Discriminative Autoencoders for Small Targets Detection

Sebastien Razakarivony, Frédéric Jurie

Discriminative Autoencoders for Small Targets Detection.

Sebastien Razakarivony
SAGEM D.S. – SAFRAN Group
CNRS UMR 6072 – University of Caen – ENSICAEN
Email: sebastien.razakarivony@sagem.com

Frédéric Jurie
CNRS UMR 6072 – University of Caen – ENSICAEN
Email: frederic.jurie@unicaen.fr

***Abstract*—This paper introduces the new concept of *discriminative autoencoders*. In contrast with the standard autoencoders – which are artificial neural networks used to learn compressed representation for a set of data – *discriminative autoencoders* aim at learning low-dimensional discriminant encodings using two classes of data (denoted such as the positive and the negative classes). More precisely, the discriminative autoencoders build a latent space (manifold) under the constraint that the positive data should be better reconstructed than the negative data. It can therefore be seen as a generative model of the discriminative data and hence can be used favorably in classification tasks. This new representation is validated on a target detection task, on which the discriminative autoencoders not only give better results than the standard autoencoders but are also competitive when compared to standard classifiers such as the Support Vector Machine.**

I. Introduction

This paper addresses the problem of the detection of small targets in aerial images, a task commonly referred as *Automatic Target Recognition* (ATR). Typical images illustrating this task can be seen from Fig. 1. This is an old – yet unsolved – computer vision task. This task is complex and challenging not only because of the smallness of objects but also because of illumination and color changes, pose differences or occlusions. However, this task is important and involved in many applications such as visual surveillance, safety or protection.

Despite the fact that object detection has received a lot of attention during the last 5 years (*e.g.* [1], [2]), our problem differs from these recent works for several reasons. First, representing small targets is difficult: indeed, the afore mentioned recent approaches rely on the use of accurate descriptors capturing fine discriminating details of the objects (*e.g.* the Dalal and Triggs's HOG descriptor for pedestrian detection [3]) and/or on the use of part-based models [4]. These descriptors and models can hardly be used for as small as *e.g.* 20 pixel-wide objects. Secondly, it is usually difficult to obtain large enough training sets as it is expensive to collect and annotate aerial images. Thirdly, image backgrounds (*i.e.* the pixels surrounding the targets) are usually not correlated with the objects themselves: vehicles can be on a roads, fields *etc.* Finally, in some cases, vehicles can even be camouflaged and cannot be distinguished easily from the background.

Interestingly, while target's appearance usually belongs to a high dimensional space – *e.g.* a 20×20 pixel targets lies in a 400-d space – only a few parameters (often called the latent variables) are necessary to explain their appearances.

Fig. 1. Typical images for automatic target recognition, from the VeDAI Dataset.

Such parameters can be, for example, the 3D pose or the illumination. Furthermore, it has been shown in the past that manifolds are good candidates to represent small size objects for which distinct features can hardly be extracted ([5], [6], [7]). Indeed, they allow to represent the high-dimensional manifolds containing the data by low-dimensional representational space. Supporting this assumption, the work of Zhang [6] shows that images of 3D objects seen from different view-points can be represented as points on a low-dimensional manifold. Different works used manifold learning as a generative model, such as the famous work of Pentland, based on linear manifolds obtained by Principal Component Analysis [5], or the work of Feraud *et al.* [7] based on non-linear manifold learning.

However, one strong limitation of these manifold-based approaches is that, while they accurately model object's appearance, they do not focus on the discriminative information, contrarily to state-of-the-art approaches using boosting [8], Support Vector Machines (SVM) [9] or neural networks (NN) [10]. Once the manifold is learned, hence constituting a generative model of the data, the discriminant information is irremediably lost. This is what motivates this paper. Indeed,

this paper proposes a new type of autoencoders denoted as *discriminative autoencoders* which uses an autoencoder to build a generative model of the discriminative information. In contrast with the standard autoencoders, our *discriminative autoencoders* learn a manifold which is by construction good at reconstructing the targets while the backgrounds are poorly reconstructed. It opens the door to very simple classification frameworks in which the reconstruction error can be used as a natural, simple and efficient way to classify image windows in a sliding window framework.

Experiments on a dataset including vehicles in aerial images (the VeDAI dataset) show that the proposed approach is not only better than the standard autoencoders but also performs better than discriminative classifiers such as the SVM classifiers.

The rest of the paper is as follows: we first introduce the related works in Section II, we present the discriminative autoencoders in Section III, and, finally, give the experimental validation of the proposed approach in Section IV.

II. Related Works

Even if target detection has a long history in the computer vision literature, the recent literature on object detection focuses mainly on the detection of large objects in consumer images. Most of the techniques are based on the *sliding windows* framework, combining descriptors, such as Histogram of Oriented Gradient [3] or Haar Wavelets [11], with powerful discriminative classifier, such as boosting [8] or SVM [9]. Many improvements have been proposed to enhance these techniques *e.g.* new kernels [12] or part-based models [4].

More directly related to our problem, some approaches have been specifically designed for the detection of vehicles. In [13], Zhao and Nevatia pose car detection as a 3D object recognition problem, to account for the variation in viewpoint and shadow. Their experiments show promising results on challenging images, but the cars that are not on roads do not seem to be well detected. Eikvil *et al.* [14] use several different features combined with Linear Discriminant Analysis [15]. A segmentation step, followed by a two-stages classifier is used. Their work relies on the availability of multispectral and panchromatic images, and on the knowledge of the road network. Despite the authors' promising results, the vehicles can be detected only because they are supposed to be on roads. In [16], Stilla *et al.* propose several 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 the context as well. [17] 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 [3], obtaining similar performance. Other works address the detection of small vehicles such as [14], [18]. However all of them assume that the vehicles are located on roads, to make the detection easier, and cannot be used in our context. Finally, it worths noting that none of their experiments can be reproduced because neither the protocols nor the datasets have been made publicly available.

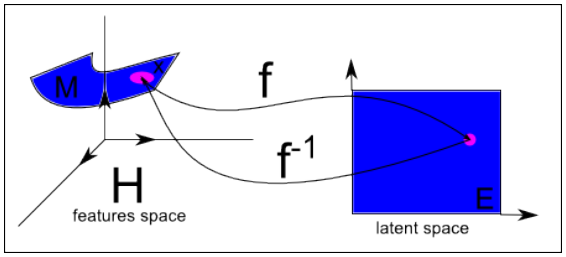

Fig. 2. Manifolds: vectors of the original Euclidean space H belong to a manifold spanned by the subspace E. f is a mapping function, giving the correspondence between a point in the manifold $\mathcal{M}$ in the original space and its projection in the subspace. f^{-1} is its inverse, going from E to $\mathcal{M}$

There are very few papers specifically addressing the detection of small objects. These papers are often based on the detection of *salient regions*. In this case, the objects to be detected are defined as the regions of the image which do not have the same statistics as the background *e.g.* [19], [20]. Among the rare papers which tried to model small targets explicitly, we can mention the work of [21], which – in addition to introducing a new dataset of 36×18 pixels pedestrian images – has shown that good performance can be obtained by combining standard features such as Haar wavelets or HOG features with SVM/boosting classifiers [22].

Manifolds have been successfully used to model object's appearance. A manifold is a subspace embedded in a higher dimension space which can be locally approximated by an Euclidean subspace, denoted as the *latent space*. The geodesic distances, which are the shortest paths between two points inside a manifold, are locally preserved in the latent space. Fig. 2 gives an illustration by showing the relationship between the manifold (E) and the Euclidean space (H). A manifold can be learned through different ways. The simplest methods are the linear ones *e.g.* Linear Discriminant Analysis [15], or the simple Principal Components Analysis. Regarding non linear methods, some of them are based on the conservation of geodesic distances such as Isomap [23]. Local Linear Embedding [24] and its variants learn linear local approximations of the manifold. Other approaches learn the manifold in a global way, such as the Maximum Variant Unfolding [25], or autoencoders [26]. It worths pointing out that most of these algorithms have been designed to visualize high-dimensional data in 2D, and thus only give the mapping f (see figure 2), but 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}.

Manifold learning has already been used by several authors to address detection tasks. In [5], Pentland introduced the well known *eigenfaces*, using Principal Component Analysis to build linear face manifolds used for face detection. It has also been applied later to hand detection in [27]. In the same spirit, [28] uses PCA for object detection, by modeling the background and the objects as linear manifolds. Interesting results are reported on good quality car and pedestrian images,

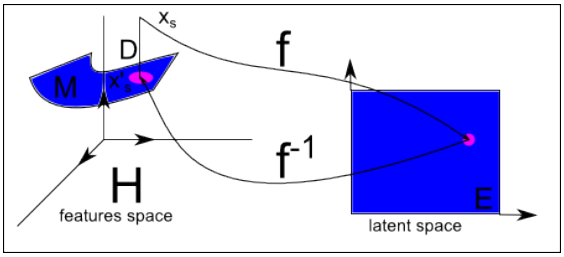

Fig. 3. Illustration of the concept of *distance to the manifold*. Let X_s be a vector and $X'_s = f^{-1} \circ f(X_s)$ is its projection on the latent space. $||X'_s - X_s||$ is referred as the distance to the manifold. f goes from $\mathcal{H}$ to E, and f^{-1} goes from E to $\mathcal{M}$ – which is included in $\mathcal{H}$.

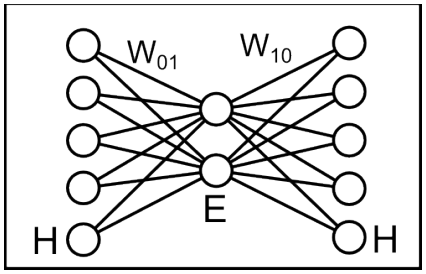

Fig. 4. Let H be the input space and E the latent space. It is an illustration of the minimal autoencoder, which is made of 3 layers.

for high dimensional manifolds. In [7], the authors use autoencoders to build face manifolds for face detection, but the false alarm rate is high, probably because the background model can not be learned with such a model. In [29], the authors counter this effect by using a distinct background model.

Our approach builds on these recent works by using one of the best current image representations (HOG) combined with a manifold learning approach. The contribution of the paper lies in the new framework allowing to learn *discriminative autoencoders*. As far as we know, this is the first time such a model is proposed.

III. Discriminative Autoencoder

Before presenting the proposed discriminative autoencoders, we start by explaining how manifolds can be used as classifiers, and how to learn manifolds with autoencoders.

A. Manifold as a generative model

Let H denote the input space and $\mathbf{x} \in \mathcal{H}$ a *visual signature* (also called *visual feature*) extracted from an image. We remind that building a Riemannian manifold $\mathcal{M}$ representative of the visual signatures is equivalent to finding a function f, such as:

$$\forall \mathbf{x} \in \mathcal{M}, \exists ! \bar{\mathbf{x}} \in \mathcal{R}^n, \bar{\mathbf{x}} = f(\mathbf{x}) \tag{1}$$

f is called the embedding of $\mathcal{M}$, and is an isometric function.

Obviously, if $\mathbf{x}$ lies on the manifold, $f^{-1} \circ f(\mathbf{x}) = \mathbf{x}$. $f^{-1} \circ f$ 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:

$$D_{\mathcal{M}}(\mathbf{x}) = ||\mathbf{x} - P_{\mathcal{M}}(\mathbf{x})|| \tag{2}$$

where $||x||$ represent the Euclidean norm of $\mathbf{x}$. The principle of this projection is illustrated by Fig. 3. This distance can then be used to model a category, as the closer to the manifold a vector is, the more likely it belongs to the category.

B. Autoencoder

Autoencoders are symmetrical neural networks, which learn the identity function under constraints. A typical simple autoencoder is presented Fig. 4, but more complex architectures can be used. 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 connecting 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 Fig. 4. As the network is symmetric, $dimension(W_{ji}) = dimension(W_{ij}^T)$. Each layer j has an output $\mathbf{r}(\mathbf{x})$, fully defined by the layer input $\mathbf{x}$ and the weights matrix:

$$\mathbf{r}(\mathbf{x}) = h(W_{ij}\mathbf{x}) \tag{3}$$

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 [26]. Contrary to this, using non-linear h functions allow the network to approximate any function [30].

Let us denote χ the set of training vectors $\mathbf{x}$. The standard autoencoder minimizes the following loss function [26]:

$$L(\chi) = \sum_{\mathbf{x} \in \chi} ||\mathbf{x} - \tilde{\mathbf{x}}||^2, \tag{4}$$

hence minimizing the reconstruction error. $\tilde{\mathbf{x}}$ is the reconstruction of $\mathbf{x}$ given by the autoencoder. The loss is usually minimized using a stochastic gradient descend, within a *back-propagation* framework [31]. f and its inverse are therefore learned simultaneously. The latent space E is available as the output of the middle layer. More efficient convergence rates can be achieved using Restricted Boltzmann Machine [32] and Contrastive Divergence [33]. The interested reader can see [34] for further details.

In the context of manifold learning, the network is usually used to learn f and f only, providing an embedding of data [34]. In contrast, we learn the full network, which gives the projection on the manifold $P_{\mathcal{M}}(x)$ we are looking for. The distance from the class (which can be used as a classification score) can be computed as simply as in Eq. (2).

C. Discriminative autoencoder

In contrast with standard autoencoders, we introduce the concept of *discriminative autoencoders*, which use data from two classes (denoted in the following as χ^+, the set of positive training vectors, and χ^- the set of negative ones) and learn a manifold which is good at reconstructing the data of the positive class while ensuring that those of the negative class are pushed away from the manifold. By doing this, we intend to take advantage of the information carried by negative examples. Let us denotes as $t(\mathbf{x})$ the label of the example $\mathbf{x}$,

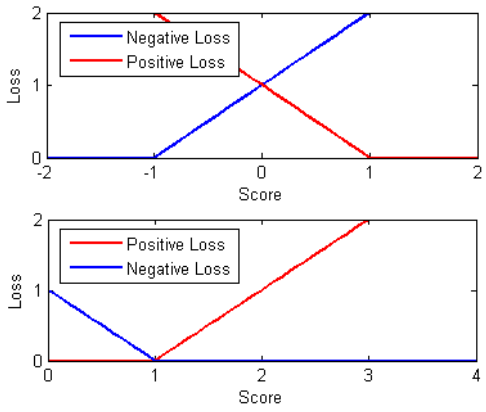

Fig. 5. Hinge Loss. Top: graph of the standard hinge loss. Bottom: the hinge loss used for metric learning.

with $t(\mathbf{x}) \in \{-1, 1\}$ and $e(\mathbf{x})$ the distance of that example to the manifold with $e(x) = ||\mathbf{x} - \tilde{\mathbf{x}}||$. We substitute the loss function given by Eq. 4 by the following one:

$$L_d(\chi^+ \cup \chi^-) = \sum_{\mathbf{x} \in \chi^+ \cup \chi^-} \max\left(0, t(\mathbf{x}) \cdot (e(\mathbf{x}) - 1)\right) \quad (5)$$

which is nothing else than the hinge loss function (see Fig. 5 for an illustration), used in many different classification algorithms such as the SVM. In practice, we use the slightly different version of the standard hinge loss, as proposed by [35] – the standard one should be $L_d = \sum_{\mathbf{x} \in \chi^+ \cup \chi^-} \max(0, 1 - t(\mathbf{x}) \cdot e(\mathbf{x}))$ – more adapted to our problem as the reconstruction errors are all positive. In that sense, our problem is closer to a metric learning task than to a classification task. When the minimum is reached, positive (resp. negative) examples are expected to have a reconstruction error lower (resp. greater) than 1.

To optimize the loss function, we use here again a back-propagation of the error. First we give the equations of the autoencoder, which are:

$$\mathbf{y} = h(W_{10}\mathbf{z}) \quad \text{and} \quad \mathbf{z} = k(W_{01}\mathbf{x}) \quad (6)$$

As done in [34], the activation function k is the identity function, and h is the sigmoid function. To simplify these equations, let us denote $\mathbf{u}$ and $\mathbf{v}$ as:

$$\mathbf{u} = W_{10}\mathbf{z} \quad \text{and} \quad \mathbf{v} = W_{01}\mathbf{x} \quad (7)$$

The objective is to estimate the coefficients w_{ki} of W_{01} and W_{10} by minimizing $L(\chi^+ \cup \chi)$. The optimum values of w_{ki} verify:

$$\frac{\partial L}{\partial w_{ki}} = 0 \quad (8)$$

which can be solved using a stochastic gradient descend. The partial derivatives can be written as:

$$\frac{\partial L}{\partial w_{ki}} = \frac{\partial L}{\partial \mathbf{e}^i} \cdot \frac{\partial \mathbf{e}^i}{\partial \mathbf{y}^i} \cdot \frac{\partial \mathbf{y}^i}{\partial \mathbf{u}^i} \cdot \frac{\partial \mathbf{u}^i}{\partial w_{ki}} \quad (9)$$

with:

$$\frac{\partial \mathbf{e}^i}{\partial \mathbf{y}^i} = -1; \quad \frac{\partial \mathbf{y}^i}{\partial \mathbf{u}^i} = \frac{\partial h(\mathbf{u}^i)}{\partial \mathbf{u}^i}; \quad \frac{\partial \mathbf{u}^i}{\partial w_{ki}} = z^k \quad (10)$$

Furthermore, in the case of the sigmoid function:

$$\frac{\partial h(\mathbf{u}^i)}{\partial \mathbf{u}^i} = \mathbf{y}^i \cdot (1 - \mathbf{y}^i) \quad (11)$$

Up to here, the derivations are close to the classic back-propagation. Then, by introducing the hinge loss:

$$\frac{\partial L}{\partial \mathbf{e}^i} = \begin{cases} \mathbf{e}^i & \text{if } t(\mathbf{x}) \cdot (e(\mathbf{x}) - 1) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

we thus obtain the following gradient step:

$$\Delta w_{ki} = -\eta \delta_i \mathbf{z}^k \quad (13)$$

with $\delta_i = \mathbf{e}^i \frac{\partial h(\mathbf{u}^i)}{\partial \mathbf{u}^i}$ if $t(\mathbf{x}) \cdot (e(\mathbf{x}) - 1) > 0$ and 0 otherwise.

For the hidden layer (we give the derivations for only one hidden layer, but this is the same with more hidden layers):

$$\frac{\partial L}{\partial w_{lk}} = \frac{\partial L}{\partial \mathbf{z}^k} \cdot \frac{\partial \mathbf{z}^k}{\partial \mathbf{v}^l} \cdot \frac{\partial \mathbf{v}^l}{\partial w_{lk}} \quad (14)$$

The two last terms do not change, but the first one becomes:

$$\frac{\partial L}{\partial \mathbf{z}^k} = \sum_n e^n \frac{\partial e^n}{\partial \mathbf{z}^k} \text{ if } t(\mathbf{x}) \cdot (e(\mathbf{x}) - 1) > 0 \quad (15)$$

which give us:

$$\frac{\partial L}{\partial \mathbf{z}^k} = \sum_n e^n \frac{\partial (\mathbf{x}^k - h(\mathbf{u}^n))}{\partial \mathbf{u}^n} \cdot \frac{\partial \mathbf{u}^n}{\partial \mathbf{z}^k} \quad (16)$$

$$= -\sum_n e^n \frac{\partial (h(\mathbf{u}^n))}{\partial \mathbf{u}^n} w_{kn} = -\sum_n \delta(n) w_{kn} \quad (17)$$

The increment is therefore:

$$\Delta w_{lk} = -\eta \delta_k \mathbf{x}^l \quad (18)$$

with $\delta_k = \frac{\partial h(\mathbf{v}^k)}{\partial \mathbf{v}^k} \sum_n \delta(n) w_{kn}$ if $t(\mathbf{x}) \cdot (e(\mathbf{x}) - 1) > 0$ and 0 otherwise. For both increment, η is the learning rate. Different ways exist to optimize it. As it is not the main subject, the reader should see [36] for more information.

These equations can be made more robust by adding a margin w to the equation, $t(\mathbf{x}) \cdot (e(\mathbf{x}) - 1) > 0$, thus becoming $t(\mathbf{x}) \cdot (e(\mathbf{x}) - 1) + w > 0$. w is chosen by cross validation. Finally, the reconstruction error for a new vector x – which can be used as a classification score – is:

$$e(\mathbf{x}) = ||\mathbf{x} - \tilde{\mathbf{x}}|| \quad (19)$$

IV. Experiments

This section experimentally validates the proposed discriminative autoencoder on the task of small targets detection. We first introduce the dataset used in our experiments, the VeDAI (Vehicle Detection in Aerial Imagery) dataset, and then present our detection pipeline. We finally report the performance of the discriminative autoencoders and compare it with the performances of standard autoencoders and standard discriminative classifiers. We also compare the proposed detector to the Deformable Part Model of [4].

TABLE I. RESULTS ON THE VEDAI DATASET

Detector	mAP	Recall @ 0.01 FPPI	Recall @ 0.1 FPPI	Recall @ 1 FPPI
Deformable Part Model [4]	60.5±4.2	13.4±6.8	31.4±5.8	74.5±4.5
HOG-SVM (1st stage only)	58.9±3.5	13.2±5.1	30.4±3.9	72.1±4.1
HOG-SVM followed by Standard Autoencoder	30.0± 3.9	1.5±1.6	6.8 ±1.8	39.5± 4.1
HOG-SVM combined with Standard Autoencoder	58.8±3.8	12.9±3.5	34.0 ±4.5	71.8±5.4
HOG-SVM followed by Discriminative Autoencoder	68.0±4.2	**21.2±6.9**	46.7 ±6.8	78.7±3.4
HOG-SVM combined with Discriminative Autoencoder	**69.6±3.4**	20.4 ±6.2	**49.0±3.6**	**80.3±3.1**

Fig. 6. 100x100 pixels regions centred on cars, extracted from the VeDAI dataset. Small size, specular reflections, shadows or occlusion make the detection challenging.

A. VeDAI dataset

We did the experimental validation on a dataset built to benchmark small targets detection algorithms, the VeDAI dataset[1]. It contains a total of 1,210 images (of 1024×1024 pixels, 3 color bands) with various backgrounds and vehicles (see illustrative images Fig. 1). These images come from the Utah ARGC website [37], and more precisely from the 2012 HRO 6 inch orthophotography set. The image resolution is 12.5×12.5cm per pixel. The cars have a size around 20×40 pixels. Their detection is challenging because of occlusions, specular reflections and shadows, as shown by Fig. 6. The intra-class variation is important. We used a 10-fold cross validation process: the 1,210 images are split in 10 folds, each of them containing 134 targets (cars) in 121 different images. During the evaluation, 9 folds are used for training and the last for testing. Each fold is used in turn as the test set.

B. Detection pipeline

Our detection pipeline builds on the standard *sliding window* framework using *manifold learning* to score the windows. All the possible rectangular regions of a given aspect ratio are evaluated one by one by our object classifier. This is done in practice by using a multi-scale grid. We used a typical step-size of 8 pixels and a ratio of $2^{\frac{1}{10}}$ between each scales, such as done by [4]. As the aspect ratio of vehicles can vary a lot depending on the orientation, several distinct classifiers are trained. The aspect ratio clusters are obtained by clustering training image regions. Only four scales were used as the distance to target is the same from one image to another. To improve the efficiency, we adopt a two stage cascade. The first stage is made of 12 linear-SVM classifiers based on HOG features (2 different orientations × 6 different aspect ratios), while the second stage re-scores the detections using 12 discriminative autoencoders (one paired with each SVM classifier). First-stage detectors are trained using initial training data while those of the second stage are trained from the *hard negative* (*i.e.* the false positive of the first stage with highest scores) of the training images. The number of neurons in the autoencoders (with only one hidden layer) is set by cross validation. During testing, the 12 SVM detectors are run over the entire image and only the windows with a score over -1.0 are kept. Then, as usually with sliding-windows, a non-maximum filtering stage is applied. We use a simple and efficient iterative greedy strategy consisting in keeping only the windows which have the maximum score over a disk (which radius is half the window width). We set the windows so that the selected windows do not overlap by more than 50 percent. Finally, the selected windows are re-scored with a standard or discriminative autoencoder. Our algorithm can virtually use any type of image features as input. In practice, all the presented experiments are done with HOG.

We also tried to combine the two scores (*i.e.* the ones of the first-stage SVM classifiers with the ones of the discriminative autoencoders). In this case the final score is computed as:

$$\alpha S_{autoencoder}(\mathbf{x}) + (1-\alpha) S_{SVM}(\mathbf{x}) \qquad (20)$$

α being fixed by cross validation.

C. Results

We measure the performance of the target detector by the mean Average Precision (mAP) over the 10 folds, as well as by the mean detection rate (Recall) at 0.01 false positive per image (FPPI), 0.1 FPPI and 1 FPPI. The mean average precision is computed from an 11 points extrapolation of the precision-recall curve, as done in many detection benchmarks [1]. The mean and the standard deviation are computed over the 10 folds. The results are given in Table I. We first observe that the standard autoencoder (using only positive examples) does not give good results on its own. Even when combined with the first-stage SVM, it does not significantly improve the performance of the SVM alone. On the other hand, the discriminative autoencoder performs much better than the standard autoencoder (+38.0 of mAP), but also significantly better than the standard SVM (+9.1 of mAP), even if it uses exactly the same training examples. For the different operating points, the discriminative autoencoder gains +16.3 of recall at 0.1 FPPI, and +6.6 of recall at 1 FPPI compared to the SVM alone. The large standard deviation of 0.01 FPPI makes conclusions not reliable, but the observed gain is of +8.0% on the detection rate. The combination of the score of the discriminative autoencoder and the one of the SVM gives slightly better results than the discriminative autoencoder alone, with a gain of +1.6 of mAP, +1.6 of recall at 1 FPPI and of +2.3 of recall at 0.1 FPPI. Finally, when compared to the Deformable Part Model of [4] (we used the latest release

[1]can be downloaded from https://jurie.users.greyc.fr/

of the author's code), one of the state-of-the-art detector at the moment, the gain is of about 10% of mAP. These results clearly show that not only discriminative autoencoders largely outperform standard autoencoders but also that it significantly outperforms linear SVM or DPM detectors, on this small objects detection task.

V. Conclusions

This paper introduces the new concept of *discriminative autoencoders*, which, in addition to optimizing the reconstruction of the positive examples (as standard autoencoders do), push the manifold away from the negative examples. We also show how such autoencoders can be trained, inspired by recent metric learning techniques. In the context of small target detection, where manifold learning is very relevant, we have shown that the discriminative autoencoders perform much better that the traditional autoencoders, and offer a significant gap over other approaches (including the Deformable Part Model) when it is associated with a linear SVM classifier.

Acknowledgment

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

References

[1] M. Everingham, L. V. Gool, Williams, C. K. I., J. Winn, and A. Zisserman, "The pascal voc challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, 2010.

[2] P. Carbonetto, G. Dorkó, C. Schmid, H. Kück, and N. De Freitas, "Learning to recognize objects with little supervision," *International Journal of Computer Vision*, vol. 77, pp. 219–237, 2008.

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886–893.

[4] P. Felzenszwalb, R. Girshick, D. Mcallester, and D. Ramanan, "Object detection with discriminatively trained part based models," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, 2009, pp. 1627–1645.

[5] A. Pentland, "Viewbased and modular eigenspaces for face recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1994.

[6] X. Zhang, X. Gao, and T. Caelli, "Parametric manifold of an object under different viewing directions," in *ECCV*, 2012, pp. 186–199.

[7] R. Feraud, O. Bernier, J. Viallet, and M. Collobert, "A fast and accurate face detector based on neural networks," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, 2001, pp. 42–53.

[8] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *International Journal of Computer Vision*, vol. 63, pp. 153–161, 2005.

[9] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, pp. 2278–2324, 1998.

[11] S. G. Mallat, "A theory for multiresolution signal decomposition - the wavelet representation," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, 1989, pp. 674–693.

[12] A. Vedaldi and A. Zisserman, "Sparse kernel approximations for efficient classification and detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[13] T. Zhao and R. Nevatia, "Car detection in low resolution aerial images," *Image and Vision Computing*, vol. 21, pp. 693–703, 2003.

[14] L. Eikvil, L. Aurdal, and H. Koren, "Classification-based vehicle detection in high-resolution satellite images," *Journal of International Society for Photogrammetry and Remote Sensing*, vol. 64, pp. 65–72, 2009.

[15] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, pp. 179–188, 1936.

[16] U. Stilla, E. Michaelsen, U. Soergel, S. Hinz, and H. Ender, "Airborne monitoring of vehicle activity in urban areas," *International Archives of Photogrammetry and Remote Sensing*, vol. 35, pp. 973–979, 2004.

[17] A. Kembhavi, D. Harwood, and L. S. Davis, "Vehicle detection using partial least squares," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, 2011, pp. 1250–1265.

[18] H. Zheng and L. Li, "An artificial immune approach for vehicle detection from high resolution space imagery," *International Journal of Computer Science and Network Security*, vol. 7, pp. 67–72, 2007.

[19] U. Rutishauser, D. Walther, C. Koch, and P. Perona, "Is bottom-up attention useful for object recognition?" in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2004, pp. II–37.

[20] H. Seo and P. Milanfar, "Visual saliency for automatic target detection, boundary detection, and image quality assessment," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2010, pp. 5578–5581.

[21] S. Munder, "An experiment study on pedestrian classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, 2006.

[22] M. Enzweiler and D. Gavrila, "Monocular pedestrian detection: Survey and experiments," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, 2009, pp. 2179–2195.

[23] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science*, vol. 290, pp. 2319–2323, 2000.

[24] L. Saul and S. Roweis, "Think globally, fit locally: unsupervised learning of low dimensional manifolds," *Journal of Machine Learning Research*, vol. 4, pp. 119–155, 2003.

[25] K. Q. Weinberger and L. K. Saul, "Unsupervised learning of image manifolds by semidefinite programming," *International Journal of Computer Vision*, vol. 70, pp. 77–90, 2006.

[26] M. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *American Institute of Chemical Engineers Journal*, vol. 37, pp. 233–243, 1991.

[27] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, 1997, pp. 696–710.

[28] G. V. Carvalho, L. B. Moraes, G. D. Cavalcanti, and T. I. Ren, in *IJCNN*.

[29] S. Razakarivony and F. Jurie, "Small target detection combining foreground and background manifolds," in *IAPR International Conference on Machine Vision and Application*, 2013.

[30] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, pp. 303–314, 1989.

[31] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," DTIC Document, Tech. Rep., 1985.

[32] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for boltzmann machines," *Cognitive Science*, vol. 9, pp. 147–169, 1985.

[33] G. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, p. 2002, 2000.

[34] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, pp. 504–507, 2006.

[35] A. Mignon and F. Jurie, "Pcca: A new approach for distance learning from sparse pairwise constraints," in *CVPR*. IEEE, 2012, pp. 2666–2672.

[36] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 1998, pp. 9–50.

[37] website, "Utah agrc website," 2012. [Online]. Available: \url{http://gis.utah.gov/}