. The steps marked with $(^*)$ are the differences from original LogitBoost algorithm. For functions $\{g_k\}_{k=1\ldots K}$, it is possible to use any form of weighted least squares regression such as linear functions, regression stumps, etc., since the domain of the functions are in $\mathbb{R}^d$.

Input: Training set $\{(\mathbf{X}_i, l_i)\}_{i=1...N}$, $\mathbf{X}_i \in M$, $l_i \in \{0,1\}$

- Initialize weights $w_i = 1/N$, $i = 1...N$, $F(\mathbf{X}) = 0$ and $p(\mathbf{X}_i) = \frac{1}{2}$
- Repeat for $k = 1...K$
  - Compute the response values and weights $z_i = \frac{y_i - p(\mathbf{X}_i)}{p(\mathbf{X}_i)(1-p(\mathbf{X}_i))}$, $w_i = p(\mathbf{X}_i)(1-p(\mathbf{X}_i))$
  - Compute weighted mean of the points through (16) $\mu_k = \arg\min_{\mathbf{X} \in M} \sum_{i=1}^{N} w_i d^2(\mathbf{X}_i, \mathbf{X})$ $(^*)$
  - Map the data points to the tangent space at μ_k $\mathbf{x}_i = \mathrm{vec}_{\mu_k}(\log_{\mu_k}(\mathbf{X}_i))$ $(^*)$
  - Fit the function $g_k(\mathbf{x})$ by weighted least-square regression of z_i to $\mathbf{x}_i$ using weights w_i
  - Update $F(\mathbf{X}) \leftarrow F(\mathbf{X}) + \frac{1}{2} f_k(\mathbf{X})$ where f_k is defined in (23) and $p(\mathbf{X}) \leftarrow \frac{e^{F(\mathbf{X})}}{e^{F(\mathbf{X})} + e^{-F(\mathbf{X})}}$
- Store $F = \{\mu_k, g_k)\}_{k=1...K}$
- Output the classifier $\mathrm{sign}[F(\mathbf{X})] = \mathrm{sign}[\sum_{k=1}^{K} f_k(\mathbf{X})]$

Fig. 4 LogitBoost on Riemannian manifolds.

4 Region Covariance Descriptors

Let $\{\mathbf{z}_i\}_{i=1..S}$ be the d-dimensional features (such as intensity, color, gradients, filter responses, etc.) of pixels inside a region R. The corresponding $d \times d$ region covariance descriptor is

$$\mathbf{C}_R = \frac{1}{S-1} \sum_{i=1}^{S} (\mathbf{z}_i - \mu)(\mathbf{z}_i - \mu)^T \tag{24}$$

where μ is a vector of the means of the features inside the regions. In Figure 5, we illustrate the construction of region covariance descriptors. The diagonal entries of the covariance matrix represent the variance of each feature and the nondiagonal entries their respective correlations. Region covariance descriptors constitute the space of positive semi-definite matrices $Sym_d^{0,+}$. By adding a small diagonal matrix (or guaranteeing no features in the feature vectors would be exactly identical), they can be transformed into Sym_d^+.

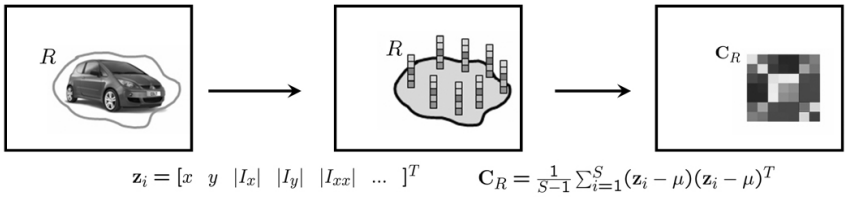

Fig. 5 Region covariance descriptor. The d-dimensional feature image Φ is constructed from input image I. The region R is represented with the covariance matrix, $\mathbf{C}_R$, of the features $\{\mathbf{z}_i\}_{i=1..S}$.

There are several advantages of using covariance matrices as region descriptors. The representation proposes a natural way of fusing multiple features which might be correlated. A single covariance matrix extracted from a region is usually enough to match the region in different views and poses. The noise corrupting individual samples are largely filtered out with the average filter during covariance computation. The descriptors are low-dimensional and due to symmetry $\mathbf{C}_R$ has only $d(d+1)/2$ different values (d is often less than 10) as opposed to hundreds of bins or thousands of pixels. Given a region R, its covariance $\mathbf{C}_R$ does not have any information regarding the ordering and the number of points. This implies a certain scale and rotation invariance over the regions in different images. Nevertheless, if information regarding the orientation of the points are represented, such as the gradient with respect to x and y, the covariance descriptor is no longer rotationally invariant. The same argument is also correct for illumination, too.

4.1 Region Covariance Descriptors for Human Detection

For human detection, we define the features as

$$\left[x \;\; y \;\; |I_x| \;\; |I_y| \;\; \sqrt{I_x^2 + I_y^2} \;\; |I_{xx}| \;\; |I_{yy}| \;\; \arctan \frac{|I_x|}{|I_y|} \right]^T \tag{25}$$

where x and y are pixel location, $I_x, I_{xx}, ..$ are intensity derivatives and the last term is the edge orientation. With the defined mapping, the input image is mapped to a $d = 8$ dimensional feature image. The covariance descriptor of a region is an 8×8 matrix, and due to symmetry only upper triangular part is stored, which has only 36 different values. The descriptor encodes information of the variances of the defined features inside the region, their correlations with each other and spatial layout.

Given an arbitrary sized detection window R, there are a very large number of covariance descriptors that can be computed from subwindows $r_{1,2,...}$. We perform sampling and consider all the subwindows r starting with minimum size of $1/10$ of the width and height of the detection window R, at all possible locations. The size of r is incremented in steps of $1/10$ along the horizontal or vertical, or both, until $r = R$. Although the approach might be considered redundant due to overlaps, there is significant evidence that the overlapping regions are an important factor in detection performances [4, 31]. The greedy feature selection mechanism, that will be described later, allows us to search for the best regions during learning classifiers.

Although it has been mentioned that the region covariance descriptors are robust towards illumination changes, we would like to enhance the robustness to also include local illumination variations in an image. Let r be a possible feature subwindow inside the detection window R. We compute the covariance of the detection window $\mathbf{C}_R$ and subwindow $\mathbf{C}_r$ using integral representation [19]. The normalized covariance descriptor of region r, $\hat{\mathbf{C}}_r$, is computed by dividing the columns and rows of $\mathbf{C}_r$ with the square root of the respective diagonal entries of $\mathbf{C}_R$,

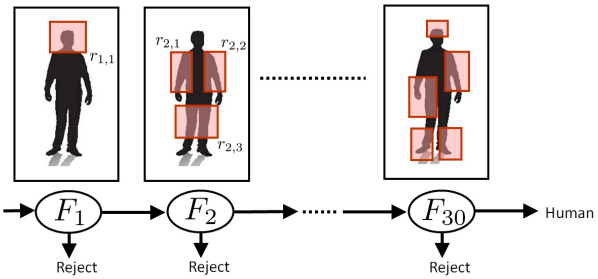

Fig. 6 Cascade of LogitBoost classifiers. The mth LogitBoost classifier selects normalized covariance descriptors of subwindows $r_{m,k}$.

$$\hat{\mathbf{C}}_r = diag(\mathbf{C}_R)^{-\frac{1}{2}} \mathbf{C}_r diag(\mathbf{C}_R)^{-\frac{1}{2}} \tag{26}$$

where $diag(\mathbf{C}_R)$ is equal to $\mathbf{C}_R$ at the diagonal entries and the rest is truncated to zero. The method described is equivalent to first normalizing the feature vectors inside the region R to have zero mean and unit standard deviation, and after that computing the covariance descriptor of subwindow r. Notice that under the transformation, $\hat{\mathbf{C}}_R$ is equal to the correlation matrix of the features inside the region R. The process only requires d^2 extra division operations.

5 Application to Human Detection

Due to the significantly large number of possible candidate detection windows R in a single image as a result of search in multiple scales and locations, and due to the considerable cost of the distance computation for each weak classifier, we adopt a rejection cascade structure to accelerate the detection process.

The domain of the classifier is the space of 8-dimensional symmetric positive definite matrices, Sym_8^+. We combine $K = 30$ strong LogitBoost classifiers on Sym_8^+ with rejection cascade, as shown in Figure 6. Weak learners $g_{m,k}$ are linear regression functions learned on the tangent space of Sym_8^+. A very large number of covariance descriptors can be computed from a single detection window R. Therefore, we do not have a single set of positive and negative features, but several sets corresponding to each of the possible subwindows. Each weak learner is associated with a single subwindow of the detection window. Let $r_{m,k}$ be the subwindow associated with k-the weak learner of cascade level m.

Let R_i^+ and R_i^- refer to the N_p positive and N_n negative samples in the training set, where $N = N_p + N_n$. While training the m-th cascade level, we classify all the negative examples $\{R_i^-\}_{i=1\ldots N_n}$ with the cascade of the previous $(m-1)$ LogitBoost classifiers. The samples which are correctly classified (samples classified as negative) are removed from the training set. Any window sampled from a negative image

is a negative example, therefore the cardinality of the negative set, N_n, is very large. During training of each cascade level, we sample 10000 negative examples.

We have varying number of weak learners K_m for each LogitBoost classifier m.Each cascade level is optimized to correctly detect at least 99.8% of the positive examples, while rejecting at least 35% of the negative examples. In addition, we enforce a margin constraint between the positive samples and the decision boundary. Let $p_m(R)$ be the learned probability function of a sample being positive at cascade level m, evaluated through (20). Let R_p be the positive example that has the $(0.998N_p)$-th largest probability among all the positive examples. Let R_n be the negative example that has the $(0.35N_n)$-th smallest probability among all the negative examples. We continue to add weak classifiers to cascade level m until $p_m(R_p) - p_m(R_n) > 0.2$. When the constraint is satisfied, the threshold (decision boundary) for cascade level m is stored as $\tau_m = F_m(R_n)$.

A test sample is classified as positive by cascade level m if $F_m(R) > \tau_m$ or equivalently $p_m(R) > p_m(R_n)$. With the proposed method, any of the positive training samples in the top 99.8 percentile have at least 0.2 margin more probability than the points on the decision boundary. The process continues with the training of $(m+1)$-th cascade level, until $m = 30$.

We incorporate a greedy feature selection method to produce a sparse set of classifiers focusing on important subwindows. At each boosting iteration k of the m-th LogitBoost level, we sample 200 subwindows among all the possible subwindows, and construct normalized covariance descriptors. We learn the weak classifiers representing each subwindow, and add the best classifier that minimizes the negative binomial log-likelihood (21) to the cascade level m. The procedure iterates with the training the $(k+1)$-th weak learner until the specified detection rates are satisfied.

The negative sample set is not well characterized for detection tasks. Therefore, while projecting the points to the tangent space, we compute the weighted mean of only the positive samples. Although rarely happens, if some of the features are fully correlated, there will be singularities in the covariance descriptor. We ignore those cases by adding very small identity matrix to the covariance.

The learning algorithm produces a set of 30 LogitBoost classifiers which are composed of K_m triplets $F_m = \left\{(r_{m,k}, \mu_{m,k}, g_{m,k})\right\}_{k=1\ldots K_m}$ and τ_m, where $r_{m,k}$ is the selected subwindow, $\mu_{m,k}$ is the mean and $g_{m,k}$ is the learned regression function of the k-th weak learner of the m-th cascade. To evaluate a *test region* R with m-th classifier, the normalized covariance descriptors constructed from regions $r_{m,k}$ are projected to tangent spaces $T\mu_{m,k}$ and the features are evaluated with $g_{m,k}$

$$\operatorname{sign}\left[F_m(R) - \tau_m\right] = \operatorname{sign}\left[\sum_{k=1}^{K_m} g_{m,k}\left(\operatorname{vec}\mu_{m,k}\left(\log\mu_{m,k}\left(\hat{\mathbf{C}}_{r_{m,k}}\right)\right)\right) - \tau_m\right]. \qquad (27)$$

The initial levels of the cascade are learned on relatively easy examples, thus there are very few weak classifiers in these levels. Due to the cascade structure, only a few are evaluated for most of the test samples, which produce a very efficient solution.

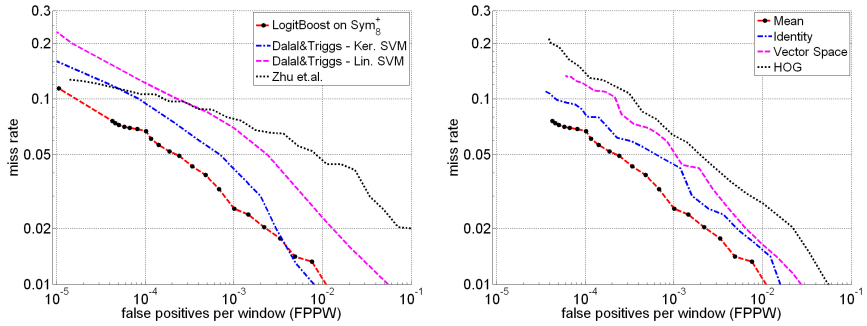

Fig. 7 Left: Comparison with methods of Dalal & Triggs [4] and Zhu et.al. [31] on INRIA dataset. The curves for other approaches are generated from the respective papers. Right: Detection rates of different approaches for our method on INRIA dataset.

6 Experiments and Discussion

We conduct experiments on INRIA and DaimlerChrysler datasets. Since the sizes of the pedestrians in a scene are not known apriori, the images are searched at multiple scales. There are two searching strategies. The first strategy is to scale the detection window and apply the classifier at multiple scales. The second strategy is to scale the image and apply the classifier at the original scale. In covariance representation we utilized gradient based features which are scale dependent. Therefore evaluating classifier at the original scale (second strategy) produces the optimal result. However, in practice up to scales of 2x we observed that the detection rates were almost the same, whereas in more extreme scale changes the performance of the first strategy degraded. The drawback of the second strategy is slightly increased search time, since the method requires computation of the filters and the integral representation at multiple scales.

INRIA pedestrian dataset [4] contains 1774 pedestrian annotations (3548 with reflections) and 1671 person free images. The pedestrian annotations were scaled into a fixed size of 64×128 windows which include a margin of 16 pixels around the pedestrians. The dataset was divided into two, where 2416 pedestrian annotations and 1218 person free images were selected as the training set, and 1132 pedestrian annotations and 453 person free images were selected as the test set. Detection on INRIA pedestrian dataset is challenging since it includes subjects with a wide range of variations in pose, clothing, illumination, background and partial occlusions.

First, we compare our results with [4] and [31]. Although it has been noted that kernel SVM is computationally expensive, we consider both the linear and kernel SVM method of [4]. In [31], a cascade of AdaBoost classifiers was trained using HOG features, and two different results were reported based on the normalization of the descriptors. Here, we consider only the best performing result, the ℓ_2-norm. In Figure 7-left, we plot the detection error trade-off curves on a log-log scale. The vertical-axis corresponds to the miss rate $\frac{\text{FalseNeg}}{\text{FalseNeg}+\text{TruePos}}$, and the horizontal-axis

corresponds to false positives per window (FPPW) $\frac{\text{FalsePos}}{\text{TrueNeg+FalsePos}}$. The curve for our method is generated by adding one cascade level at a time. For example, in our case the rightmost marker at $7.5 * 10^{-3}$ FPPW corresponds to detection using only the first 11 levels of cascade, whereas the marker positioned at $4 * 10^{-5}$ FPPW corresponds to cascade of all 30 levels. The markers between the two extremes correspond to a cascade of between 11 to 30 levels.

To generate the result at 10^{-5} FPPW (leftmost marker), we shifted the decision boundaries of all the cascade levels, τ_m, to produce less false positives at the cost of higher miss rates. We see that at almost all the false positive rates, our miss rates are significantly lower than the other approaches. The closest result to our method is the kernel SVM classifier of [4], which requires kernel evaluation at 1024 dimensional space to classify a single detection window. If we consider 10^{-4} as an acceptable FPPW, our miss rate is 6.8%, where the second best result is 9.3%.

Since the method removes samples which were rejected by the previous levels of cascade, during the training of last levels, only very small amount of negative samples, order of 10^2, remained. At these levels, the training error did not generalize well, such that the same detection rates are not achieved on the test set. This can be seen by the dense markers around FPPW $< 7 * 10^{-5}$. We believe that better detection rates can be achieved at low false positive rates with introduction of more negative images. In our method, 25% of false positives are originated from a single image which contains a flower texture, where the training set does not include a similar example. We note that, in [23] a pedestrian detection system utilizing shapelet features is described which has $20-40\%$ lower miss rates at equal FPPWs on INRIA dataset, compared to our approach. The drawback of the method is the significantly higher computational requirement.

We also consider an empirical validation of the presented classification algorithm on Riemannian manifolds. In Figure 7-right, we present the detection error tradeoff curves for four different approaches: 1) The original method, which maps the points to the tangent spaces at the weighted means. 2) The mean computation step is removed from the original algorithm and points are always mapped to the tangent space at the identity. 3) We ignore the geometry of Sym_8^+, and stack the upper triangular part of the covariance matrix into a vector, such that learning is performed on the vector space. 4)We replace the covariance descriptors with HOG descriptors, and perform original (vector space) LogitBoost classification.

The original method outperforms all the other approaches significantly. The second best result is achieved by mapping points to the tangent space at the identity matrix followed by the vector space approaches. Notice that, our LogitBoost implementation utilizing HOG descriptors has 3% more miss rate at 10^{-4} FPPW than [31] which trains an AdaBoost classifier. The performance is significantly degraded beyond this point.

DaimlerChrysler dataset [16] contains 4000 pedestrian (24000 with reflections and small shifts) and 25000 non-pedestrian annotations. As opposed to INRIA dataset, non-pedestrian annotations were selected by a preprocessing step from the negative samples, which match a pedestrian shape template based on average Chamfer distance score. Both annotations were scaled into a fixed size of 18×36 win-

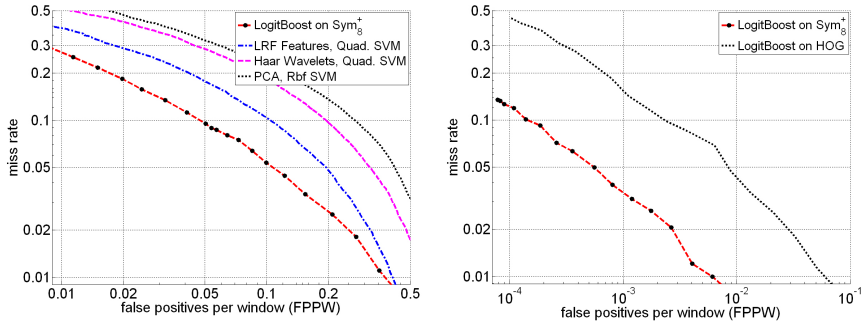

Fig. 8 Left: Comparison with [16] on DaimlerChrysler dataset. The curves for other approaches are generated from the original paper. Comparison of covariance and HOG descriptors on DaimlerChrysler dataset.

dows, and pedestrian annotations include a margin of 2 pixels around. The dataset was organized into three training and two test sets, each of them having 4800 positive and 5000 negative examples. The small size of the windows combined with a carefully arranged negative set makes detection on DaimlerChrysler dataset extremely challenging. In addition, 3600 person free images with varying sizes between 360×288 and 640×480 were also supplied.

In [16], an experimental study was described comparing three different feature descriptors and various classification techniques. The compared feature descriptors were the PCA coefficients, Haar Wavelets and local receptive fields (LRFs) which are the output of the hidden layer of a specially designed feed forward NN. We compare our method with the best results for each descriptor in [16]. The same training configuration is prepared by selecting two out of three training sets. Since the number of non-pedestrian annotations was very limited for training of our method, we adapted the training parameters. A cascade of $K = 15$ LogitBoost classifiers on Sym_8^+ is learned, where each level is optimized to detect at least 99.75% of the positive examples, while rejecting at least 25% negative samples.

In Figure 8-left, we plot the detection error trade-off curves. The cascade of 15 LogitBoost classifiers produced a FPPW rate of 0.05. The detection rates with lower FPPW are generated by shifting the decision boundaries of all the cascade levels gradually, until FPPW $=$ 0.01. We see that our approach has significantly lower miss rates at all the false positive rates. This experiment should not be confused with the experiments on INRIA dataset, where much lower FPPW rates were observed. Here, the negative set consists of hard examples selected by a preprocessing step.

We also we set up a different test configuration on DaimlerChrysler dataset. The 3600 person free images are divided into two, where 2400 images are selected as the negative training set, and 1200 images are selected for the negative test set. For both the covariance descriptors and the HOG descriptors, we trained cascade of 25 classifiers. We observed that the object sizes were too small for HOG descriptors to separate among positive and negative examples at the later levels of cascade. The classifiers trained utilizing HOG descriptors failed to achieve the specified detection

Fig. 9 Detection examples. White dots show all the detection results. Black dots are the modes generated by mean-shift smoothing and the ellipses are average detection window sizes. There are extremely few false positives and negatives.

(99.8%) and the rejection rates (35.0%). We stopped adding weak learners to a cascade level after reaching $K_m = 100$. The detection error trade-off curves are given in Figure 8-right where we see that the covariance descriptors significantly outperform HOG descriptors.

Utilizing the classifier trained on the INRIA dataset, we generated several detection examples for crowded scenes with pedestrians having variable illumination, appearance, pose and partial occlusion. The results are shown in Figure 9. The images are searched at five scales using the first strategy, starting with the original window size 64×128 and two smaller and two larger scales of ratio 1.2. The white dots are all the detection results and we filtered them with adaptive bandwidth mean-shift filtering [2] with bandwidth 0.1 of the window width and height. Black dots show the modes, and ellipses are generated by averaging the detection window sizes converging to the modes.

7 Remarks

The presented LogitBoost learning algorithm is not specific to Sym_d^+ and can be used to train classifiers for points lying on any connected Riemannian manifold. In addition, the approach can be combined with any boosting method including GentleBoost and AdaBoost classifiers on Riemannian manifolds using LDA, decision stumps and linear SVMs as weak learners. In our experiments, the results of the methods were comparable.

References

1. Y. Bengio, I. Goodfellow and A. Courville. Deep Learning. *Book, MIT Press*, 2014.
2. D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24, 603–619, 2002.
3. C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20, 273–297, 1995.
4. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *IEEE Conference on Computer Vision and Pattern Recognition*, 886–893, 2005.
5. G. Dorkó and C. Schmid. Selection of scale-invariant parts for object class recognition. *International Conference on Computer Vision*, 634–640, 2003.
6. B. Efron. The Efficiency of Logistic Regression Compared to Normal Discriminant Analysis. *Journal of the American Statistical Association*, 70(352), 892-898, 1975.
7. R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. *IEEE Conference on Computer Vision and Pattern Recognition*, 264–271, 2003.
8. P. T. Fletcher and S. Joshi. Riemannian geometry for the statistical analysis of diffusion tensor data. *Signal Process.*, 87(2), 250–262, 2007.
9. W. Förstner and B. Moonen. A metric for covariance matrices. *Technical report, Dept. of Geodesy and Geoinformatics, Stuttgart University*, 1999.
10. J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *Annals of Statisics.*, 28(2), 337–407, 2000.
11. K. Guo, P. Ishwar, and J. Konrad. Action recognition using sparse representation on covariance manifolds of optical flow. *IEEE International Conference Advanced Video and Signal Based Surveillance (AVSS)*, 188-195, 2010.
12. S. Haykin. Neural Networks: A Comprehensive Foundation. *Prentice Hall, 2nd edition*, 1998.
13. K. Grove and H. Karcher. How to conjugate C^1-close group actions. *Math.Z.*, 132, 11–20, 1973.
14. P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCu. Pedestrian Detection with Unsupervised Multi-Stage Feature Learning. *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.
15. M. Moakher. A differential geometric approach to the geometric mean of symmetric positive-definite matrices. *SIAM Journal on Matrix Analysis and Applications*, 26, 735–747, 2005.
16. S. Munder and D. M. Gavrila. An experimental study on pedestrian classification. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 28, 1863–1868, 2006.
17. P. Papageorgiou and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38(1), 15–33, 2000.
18. X. Pennec, P. Fillard, and N. Ayache. A Riemannian framework for tensor computing. *International Journal of Computer Vision*, 66(1), 41–66, 2006.
19. F. Porikli. Integral histogram: A fast way to extract histograms in Cartesian spaces. *IEEE Conference on Computer Vision and Pattern Recognition*, 829–836, 2005.
20. F. Porikli, O. Tuzel, and P. Meer. Covariance tracking using model update based on Lie algebra. *IEEE Conference on Computer Vision and Pattern Recognition*, 728–735, 2006.
21. W. Rossmann. Lie Groups: An Introduction Through Linear Groups. *Oxford Press*, 2002.

22. H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 20, 22–38, 1998.
23. P. Sabzmeydani and G. Mori. Detecting pedestrians by learning shapelet features. *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
24. R. E. Schapire. The boosting approach to machine learning, an overview. *MSRI Workshop on Nonlinear Estimation and Classification*, 2002.
25. T. Schmah, G. E Hinton, R. Zemel, S. L. Small and S. Strother. Generative versus discriminative training of RBMs for classification of fMRI images. *Advances in Neural Information Processing Systems*, 21, 2009.
26. O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. *European Conference on Computer Vision*, 589–600, 2006.
27. O. Tuzel, F. Porikli, and P. Meer. Pedestrian detection via classification on Riemannian manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10), 1713–1727, 2008.
28. P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *IEEE Conference on Computer Vision and Pattern Recognition*, 511–518, 2001.
29. M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. *European Conference on Computer Vision*, 18–32, 2000.
30. S. Zhang, S. Kasiviswanathan2, P.C. Yuen and M. Harandi. Online Dictionary Learning on Symmetric Positive Definite Manifolds with Vision Applications. *AAAI Conference on Artificial Intelligence*, 2015.
31. Q. Zhu, S. Avidan, M. C. Yeh, and K. T. Cheng. Fast human detection using a cascade of histograms of oriented gradients. *IEEE Conference on Computer Vision and Pattern Recognition*, 1491–1498, 2006.