



**HAL**  
open science

# A Novel Target Detection Algorithm Combining Foreground and Background Manifold-Based Models

Sebastien Razakarivony, Frédéric Jurie

► **To cite this version:**

Sebastien Razakarivony, Frédéric Jurie. A Novel Target Detection Algorithm Combining Foreground and Background Manifold-Based Models. *Machine Vision and Applications*, 2015, pp.14. hal-01131402

**HAL Id: hal-01131402**

**<https://hal.science/hal-01131402>**

Submitted on 13 Mar 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A Novel Target Detection Algorithm Combining Foreground and Background Manifold-Based Models

Sebastien Razakarivony · Frederic Jurie

Received: – / Accepted: –

**Abstract** This paper focuses on the detection of small objects – more precisely on vehicles in aerial images – on complex backgrounds such as natural backgrounds. A key contribution of the paper is to show that, in such situations, learning a target model and a background model separately is better than training a unique discriminative model. This contrasts with standard object detection approaches for which objects vs. background classifiers use the same model as well as the same types of visual features for both. The second contribution lies in the manifold learning approach introduced to build these models. The proposed detection algorithm is validated on the publicly available OIRDS dataset, on which we obtain state-of-the-art results.

**Keywords** Detection · Low Resolution · Vehicles · Database · Aerial Imagery · Infrared

## 1 Introduction

Automatic Target Recognition (ATR), which is the task of automatically detecting targets in images, has a long history in the computer vision community. The primary aim of ATR systems is to assist or remove the human role from the process of detecting and recognizing targets. Two typical applications are surveillance and reconnaissance, which are applications of major importance for safety. As explained by Wong [62], a

surveillance mission over a 200 mile square area with a one foot resolution (30cm), will generate approximately  $1 : 5 \times 10^{12}$  pixels of data. If the area is split in 10 million pixels images, photo interpreters would have to examine over 100,000 images, which is an impractical workload and results in a delayed or incomplete analysis. Furthermore, this delay would allow movable targets to relocate so that they can not be found in subsequent missions.

Contrasting with most of these recent approaches on object detection, this paper focuses on the detection of small rigid targets (such as vehicles), in any arbitrary position, on complex textured backgrounds (see Fig. 2). In addition to the targets size, orientation changes, as well as complex highly textured backgrounds make the task different. Indeed, it strongly contrasts with the dominant trend in object detection which addresses the detection of daily life objects in high quality images, such as the PASCAL VOC challenge [17] – illustrated Fig. 1 – that attracted most of the effort of the community during the 5 last years.

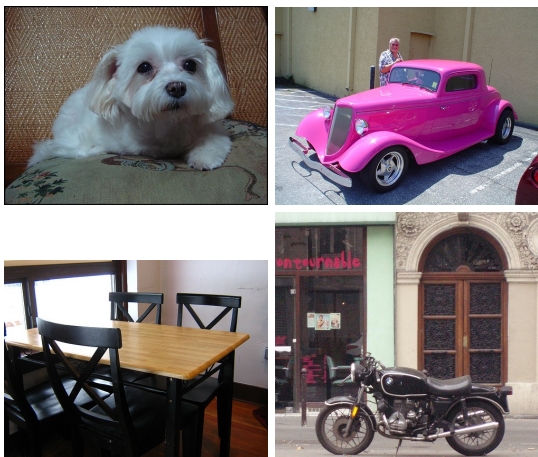
As we deal with surveillance applications, the task is made even more difficult by the fact that some objects can be camouflaged and, in general, do their best not to be detected. Furthermore, it is often difficult to obtain large training sets, as getting images of the desired targets in real conditions is usually costly. Finally, object's context in image (*i.e.* the pixels surrounding the objects) is not strongly correlated with the object itself and we cannot base detections on context, in contrast to other applications where it usually helps.

State-of-the-art methods for object detection rely – in general – on the use of a discriminative classifier trained to learn class boundaries in the representation space. One typical example is the well known Dalal and Triggs's person detector [10], which repre-

---

S. Razakarivony  
University of Caen and SAGEM D.S.  
Tel.: (+33)-158-110-150  
E-mail: sebastien.razakarivony@sagem.com

F. Jurie  
University of Caen  
Tel.: (+33)-231-567-498  
E-mail: frederic.jurie@unicaen.fr



**Fig. 1** Illustrative examples of the PASCAL VOC 2007 challenges [17]. The detection task consists in detecting 20 different classes of objects such as the 'dog', 'car', 'table' or 'motorbike' classes visible in these images.

sents image pixels with Histogram of Oriented Gradient (HOG) features and classifies person vs. background bounding boxes with Support Vector Machine (SVM) classifiers [6].

However, this type of approach does not seem to be relevant to the detection of small targets on complex backgrounds. First, if the background is rich and the number of (positive) training images is limited, learning reliable discriminative features without over-fitting can not be done without strong regularization, which contrasts with the need of having an accurate model of the targets. Second, targets and backgrounds have such different visual properties that it is hard to believe that the same models/features can be adapted to both. Based on these observations, we propose a detection algorithm using two distinct models, one for the background and another for the target, combined to score the candidate windows.

Manifolds are good candidates to model accurately small targets. If a target size is *e.g.*  $40 \times 40$  pixels its visual appearance lies in a 1,600-d space despite the fact that only a small number of parameters (among them: the pose, the illumination, etc.) are sufficient to explain its appearance. Manifolds are precisely adapted to represent high dimensional subspaces that can be generated from a space of fewer dimensions. Supporting this assumption, the work of Zhang [64] shows that images of 3D objects seen from different view-points can be represented as points on a low-dimensional manifold. On the other hand, backgrounds do not require (and can not) be modeled as accurately as targets. Regarding their modeling, we follow the work of [4] and use a PCA-based manifold model. Finally, target and



**Fig. 2** Typical images from the OIRDS dataset. Small size vehicles can have any orientations. Shadows, highlights and complex textured background make the task very challenging.

background models are combined within a probabilistic framework.

This article is an extension of [40], providing a richer description of the related works, more details on the methods as well as a much more extensive experimental validation.

The rest of the paper is as follows: we first introduce the related work in Section 2, then present our approach in Section 3, while finally Section 4 gives a description of the related state-of-the-art algorithms to be compared with, as well as an experimental validation of the proposed vehicle detector.

## 2 Related works

This section reviews different aspects of the target detection task. It first presents state-of-the-art object detectors, explaining how do they encode and classify image regions of interest, within a sliding-window framework. It then introduces some more specific works that make use of *manifolds* in a detection context, as well as presenting the various approaches used for learning the manifolds.

### 2.1 State-of-the-art object detectors

Object detection has attracted a lot of attention in the computer vision community, during the last ten years. We first present the detectors addressing the task *in general* (without any constraints on the size or the type

of objects) and then our attention is drawn to vehicle detectors and small target detectors.

*Generic object detectors.* The sliding window approach is the prevailing approach in the recent literature (see for example [10, 18, 56]). It consists in applying a classifier function to all possible sub-images within an image and taking the maximum of the classification score as indication for the presence of an object in this region. In this manner, the main difference between the different detectors lies in the way images are represented by visual features, as well as in the utilized classifier.

Regarding visual features, one of the most popular one for object detection is the HOG feature (Histogram of Oriented Gradient) [10], that we describe later in the paper. In addition, many other features have been proposed such as Haar-like wavelets [56], edgelets [63], shapelets [44], multi-scale spatial pyramids [2], co-occurrence features [46], covariance descriptors [54], color-self-similarity [57], and also Local Binary or Ternary Patterns [50, 59], Bag-of-Words [8] or finally, some combinations of such features *e.g.* [12, 16, 46, 57, 59].

In regards to classification, two different approaches dominate. One of the most popular is the Support Vector Machine [6] which maximizes the margin between positive examples and negative examples and has good generalization properties. It has been used for example in [10]. The other dominant classifier is boosting (Adaboost [22] or its variant such as Gentle Boost [23]), used in [56] and allowing to use *cascades* of classifiers. Neural networks have also been used for object detection *e.g.* [58].

Several additional ideas have allowed the detector to improve, such as the use of *context learning* *e.g.* [3, 7, 53], new kernels [55], efficient pruning strategies [34], or Deformable Part Models [18].

The Deformable Part Model [18] is certainly the generic detector giving the best results at the moment. It relies on deformable parts and combines a margin-sensitive approach for data mining hard negative examples with a formalism called latent SVM. The DPM requires objects' parts to be big enough to be detected and is therefore not adapted to the detection of small targets.

Some more recent work try to learn the features and the classifier. It is all the algorithms based on convolutional networks [35]. Even if these networks are an old technique, they achieved great performances only recently [31, 48].

Finally, it is worth pointing out that if object detection has progressed in the recent years, it is also because several datasets allowing the detectors to be trained and

evaluated have been made publicly available. Among them, we can cite the PASCAL VOC dataset [17], the CALTECH dataset [13], the MIT face dataset [41] or the ETHZ dataset [20].

*Vehicle detection.* More directly related to our problem, some approaches have been specially designed for the detection of vehicles. In [66], Zhao and Nevatia pose car detection as a 3D object recognition problem to account for the variations due to viewpoint and shadow changes. They selected the boundary of the car body, the boundary of the front windshield and the shadow as the features, this information being represented by a Bayesian network used to integrate all features. Their experiments show promising results on challenging images, but the cars that are not on roads seem not to be well detected. Eikvil *et al.* [15] use several different features combined with Linear Discriminant Analysis [21]. A segmentation step, followed by two stages of object classification is used. Their work is done in the context of multispectral and panchromatic images, and explicitly assumes vehicles positions are related to the road network. They show interesting results, however the vehicles are so small that they can not be detected without assuming they are on roads, explaining why the road network is needed. In [49], Stilla *et al.* propose different algorithms adapted to the different sensors they use (color, thermal infrared, radar). They also build local and global features from a 3D model, and use context. [29] reports interesting vehicle detection results, obtained by using large and rich sets of application-specific image descriptors. The features are based on several geometric and color attributes representative of the vehicles, and perform a Partial Least Square analysis on them. They compare their approach to HOG-SVM-like classifiers [10], obtaining similar performance. Other works have addressed the detection of small vehicles such as [15, 67]. However they required the assumption that vehicles are on roads to make the detection easier and hence cannot be used in our context. Finally, it is worth noting that none of their experiments can be reproduced because neither the protocols nor the datasets are available.

*Small object detection* There are very few papers addressing the detection of small objects. These papers are often based on the detection of *salient regions*. The objects to be detected are then defined as the regions of the image which do not have the same statistics as the background *e.g.* [43, 47]. Among the rare papers which tried to model small targets explicitly, we can mention the work of [37], which – in addition to introducing a new dataset of  $36 \times 18$  pixels pedestrian images – have

shown that good performance can be obtained by combining standard features such as Haar wavelets or HOG features with SVM/boosting classifiers [16].

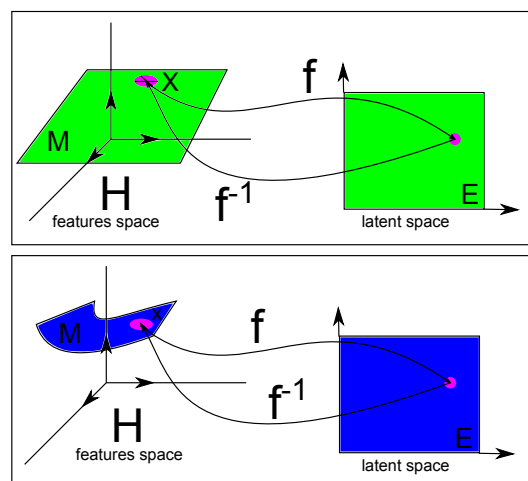
## 2.2 Manifolds used for object detection

*Modeling object appearance with manifolds.* A manifold is a subspace embedded in a higher dimension space which can be locally approximated by an Euclidean subspace. It can be represented by a Euclidean subspace, called the *latent space*. The geodesic distances, which are the shortest paths between two points inside the manifold, are locally preserved in the latent space. Fig. 3 gives an illustration by showing the relationship between the manifold ( $E$ ) and the Euclidean space ( $H$ ). Manifold learning has been used by several authors to model object appearance in the context of detection tasks. One of the most famous approaches is the Eigenfaces of Pentland *et al.* [39], using Principal Component Analysis [27] to build linear face manifolds that can be used for face detection, extended to the detection of hands by [36]. The main idea is to perform a Principal Components Analysis of the positive examples. Keeping only a few eigenvectors, which are called Eigenfaces/Eigenhand, it is possible to easily compute the distance to the object manifold (called Distance to Feature Space).

Based on the same mechanisms, [4] uses PCA for object detection by modeling background and objects with linear manifolds. Interesting results are reported on good quality images of cars and pedestrians, when using high-dimensional manifolds. In [19], the authors use autoencoders to build face manifolds for face detection. However, this approach is restricted by the lack of background model, giving a lot of false alarms.

Our approach builds on all these recent works by using the best current image features within a manifold learning framework. The contribution of the paper lies in the combination of two types of manifolds, namely autoencoders for the targets and linear manifolds for the backgrounds. As far as we know, this is the first time such a model has been proposed.

*Manifold learning.* Several different types of algorithms can be used to learn a manifold. Some of them are linear methods, such as the Linear Discriminant Analysis [21], the Independent Component Analysis [5] or the simple Principal Components Analysis [27]. For non linear methods, some of them are based on the preservation of geodesic distances such as the famous Isomap [52] derived from Multidimensional Scaling [32] or the Diffusion Maps [33]. Another alternative is to learn linear approximations of the manifold locally, which is



**Fig. 3** This figure give an illustration of the concept of *manifold*. The vectors belonging to the original Euclidean space  $H$  are spanned by the subspace  $E$ . The manifold on the top is linear while other one is non-linear.  $f$  is a mapping function, giving the correspondence between a point in the manifold in the original space and its projection in the subspace.

done by the Local Linear Embedding [45], the Hessian LLE [14], the Local Tangent Space Analysis [65] or the Locality Preserving Projections [38]. Finally, other approaches learn the manifold in a global way, such as the Maximum Variant Unfolding [61, 60], or autoencoders [30]. However, many of these algorithms have been designed to visualize high-dimensional data in 2D and only give the mapping  $f$  (see figure 3), and not its inverse (required by our approach for the detection task, as explained later). Interestingly, Principal Components Analysis and its variants and Autoencoders can be used to compute both  $f$  and  $f^{-1}$ .

## 3 Our approach

We built our approach on the standard *sliding window* framework and *manifold learning* algorithms. The contribution of the approach lies in the model used in the scoring function.

We follow the classic sliding window pipeline, which is designed as follows: all the possible rectangular regions (at any position and any scale) of a given aspect ratio are evaluated, one by one, by an object classifier. This is done in practice by using a multi-scale grid (typically with a step-size of 8 pixels and a ratio of  $2^{\frac{1}{10}}$  between each scales such as done by [18]). The aspect ratio of the sliding window is fixed. For objects with fixed width-to-height ratios (*e.g.* faces) this is not a problem. However, if the appearance of the object can varies a lot (for example the front and size view of a

car have very different aspect ratios) several distinct classifiers are usually learned, one per aspect ratio, after clustering the set of training images accordingly to their aspect ratio. In practice, in our experiments we use only one single aspect ratio, computed, by averaging the aspect ratios of the training images.

In this section we focus on the classifier, and more specifically on the models which represent the appearance of the target and the appearance of the background.

As explained in the introduction, we suggest using two distinct probabilistic models, one for backgrounds, another for objects. The score of a candidate window is computed as their log-likelihood:

$$S(X_s) = \log \left( \frac{p_{obj}(X_l = obj|X_s)}{p_{back}(X_l = back|X_s)} \right) \quad (1)$$

where  $X_l$  is the unknown class of the window ( $X_l \in \{obj, back\}$ ) and  $X_s$  is the signature of the window (*i.e.* a visual descriptor, such as the HOG descriptor). Probabilities  $p_{obj}$  and  $p_{back}$  are given by the object and background models respectively. Please note that, as we have two distinct models,  $p_{obj} \neq 1 - p_{back}$ , which contrasts with standard approaches using a single model.

Both classes (objects and background) are modeled by manifolds learned during a training stage. Let  $H$  denotes the signature space (we use terms *signatures* and *visual features* indistinctly) and  $X_{s,t} \in H$  the set of training signatures representative of a class. We recall that building a Riemannian manifold  $\mathcal{M}$  representative of these signatures is equivalent to finding a function  $f$ , such as:

$$\forall X_{s,t} \in \mathcal{M}, \exists ! Y \in \mathcal{R}^n, Y = f(X_{s,t}) \quad (2)$$

$f$  is called the embedding of  $\mathcal{M}$ , and is an isometric function. In some works the notation LOG is used instead of  $f$  noted and EXP instead of the inverse of  $f$ . However, as we do use regular log and exp functions later, we use the  $f$  and  $f^{-1}$  notation to avoid confusion.

Obviously, if  $X_s$  lies on the manifold,  $f^{-1} \circ f(X_s) = X_s$ .  $f^{-1} \circ f$  projects any point of the input space on the manifold  $\mathcal{M}$ . By denoting  $P_{\mathcal{M}} = f^{-1} \circ f$ , we can define the distance to the manifold by:

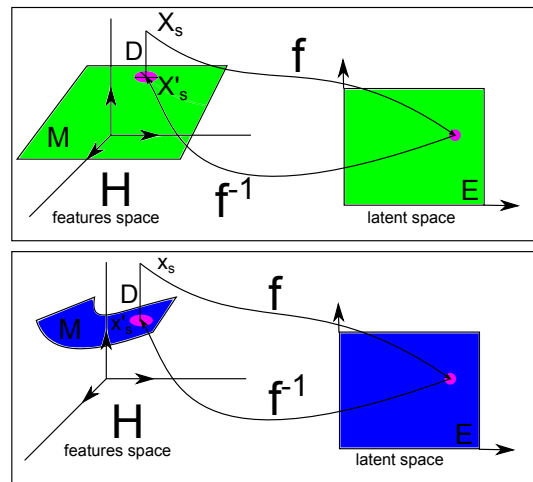
$$D_{\mathcal{M}}(X_s) = \|X_s - P_{\mathcal{M}}(X_s)\| \quad (3)$$

where  $\|y\|$  represent the Euclidean norm of  $y$ .

Finally, we use this distance to derive the probability for a signature  $X_s$  to be generated by the manifold  $\mathcal{M}$ :

$$p(X_s \in \mathcal{M}|X_s) = \alpha \exp \left( -\frac{D_{\mathcal{M}}(X_s)^2}{\sigma_{\mathcal{M}}^2} \right) \quad (4)$$

where  $\alpha$  is a normalization factor and  $\sigma_{\mathcal{M}}^2$  a parameter of the model. In practice, as scores are given by



**Fig. 4** Illustration of the concept of *distance to the manifold*.  $X_s$  is a visual signature and  $X'_s = f^{-1} \circ f(X_s)$  is its projection on the manifold. On the top part of the figure the manifold is linear while on the bottom part it is non linear.

a likelihood ratio (eq. (1)) and as we are only interested in ranking candidate windows, the normalization factor can be ignored. The only remaining parameter is the object/background ratio of  $\sigma_{\mathcal{M}}^2$ , estimated by cross-validation. An illustration of the distance to the manifold is presented in figure 4.

Following Eq (1), the score is given by the likelihood ratio, which is in this case:

$$S = \frac{D_{mback}(X_{obs})^2}{\sigma_{mback}} - \frac{D_{mobj}(X_{obs})^2}{\sigma_{mobj}} \quad (5)$$

Where  $D_{mback}$  (resp.  $D_{mobj}$ ) is the distance to background manifold (resp. to the target manifold). The ratio of the two variances  $\frac{\sigma_{mback}}{\sigma_{mobj}}$  is set by cross-validation.

*Object manifolds.* Object manifolds are learned by autoencoders [30]. Indeed, in addition to being reported as being efficient for several computer vision tasks, they make the computation of  $f$  and  $f^{-1}$  (which are both required by previous equations) possible. Furthermore, they allow very expressive models, whose complexity can be adapted by varying their number of layers (3 in our case) and hidden neurons (fixed by cross-validation in our experiments), to be built. Autoencoders are non linear versions of the Principal Components Analysis.

Autoencoders (Fig. 5) are symmetrical neural networks, which learn the identity function under constraints. The simplest autoencoder is made of 2 layers in addition to the input layer (bottom row of Fig. 5). One neuron from the layer  $i$  is connected to all the neurons of layer  $i+1$ , and only to these neurons. We denote as  $W_{ij}$  the matrix of weights between the layer  $i$  and the layer  $j$ . The layers are numbered from 0 (input) to

$N$  (middle layer) and then back from  $N$  (middle layer) to 0 (output), as shown in Fig. 5. As the network is symmetrical, we have:

$$\text{dimension}(W_{ji}) = \text{dimension}(W_{ij}^T) \quad (6)$$

Each layer  $j$  has an output  $y$ , given the layer input  $x$ :

$$y = h(W_{ij}x) \quad (7)$$

$h$  is called the *activation function*, and is typically the sigmoid function. When the activation function  $h$  is linear for all the layers, the autoencoder computes a PCA [30]. Contrary to this, using non-linear  $h$  functions allow the network to approximate any function [9].

To learn an autoencoder efficiently, it has to be initialized, as any neural network. In order to do so, the  $W_{ij}, \forall i = 0..N - 1$  are initialized separately, from the input layer to the middle layer. Each pair of layers and its associated weights are considered as a Restricted Boltzmann Machine [1], and the weights between the layers are initialized using the contrastive divergence algorithm [25]. We do not give more details here, as this is not the main topic of the paper, encouraging the reader to read [26] for further details. Once this step is done, the network is *unfolded*, meaning that the weights of the first layers are used to initialize the weights of other layers:

$$W_{ij} = W_{ji}, \forall i = 0..N - 1 \quad (8)$$

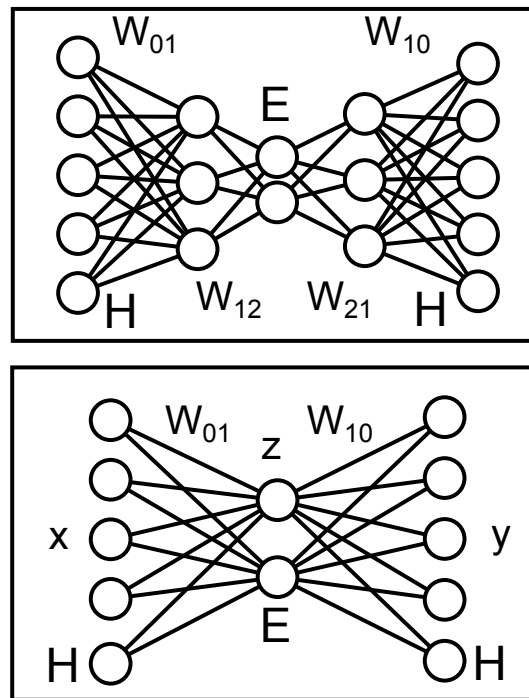
Finally, a standard back-propagation of the error can be done [42] to jointly learn  $f$  and its inverse. The latent space  $E$  can be accessed easily by reading the output of the middle layer.

The autoencoder is trained by minimizing the reconstruction error of training examples:

$$\text{Error} = \sum_{X_{s,t} \in \text{train}} \|X_{s,t} - g \circ f(X_{s,t})\|^2 \quad (9)$$

where  $f$  is the function connecting the input to the central layer of the autoencoder, and  $g$  the function connecting the central layer to the output.

$g \circ f$  is then equivalent to the previously mentioned  $f^{-1} \circ f$ . In the context of manifold learning, the network is usually used to learn  $f$  and  $f$  only, providing an embedding of data [26]. In contrast, we keep the full network, which gives us the projection  $P_{\mathcal{M}}(X_s)$  we are trying to find. In practice, we use sigmoid activation functions and train autoencoders, after doing a contrastive divergence initialization [25], with a standard back-propagation algorithm. Contrastive divergence is the key to good results, as it helps the neural network to focus on data that were given (instead of the identity function). In addition and to learn the manifold, we take a representative set of training windows, compute their signatures and optimize autoencoder parameters as explained above.



**Fig. 5** Two simple autoencoders.  $H$  is the input space and  $E$  is the latent space. Top: a 5 layers autoencoder. Bottom: the minimal autoencoder made of 3 layers.

*Background manifold.* Modeling background with too complex models (*i.e.* using models with too many parameters), would be risky in terms of over-fitting. Our hypothesis is that linear models such as the PCA is best suited to model backgrounds. PCA finds a subspace which minimizes the reconstruction error. We can use a limited number of principal components and project the data on them very easily. A signature  $X_s$  can be written as  $X_s = \sum \beta_i * PC_i$  where  $\beta$  is the representation of  $X_s$  in the PCA-projected space ( $PC_i$  are the principal component). We can then project  $X_s$  into a  $N$ -dimensional subspace using the  $N$  first principal components.

$$X_s = \sum_{i=1:N} \beta_i PC_i + \sum_{j=N+1:M} \beta_j PC_j = P(X_s) + \bar{P}(X_s) \quad (10)$$

$P(X)$  is the projection of  $X$  on the manifold while  $\bar{P}(X)$  is the projection on the space orthogonal to manifold. Interestingly,  $\|\bar{P}(X)\|$  is the distance to the manifold, which is proportional to the mean square reconstruction error. This can be seen as an approximation of the distance to the manifold given by the first  $N$  components, as  $D_{\mathcal{M}}(X_s) = \bar{P}(X_s)$ . In our experiments, we randomly sample background windows from training images, compute their signatures and find the best basis by doing a SVD decomposition of their covariance matrix.

*Post and pre processing.* Our algorithm can use any type of image features as input. For the post processing, as for any usual sliding-window approach, a non-maximum suppression is needed. We use a simple and efficient iterative greedy strategy consisting in keeping only the windows which have the maximum scores over a disk (which radius is half the window width). We set the windows so that they do not overlap more than 50 percent.

## 4 Experiments

### 4.1 Dataset and protocol.

This section presents experiments on the OIRDS dataset [51], which is one of the rare publicly available dataset for Automatic Vehicle Detection with aerial images.

OIRDS contains a set of approximately 1,000 aerial images coming from different sources (*e.g.* USGS and VIVID), for a total of about 1800 targets. Shadows, specularities, occlusions, as well as the large intra-class variability (*e.g.* regular cars, pickups, mini-vans, etc.) make this dataset challenging. The dataset is provided with rich annotations: distance from the camera to the ground, target size (in pixels), bounding boxes, percentage of occlusion, type, etc. are given for each vehicle. Fig. 2 shows typical images of this dataset. As this dataset is very heterogeneous, we have separated a dozen of images that were very different by their size. This set will be further reported as the *large-images set*. The rest of dataset is split in 10 folds and we evaluate the performance using a 10 fold cross validation procedure, by reporting the mean average precision (we use the experimental protocol of [17]).

As we are primarily interested in knowing the performance of our detector for small targets detection, images were downscaled to produce a dataset in which targets are not bigger than  $40 \times 40$  pixels. It will be called the *small-images set* (in opposition to the *large-images set*).

### 4.2 Baseline algorithms

Unfortunately, no reproducible results have been published so far on this dataset (nor on any publicly available dataset for small target detection, at least to our knowledge). In consequence, we compared the performance of our algorithm to different baselines.

First, we implemented the algorithm of [10], which is an SVM-HOG sliding windows algorithm known to obtain state-of-the-art results on such tasks. This algo-

rithm is presented section 2. The linear SVM classifier is taken from the *svmlight* library [28].

In addition, we have also implemented a generative model based on a Gaussian Mixture Model (GMM), which is a reference for generative models. The Gaussian mixture model learns a Gaussian mixture having  $N$  Gaussian components computed to best fit positive data.  $N$  is set by cross validation. In this case, the background is modeled by a GMM as well. Both models are learned using the Expectation-Maximization algorithm [11].

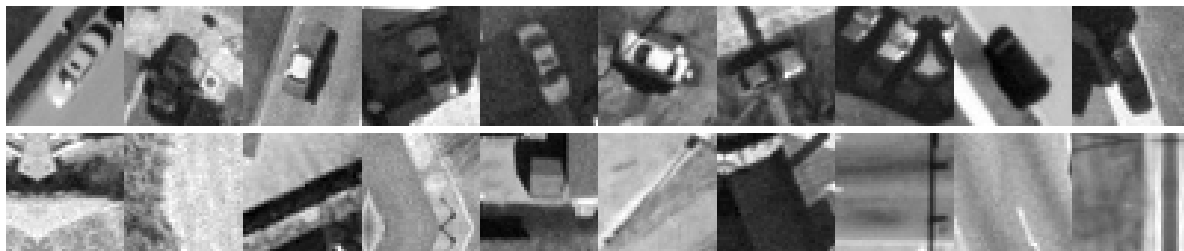
In order to provide comparisons with more powerful and more recent methods, we also experimented a Convolutional Neural Network architecture [35], as well as the popular Deformable Part Model [18], using the released code provided by Felszenswalb (we used release 5). It is worth pointing out that the CNN takes grayscale windows as input, learning its own features, and that the DPM uses HOG31 features. The CNN we used is composed of a convolutional layer, followed by a downsampling layer, and finally, two fully connected layers. We avoid using more layers, as the number of examples is not sufficient to avoid over-fitting (we would have too few examples regarding to the number of parameters).

Finally, to illustrate the importance of using different models for targets and backgrounds, we compare our approach – which has a PCA-based model for the background and an autoencoder for the targets – to the same approach but using PCA/autoencoder for both models (same model to represent backgrounds and targets).

Training data is obtained by cropping the positive examples of the folds used for training (remember that a 10-folds cross validation procedure is used) and 13,000 negative windows randomly sampled from the background (having no overlap with the targets). Moreover, the training set is extended by adding positive examples obtained by flipping up/down/left and right and by rotating the initial training set. This gives more positive examples to learn the model, resulting in a total of about 3,800 positive examples per fold. As the step size of our sliding window is of 8 pixels, when we crop positive images for training, we add a random shift up to 4 pixels to make the model more tolerant to small shifts. Some typical positive and negative training examples are given Fig. 6.

Regarding image signatures, we experimented with three different signatures: (i) normalized raw level intensities, often used in the literature for the detection of small targets (ii) image gradients, which are more robust to illumination changes and (iii) HOG features, usually considered to be the best choice for this task.





**Fig. 6** Illustrative images of the OIRDS dataset. First row: 10 typical target-centered regions. Second row: some background regions.

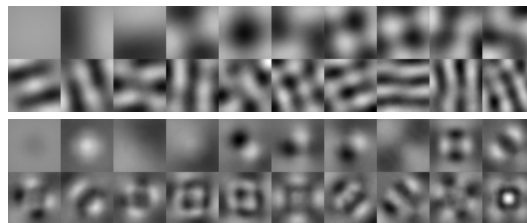
In practice, raw pixel intensities are computed as the mean of the three color channels (OIRDS images are in color). Gradient images are computed by a Sobel filter. Finally, HOG31 is an histogram of oriented gradient, with 8 pixels overlap cells of  $16 \times 16$  pixels. They contain a 9 bins histogram of unsigned orientations concatenated with a 18 bins histogram of signed orientations. It is then normalized according to the neighbor histogram, as explained in [18].

The manifold dimensionality of backgrounds models is of 40, 10 and 16 for intensity, gradient and HOG signatures respectively. Autoencoders have 3 layers and have respectively 35, 8 and 10 inner nodes for intensity, gradient and HOG signatures. All these parameter values were determined by preliminary experiments on toy data and remained fixed for the final experiments.

### 4.3 Complexity Analysis

The computation complexity is crucial when addressing automatic target recognition tasks. In practice, we are more interested by the complexity of the test stage, as learning can be done offline and once for all. All the tested algorithms have a linear complexity with respect to the number of windows in the sliding window process. In addition, all the algorithms using a latent space (PCA, autoencoders, etc.) have a complexity which is also linear with respect to the dimensionality of the latent space (*i.e.* doubling the dimensionality of the latent space will result in doubling the number of operations). Finally, all the algorithms have a linear complexity with respect to the dimensionality of the input features.

Regarding the computational time, all the experimented algorithms can be implemented with simple operations (sums of products and lookup table for complex functions) in such a way that they can be real-time.



**Fig. 7** The 20 principal components of two PCA models: a background model (two first rows), a car model (two second rows).

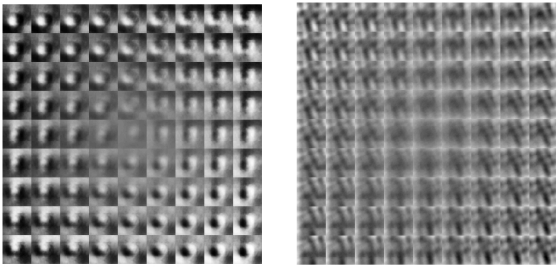
### 4.4 Qualitative results and visualizations

*Visualizing the models.* Fig. 7 shows the first principal components of (i) the PCA background model and (ii) of a PCA car model, using raw pixel intensities as input. We can note that the background model contains more low frequency textures than the car model, and that the car model's first components can reconstruct centered objects on a uniform or two-color background. The principal components of the background models are typical of natural images (see [24]).

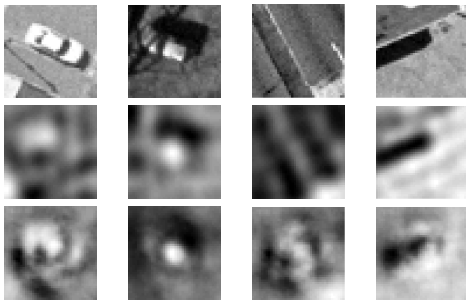
Fig. 8 shows some typical templates, which our autoencoder can generate once trained with car images (left hand-side). The right-hand side has been generated with an autoencoder learned with background regions. In this case, the autoencoder focuses on intensity differences, and does not learn any rotation, as we can expect.

Fig. 9 shows candidate windows (1st row), their projections on the background manifold (obtained by PCA, 2nd row), as well as their projections on the car manifold given by an autoencoder (last row). As it can be noticed, target images are better reconstructed by the target model than by the background one, and vice-versa.

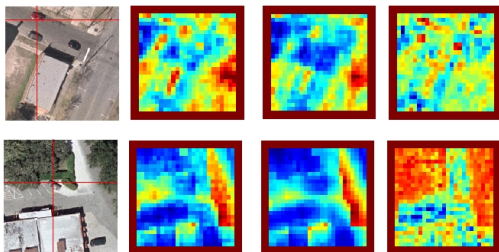
*Error maps.* Fig. 10 shows the *error maps* (*i.e.* image in which the color of a pixel represents the distance between the region centered on this pixel and the man-



**Fig. 8** Visualization of the target (left) and background (right) manifolds learned by the autoencoders.



**Fig. 9** Four candidate windows (first row), followed by their reconstruction by the PCA background manifold (2nd row), and by the target manifold learned by an autoencoder (last row).



**Fig. 10** From left to right: (i) the original image with the maximum of likelihood pixel marked by a red cross, (ii) the negative of the log error map given by the autoencoder, (iii) the negative of the log error map given by the PCA, (iv) the final log likelihood ratio. The negative of the log error map  $-\log(E)$ , where  $E$  the reconstruction error is used for better visualization.

ifold, see section 3 for details), according to a PCA-background model or and autoencoder-target model. We can see that autoencoder alone gives many false positives, and needs to be balanced with the PCA model. This is due to the fact that the manifold learned by the autoencoder can generate uniform patterns, which are in fact vehicles with uniform intensity. This can easily be removed thanks to the PCA, as an uniform background is easily reconstructed by its first few principal components.

*Detection results* Fig. 13 shows the top false positives (those having the highest scores) obtained with the SVM detector and with the AE-PCA detector. The figure also shows the most difficult targets, *i.e.* those with the lowest scores. Both are obtained on one specific fold. Each window is enlarged a little bit to reveal its context (*i.e.* the surrounding pixels). We can notice that for both SVM and AE-PCA, with HOG features, some hard negative windows looks to be positive, however the localization is not accurate enough. This is because the HOG features is computed using pixels that are a little outside the sliding window. We can also see that the false positives are very different for SVM and AE-PCA, but the lowest scored positives have some identical targets. It should be noted that specularities or heavy shadows highly perturb the detection.

#### 4.5 Setting up the parameters.

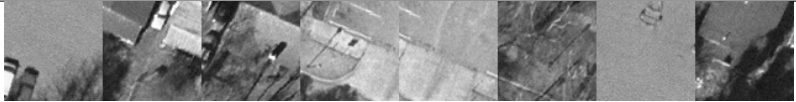

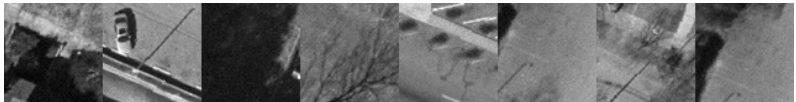
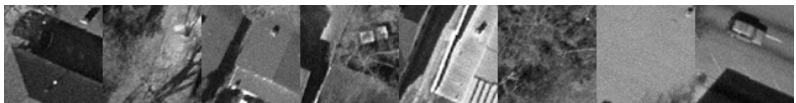
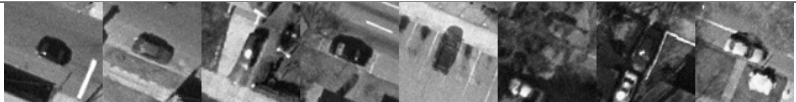
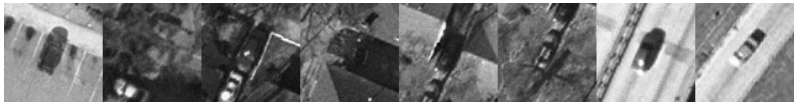
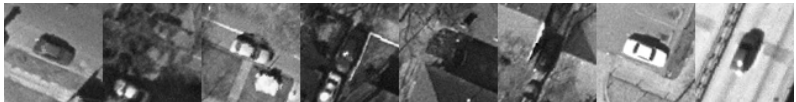
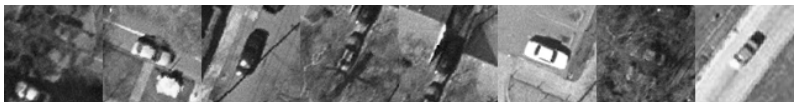
Different parameters have to be set, in particular dimensionality of the latent space for the PCA, the parameters of autoencoders, or the number of components of the Gaussian Mixtures. We choose to fix them by cross validation, on a validation set. Fig. 11 shows the performance of the AE-PCA detector, for different dimensionalities of the latent space, when using raw pixel intensities as input. We can see that there is a correlation between both, with a flat optimum of the parameters.

The refinement step (*i.e.* the back-propagation step described in section 3) is crucial to obtain good results, as can be seen Fig. 12. We can see that going through the refinement step allows significant performances to be gained. We can also see that the optimum is decreasing after 45 hidden neurons. Indeed, when this number is too large, the constraint ensuring the autoencoder learns the vehicles are too smooth, explaining why it only learns the identity function.

It is important to notice that, as shown by Fig. 14, reducing the reconstruction error on the targets by adding neurons might results in over-fitting. At the end, adding neurons is equivalent to removing constraints, and the network becomes the identity function. Having a small reconstruction error (which can be seen as a low training bias) is not a guarantee of the quality of the detector, in any case.

#### 4.6 Quantitative results.

We have experimented with 7 different detectors. The first one (so called AE-PCA) is the proposed one, using an autoencoder to model targets and a PCA based

Classifier	Features	Top false positive windows
SVM	Raw Intensity	
AE-PCA	Raw Intensity	
SVM	HOG	
AE-PCA	HOG	
Classifier	Features	Top false negative windows
SVM	Raw Intensity	
AE-PCA	Raw Intensity	
SVM	HOG	
AE-PCA	HOG	

**Fig. 13** Visualization of the top false positive windows (*i.e.* the negative regions with highest scores), on the 4 first rows, and the top false negative windows (*i.e.* the positive windows with lowest scores), on the 4 last rows. All these images come from a single fold.

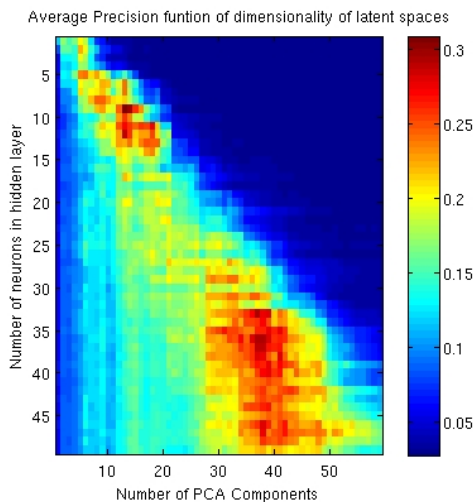
manifold for backgrounds. The second is one of the state of the art approaches for detection, namely the Dalal and Triggs’s detector [10] (so called HOG-SVM). We also compared our approach with the Deformable Part Model of Felzenszwalb et al [18], as well as a Convolutional Neural Network. In addition, we have also experimented with three other detectors: the first one used PCA to model both targets and backgrounds (PCA-PCA), the second one used Gaussian mixture model, here again for both targets and backgrounds (GM-GM), and the last one used an autoencoder for both as well (AE-AE). For these 7 detectors, we report the mean average precision over the 10 folds of the OIRDS datasets in Table 1.

The main conclusion we can draw from these results is that the proposed approach (the AE-PCA detector) outperforms any other detector, for any type of feature. The best results are obtained with HOG31 signatures. We also observe that Gaussian Mixture Models do not perform well in any case. Indeed, we have noticed that

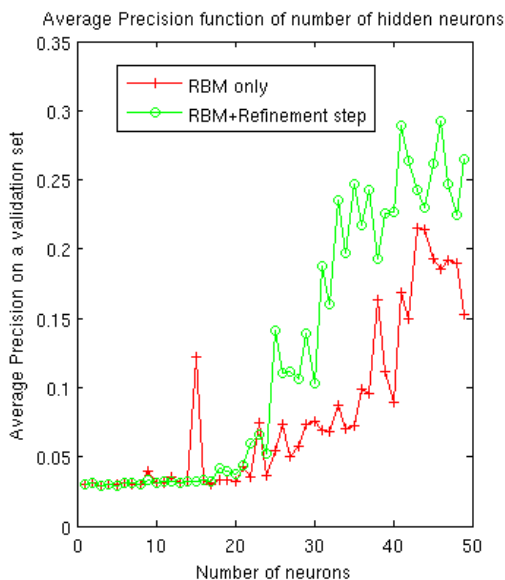
the GMM tends to be specialized to a few images, showing that EM gets stuck in local minima. From these results, we can also conclude that the HOG-SVM detector is outperformed – when using gradient and gray level signatures – by the PCA-PCA detector. HOG-SVM is however better than PCA-PCA with HOG31 features. The DPM was not able to give good results, as the code was not designed to deal with so small targets. Hard negative mining is done on images with no vehicles, so we added aerial images of background to help the DPM, but it was not sufficient.

In addition, we can also observe that using two autoencoders (one for the targets, the other for the backgrounds) does not give better results, as the background autoencoder fails to capture the diversity of the backgrounds.

Finally, as mentioned before, we kept aside a dozen images that are more difficult because of their large size, for additional experiments (using the previously learned classifiers). Targets are as small as previously but the



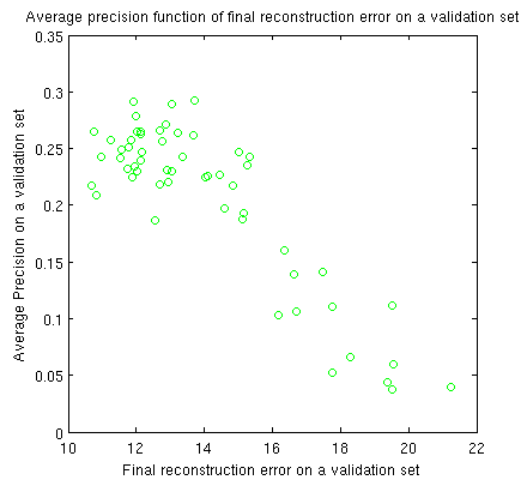
**Fig. 11** Mean Average precision on a validation test, for several pairs of (number of PCA components, number of neurons in the hidden layer).



**Fig. 12** Mean Average precision on a validation set, as a function of the number of neurons of the hidden layer, for RBM and for RBM +back-propagation.

images cover a much larger area and hence produce more false positive. Figure 15 illustrates the difficulty of these images by presenting two regions of interest, one being a positive example (*i.e.* a vehicle) the other being a negative one. Even a human can hardly predict which one is the positive one. Such large images hence make the task even more challenging.

Results are given in Table 2. The performance is not as good as on the regular OIRDS images, as expected, as the images are much larger and include more dis-



**Fig. 14** Each point of this plot represents the average precision of the detector as function of the average reconstruction error during training. Each point correspond to a fixed autoencoder with a fixed number of neurons. The reconstruction error is as low as the number of neurons is large.

	Intensity	Gradient	HOG31
GMM-GMM	8.3%	21,3%	17.7%
HOG-SVM [10]	10.5%	35.2%	46.8%
DPM [18]	–	–	6.55%
CNN	34.1%	–	–
PCA-PCA	35.0%	37.9%	42.5%
AE-AE	35.3%	33.5%	47.5%
AE-PCA (ours)	<b>35.5%</b>	<b>39.9%</b>	<b>48.9%</b>

**Table 1** Mean Average Precision on OIRDS for the five detectors we experimented.

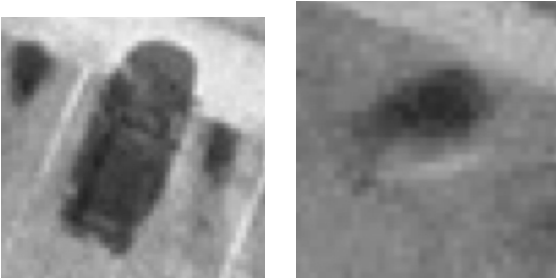
	Intensity	Gradient	HOG31
HOG-SVM [10]	1.5%	12.1%	12.6%
AE-PCA (ours)	<b>3.3%</b>	<b>16.4%</b>	<b>17.1%</b>

**Table 2** Mean Av. Precision on the large images.

tractors without containing more targets, *i.e.* there are more possible false positive without having more true positives. Here again our AE-PCA detector clearly outperforms any other approaches.

## 5 Conclusions

This paper proposes an algorithm for the detection of small targets on complex backgrounds, based on manifold learning. The core of our contribution lies in a new scoring function in which targets and background are modeled by distinct and adapted models. Targets are accurately modeled by an off-line learned autoencoder while background is modeled by a PCA based linear manifold. We have experimentally validated our ap-



**Fig. 15** Two region of interest extracted from the large images. Among these two, one is a positive example while the other one is a negative one.

proach on a publicly available vehicle dataset, and show results that outperform state-of-the-art algorithms.

**Acknowledgements** This work was supported by Agence Nationale de la Recherche et de la Technologie, through the CIFRE sponsorship No 2011/0850 and by SAGEM-SAFRAN group.

## References

1. Ackley DH, Hinton GE, Sejnowski TJ (1985) A learning algorithm for boltzmann machines. *Cognitive Science* 9:147–169
2. Bosch A, Zisserman A, Muoz X (2007) Representing shape with a spatial pyramid kernel. In: *Conference on Image and Video Retrieval*, pp 401–408
3. Carbonetto P, De Freitas N, Barnard K (2004) A statistical model for general contextual object recognition. In: *European Conference on Computer Vision*, pp 350–362
4. Carvalho GV, Moraes LB, Cavalcanti GD, Ren TI (2011) A weighted image reconstruction based on pca for pedestrian detection. In: *International Joint Conference on Neural Networks*, pp 2005–2011
5. Comon P (1994) Independent component analysis, a new concept? *Signal processing* 36:287–314
6. Cortes C, Vapnik V (1995) Support-vector networks. *Machine Learning* 20:273–297
7. Crandall DJ, Huttenlocher DP (2007) Composite models of objects and scenes for category recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp 1–8
8. Csurka G, Dance C, Fan L, Willamowski J, Bray C (2004) Visual categorization with bags of keypoints. In: *European Conference on Computer Vision*, p 22
9. Cybenko G (1989) Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems* 2:303–314
10. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp 886–893
11. Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society* 39:1–38
12. Dollár P, Tu Z, Perona P, Belongie S (2009) Integral channel features. In: *British Machine Vision Conference*, p 5
13. Dollár P, Wojek C, Schiele B, Perona P (2009) Pedestrian detection: A benchmark. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp 304–311
14. Donoho DL, Grimes C (2003) Hessian eigenmaps: Locally linear embedding techniques for high dimensional data. *Proceedings of the National Academy of Sciences* 100:5591–5596
15. Eikvil L, Aurdal L, Koren H (2009) Classification-based vehicle detection in high-resolution satellite images. *Journal of International Society for Photogrammetry and Remote Sensing* 64:65–72
- 16.ENZWEILER M, GAVRILA D (2009) Monocular pedestrian detection: Survey and experiments. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 31, pp 2179–2195
17. Everingham M, Gool LV, Williams, I CK, Winn J, Zisserman A (2010) The pascal voc challenge. *International Journal of Computer Vision* 88:303–338
18. Felzenszwalb P, Girshick R, Mcallester D, Ramanan D (2009) Object detection with discriminatively trained part based models. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 32, pp 1627–1645
19. Feraud R, Bernier O, Viallet J, Collobert M (2001) A fast and accurate face detector based on neural networks. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 23, pp 42–53
20. Ferrari V, Tuytelaars T, Van Gool L (2006) Object detection by contour segment networks. In: *European Conference on Computer Vision*, pp 14–28
21. Fisher RA (1936) The use of multiple measurements in taxonomic problems. *Annals of eugenics* 7:179–188
22. Freund Y, Schapire RE (1995) A decision-theoretic generalization of on-line learning and an application to boosting. In: *Computational learning theory*, pp 23–37
23. Friedman J, Hastie T, Tibshirani R (2000) Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors).

- The annals of statistics 28:337–407
24. Hancock PJ, Baddeley RJ, Smith LS (1992) The principal components of natural images. *Network: computation in neural systems* 3:61–70
  25. Hinton G (2000) Training products of experts by minimizing contrastive divergence. *Neural Computation* 14:2002
  26. Hinton GE, Salakhutdinov RR (2006) Reducing the Dimensionality of Data with Neural Networks. *Science* 313:504–507
  27. Hotelling H (1933) Analysis of a complex statistical variable into principal components. *Journal of educational psychology* 24:417
  28. Joachims T (1999) Making large-scale support vector machine learning practical. <http://svmlight.joachims.org/>
  29. Kembhavi A, Harwood D, Davis LS (2011) Vehicle detection using partial least squares. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 33, pp 1250–1265
  30. Kramer M (1991) Nonlinear principal component analysis using autoassociative neural networks. *American Institute of Chemical Engineers Journal* 37:233–243
  31. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*, pp 1097–1105
  32. Kruskal JB (1964) Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika* 29:1–27
  33. Lafon S, Lee AB (2006) Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 28, pp 1393–1403
  34. Lampert CH, Blaschko MB, Hofmann T (2008) Beyond sliding windows: Object localization by efficient subwindow search. In: *IEEE Conference on Computer Vision and Pattern Recognition*
  35. LeCun Y, Bengio Y (1995) Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks* 3361
  36. Moghaddam B, Pentland A (1997) Probabilistic visual learning for object representation. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 19, pp 696–710
  37. Munder S (2006) An experiment study on pedestrian classification. vol 28
  38. Niyogi X (2004) Locality preserving projections. In: *Neural information processing systems*, vol 16, p 153
  39. Pentland A (1994) Viewbased and modular eigenspaces for face recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp 84–91
  40. Razakarivony S, Jurie F (2013) Small target detection combining foreground and background manifolds. In: *IAPR International Conference on Machine Vision and Application*
  41. Rowley HA, Baluja S, Kanade T (1998) Neural network-based face detection. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 20, pp 23–38
  42. Rumelhart DE, Hinton GE, Williams RJ (1985) Learning internal representations by error propagation. Tech. rep., DTIC Document
  43. Rutishauser U, Walther D, Koch C, Perona P (2004) Is bottom-up attention useful for object recognition? In: *IEEE Conference on Computer Vision and Pattern Recognition*, vol 2, pp II–37
  44. Sabzmejdani P, Mori G (2007) Detecting pedestrians by learning shapelet features. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp 1–8
  45. Saul L, Roweis S (2003) Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research* 4:119–155
  46. Schwartz WR, Kembhavi A, Harwood D, Davis LS (2009) Human detection using partial least squares analysis. In: *International Conference on Computer Vision*, pp 24–31
  47. Seo H, Milanfar P (2010) Visual saliency for automatic target detection, boundary detection, and image quality assessment. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp 5578–5581
  48. Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R, LeCun Y (2013) Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:13126229
  49. Stilla U, Michaelsen E, Soergel U, Hinz S, Ender H (2004) Airborne monitoring of vehicle activity in urban areas. *International Archives of Photogrammetry and Remote Sensing* 35:973–979
  50. Tan X, Triggs B (2007) Enhanced local texture feature sets for face recognition under difficult lighting conditions. In: *Analysis and Modeling of Faces and Gestures*, pp 168–182
  51. Tanner F, Colder B, Pullen C, Heagy D, Eppolito M, Carlan V, Oertel C, Sallee P (2009) Overhead imagery research data set: an annotated data library and tools to aid in the development of computer vision algorithms. In: *Proceedings of IEEE*

- Applied Imagery Pattern Recognition Workshop, pp 1–8
52. Tenenbaum JB, de Silva V, Langford JC (2000) A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science* 290:2319–2323
  53. Torralba A (2003) Contextual priming for object detection. *International Journal of Computer Vision* 53:169–191
  54. Tuzel O, Porikli F, Meer P (2006) Region covariance: A fast descriptor for detection and classification. In: *European Conference on Computer Vision*, pp 589–600
  55. Vedaldi A, Zisserman A (2012) Sparse kernel approximations for efficient classification and detection. In: *IEEE Conference on Computer Vision and Pattern Recognition*
  56. Viola P, Jones MJ, Snow D (2005) Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision* 63:153–161
  57. Walk S, Majer N, Schindler K, Schiele B (2010) New features and insights for pedestrian detection. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp 1030–1037
  58. Wang X (2012) A discriminative deep model for pedestrian detection with occlusion handling. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp 3258–3265
  59. Wang X, Han T, Yan S (2009) An hog-lbp human detector with partial occlusion handling. In: *International Conference on Computer Vision*, pp 32–39
  60. Weinberger KQ, Saul LK (2006) Unsupervised learning of image manifolds by semidefinite programming. *International Journal of Computer Vision* 70:77–90
  61. Weinberger KQ, Sha F, Saul LK (2004) Learning a kernel matrix for nonlinear dimensionality reduction. In: *International Conference on Machine Learning*, p 106
  62. Wong T (2007) Atr applications in military missions. In: *IEEE Symposium on Computational Intelligence in Security and Defense Applications*, pp 30–32
  63. Wu B, Nevatia R (2005) Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In: *International Conference on Computer Vision*, pp 90–97
  64. Zhang X, Gao X, Caelli T (2012) Parametric manifold of an object under different viewing directions. In: *ECCV*, pp 186–199
  65. Zhang Z, Zha H (2004) Principal manifolds and nonlinear dimension reduction via local tangent space alignment. *Society for Industrial and Applied Mathematics Journal on Scientific Computing* 26:313–338
  66. Zhao T, Nevatia R (2003) Car detection in low resolution aerial images. *Image and Vision Computing* 21:693–703
  67. Zheng H, Li L (2007) An artificial immune approach for vehicle detection from high resolution space imagery. *International Journal of Computer Science and Network Security* 7:67–72