Autoencodeurs discriminants pour la détection de cibles faiblement résolues

Sébastien Razakarivony, Frédéric Jurie

Autoencodeurs discriminants pour la détection de cibles faiblement résolues

Sébastien Razakarivony[1,2] Frédéric Jurie[2]
[1] SAGEM D.S. – SAFRAN Group
[2] CNRS UMR 6072 – Université de Caen – ENSICAEN

Résumé

Les autoencodeurs permettent de modéliser des données au moyen de variétés. Dans un contexte de détection d'objets, ils modélisent l'apparence des objets à détecter. La distance entre un vecteur à classer et la variété peut alors être utilisée comme une mesure de probabilité d'appartenance du vecteur à la classe. Cependant, si la variété ainsi construite est telle que les vecteurs de la classe appartiennent à la variété, rien ne garantit que des vecteurs d'autres classes ne lui appartiennent pas également. Nous cherchons à lever cette limitation en proposant un nouveau type d'autoencodeurs, les autoencodeurs discriminants, qui ont la propriété de construire des variétés éloignant les formes négatives des positives. Une validation expérimentale dans un contexte de détection d'objets permet de conclure sur la pertinence de la méthode proposée.

Mots Clef

Vision par Ordinateur, Détection d'Objets, Variétés.

Abstract

Autoencoders, which model data by manifolds, can be used in the context of object detection to model object's appearances. However, if such manifolds can represent well the data of a given class, there is no guaranty that data from other classes cannot also be on the manifold. This paper addresses this limitation by proposing a new type of autoencoders called discriminative autoencoders, allowing to build manifold for which negative data are not on the manifold. This approach is validated on a task of small targets detection, task for which promising results are obtained.

1 Introduction

Cet article adresse la question de la détection de petits objets dans des images aériennes (illustration Figure 1). Il s'agit d'une tâche ancienne, non résolue à ce jour, dont la difficulté réside dans la petitesse des objets à détecter mais également dans les multiples changements d'apparence possibles des véhicules (variations de couleurs, d'illumination, présence d'occlusions, d'ombres, *etc.*). Bien que difficile, cette tâche est importante pour de nombreuses applications en lien avec la surveillance ou la sécurité.
Détecter des petites cibles ne peut se faire directement en utilisant les méthodes standards de la littérature (*e.g.* approche de Dalal et Triggs pour la détection de piétons [5]

FIGURE 1 – Images représentatives d'un problème de détection en imagerie aérienne, extraites de la base VeDAI.

ou l'utilisation de modèles à parties [8]), ces méthodes n'étant pas adaptées à la détection d'objets d'une vingtaine de pixels de large. D'autre part, il est assez difficile d'obtenir des données d'apprentissage pour cette tâche, en raison du coût de collecte des images aériennes. Finalement, la tâche est également rendue difficile par le fait que les cibles ne sont généralement pas corrélées avec leur voisinage immédiat dans l'image : un véhicule peut se trouver sur une route, un parking, dans une forêt, *etc.* Dans certains cas, les véhicules peuvent même être partiellement camouflés et en conséquence encore plus difficiles à détecter.

Une imagette 20x20 pixels peut être vue comme un vecteur dans un espace à 400 dimensions, dont les composantes peuvent s'expliquer en réalité par un petit nombre de paramètres (souvent appelés variables latentes), tels que la pose 3D ou l'illumination. Il a été montré que la théorie des variétés est un bon cadre formel pour représenter les objets de petite taille, pour lesquels des descripteurs standards peuvent difficilement être extraits ([9, 22, 32]). Cette théorie permet de représenter la variété de grande dimension par un sous espace de dimension inférieure. Par exemple, le travail de Zhang [32] montre que les images d'objets 3D vus depuis différents points de vue peuvent

être représentées par des points sur une variété de dimension réduite. Différents travaux utilisent les variétés comme modèle génératif, tel que le travail de Pentland [22], basé sur des variétés linéaires obtenues par Analyse en Composantes Principales, ou le travail de Feraud [9], basé sur de l'apprentissage de variétés non-linéaires.

Cependant, bien qu'elles modélisent fidèlement les apparences des objets, les approches basées sur les variétés ne se concentrent pas sur les informations discriminantes, contrairement aux approches de l'état de l'art, comme le boosting [30], les Séparateurs à Vastes Marges (SVM) [3], ou encore certains réseaux de neurones (RN) [18]. Cet article vise à lever cette limitation en proposant un nouveau type d'autoencodeurs, que nous désignons comme des *autoencodeurs discriminants*, qui permettent la construction d'un modèle génératif de l'information discriminante. Contrairement aux autoencodeurs standards, notre autoencodeur discriminant apprend une variété qui, par construction, permet de bien représenter les apparences des cibles à détecter tout en assurant que les apparences des fonds seront mal représentées. Une fois cette variété construite, une classification simple se basant sur l'erreur de reconstruction peut être utilisée.

Nos expériences sur une base de données présentant des véhicules dans des images aériennes (la base VeDAI, décrite section 4), montre que l'approche proposée est non seulement meilleure que les autoencodeurs standards, mais qu'elle permet d'améliorer significativement les résultats d'un classifieur discriminant classique tel que le SVM.

Le reste de l'article est construit comme suit : tout d'abord nous présentons dans la section 2 les études en relation avec le travail proposé, puis nous présentons les autoencodeurs discriminants dans la section 3. Enfin, les validations expérimentales sont présentées dans la section 4.

2 Etat de l'art

Même si la détection d'objets dans les images a une longue histoire dans la littérature de la vision par ordinateur, les travaux récents se concentrent principalement sur la détection de grands objets dans des images issues de la vie quotidienne. Ils sont pour la plupart basés sur l'utilisation de *fenêtres glissantes*, combinant des descripteurs tels que les Histogrammes de Gradient Orientés (HOG) [5] ou les Local Binary Pattern [31], avec de puissants classifieurs discriminants, tel que les techniques de boosting [30] ou le SVM [3]. Beaucoup d'améliorations ont été proposées, par exemple le développement de stratégies d'élagage [17] ou les modèles déformables à base de parties [8].

Plus proche de notre problème, certaines approches ont été développées spécifiquement pour la détection de véhicules. Dans [33], Zhao et Nevatia envisagent la détection de voitures comme un problème de détection d'objets 3D, prenant en compte les variations de points de vue et d'ombre. Dans [28], Stilla *et al.* proposent différents algorithmes adaptés aux différents capteurs utilisés (couleur, imagerie infrarouge, radar). De son côté, [15] montre des résultats intéressants reposant sur un ensemble de descripteurs développés spécifiquement pour la détection de véhicules, basés sur leurs couleurs et leurs attributs géométriques. Les résultats obtenus par les différentes approches mentionnées dans ce paragraphe ne peuvent être comparées, chacune utilisant des bases d'images et des métriques différentes. En outre, les différentes bases utilisées ne sont pas publiques.

Peu de travaux portent sur la détection de petits objets en environnement ouvert. Notons toutefois ceux de [21] qui s'intéresse à la détection de piétons (36×18 pixels) en utilisant des ondelettes de Haar ou des HOG combinés avec un classifieur SVM [6]. Les autres travaux utilisent généralement des techniques de saillance ([25, 27]), inexploitables dans notre cas en raison de la complexité des fonds.

Les variétés, qui sont des sous-espaces plongés dans des espaces de grande dimension approximés localement par un espace euclidien (espace latent ou tangent), ont été utilisées de manière efficace pour modéliser les objets. Une variété peut être apprise de différente manière. Les techniques linéaires sont les plus simples, telles que l'Analyse Linéaire Discriminante [10], ou l'Analyse en Composante Principale [14]. Pour les techniques non-linéaires, certaines méthodes sont basées sur la propriété de conservation des distances géodésiques, comme Isomap [29]. D'autres, tels que Local Linear Embedding [26] et ses variantes, apprennent des approximations locales de la variété. Enfin, d'autres approches apprennent la variété d'une manière globale, comme les autoencodeurs [16]. Il est intéressant de noter que la plupart de ces algorithmes ont été conçus pour visualiser des données de grandes dimensions en les projetant dans un espace 2D ou 3D, et ne donnent que la fonction f, qui va de l'espace des descripteurs vers l'espace latent et non son inverse (requise pour notre approche de détection, comme expliqué plus loin). En revanche, l'ACP et les autoencodeurs permettent d'apprendre f et f^{-1}.

Les variétés ont déjà été utilisées par de nombreux auteurs sur des tâches de détection d'objets. Dans [22], Pentland *et al.* introduisent le concept d'*eigenfaces*, en utilisant l'ACP pour construire une variété linéaire des visages. Dans le même esprit, [2] utilise une ACP pour modéliser non seulement les objets mais aussi les fonds. Des résultats intéressants ont été obtenus sur des images de voitures et de piétons. L'utilisation de deux variétés est nécessaire pour contrebalancer l'absence d'aspect discriminatif. Dans [9], les auteurs utilisent des autoencodeurs pour construire un modèle de visage dans le but d'effectuer de la détection, mais le taux de fausse alarme est élevé. Dans [23], les auteurs proposent une méthode limitant ce taux.

Notre approche se base sur ces travaux récents, en combinant des descripteurs classiques (HOG, LBP) avec une technique d'apprentissage de variété. La contribution de nos travaux réside dans un nouveau modèle, les *autoencodeurs discriminants*. A notre connaissance, c'est la première fois qu'un tel modèle est proposé.

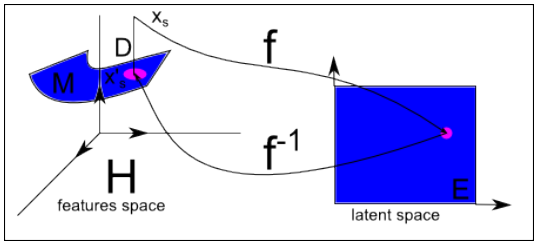

FIGURE 2 – Illustration du concept de *distance à la variété*. Soit X_s un vecteur et $X'_s = f^{-1} \circ f(X_s)$ sa projection sur la variété. $||X'_s - X_s||$ est la distance à la variété.

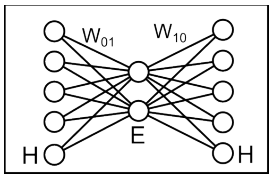

FIGURE 3 – Représentation d'un autoencodeur à 3 couches où H est l'espace des descripteurs et E l'espace latent.

3 Autoencodeurs discriminants

Avant de présenter les autoencodeurs discriminants, nous commençons par expliquer comment les variétés peuvent être utilisées dans un contexte de classification, puis comment apprendre les variétés avec des autoencodeurs.

3.1 Les variétés comme modèles génératifs

Soit $\mathcal{H}$ l'espace des descripteurs et $\mathbf{x} \in \mathcal{H}$ un descripteur visuel extrait d'une image (par exemple la signature d'une région de l'image). Nous rappelons que construire une variété Riemannienne $\mathcal{M}$ représentative des descripteurs est équivalent à trouver une fonction f telle que :

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

f est le plongement de $\mathcal{M}$ et est une fonction isométrique. Si $\mathbf{x}$ est sur la variété, $f^{-1} \circ f(\mathbf{x}) = \mathbf{x}$. $f^{-1} \circ f$ projette les points de l'espace des descripteurs sur la variété $\mathcal{M}$. Soit $P_{\mathcal{M}} = f^{-1} \circ f$, nous pouvons définir la distance à la variété comme $D_{\mathcal{M}}(\mathbf{x}) = ||\mathbf{x} - P_{\mathcal{M}}(\mathbf{x})||$ où $||x||$ représente la norme Euclidienne de $\mathbf{x}$. Le principe de cette projection est illustré Figure 2. Cette distance peut être utilisée pour modéliser une catégorie, en considérant que plus un descripteur est près de la variété, plus la probabilité qu'il fasse partie de la classe modélisée est importante. Le descripteur testé est étiquetté du label de la classe la plus probable.

3.2 Autoencodeurs

Les autoencodeurs sont des réseaux de neurones symétriques, qui apprennent la fonction identité sous contrainte. Un autoencodeur simple est montré Figure 3 mais des architectures plus complexes peuvent être utilisées. Nous nous sommes limités à cette architecture, cependant, les équations restent valables pour des architectures plus complexes. Un neurone de la couche i est connecté à la couche $i+1$, et seulement à ces neurones. Soit W_{ij} la matrice des poids entre la couche i et j. Les couches sont numérotées de 0 (entrée) à N (couche centrale) puis de nouveau de N (couche centrale) à 0 (sortie), tel que montré Figure 3. Comme le réseau est symétrique $dimension(W_{ji}) = dimension(W_{ij}^T)$. Chaque couche j a sa sortie $\mathbf{r}(\mathbf{x})$, complètement définie par la couche précédente $\mathbf{x}$ et la matrice des poids W_{ij} par $\mathbf{r}(\mathbf{x}) = h(W_{ij}\mathbf{x})$. h est appelée la fonction d'activation et est typiquement la fonction sigmoïde. Quand la fonction d'activation h est linéaire pour toutes les couches, l'autoencodeur effectue une ACP [16]. A l'inverse, utiliser une ou plusieurs couches avec des fonctions d'activation non linéaires permet au réseau d'approximer n'importe quelle fonction [4].
Soit χ l'ensemble des vecteurs d'entrainements $\mathbf{x}$. L'autoencodeur standard minimise la fonction de coût [16] :

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

minimisant ainsi l'erreur de reconstruction des exemples positifs, $\tilde{\mathbf{x}}$ étant la reconstruction de $\mathbf{x}$ donnée par l'autoencodeur. Cette erreur est généralement minimisée en utilisant une descente de gradient stochastique, dans le cadre d'une rétropropagation de l'erreur [24]. f et son inverse sont ainsi apprises simultanément. L'espace latent E est disponible à la sortie de la couche centrale. Des techniques pour avoir une meilleure convergence existent, telles que l'utilisation de Restricted Boltzmann Machine [1] et de la Contrastive Divergence [11], ou bien l'utilisation du "dropout". Le lecteur intéressé peut se reporter à [12] et [13].
Dans le contexte de l'apprentissage de variété, le réseau est utilisé pour apprendre f, donnant ainsi un plongement des données [13]. Ici à l'inverse, le réseau est entièrement appris, ce qui donne la projection $P_{\mathcal{M}}(x)$ dont nous avions besoin. La distance à la classe, qui peut être utilisée en tant que score de classification, peut être calculée simplement par les équations données dans la section précédente.

3.3 Autoencodeurs discriminants

Nous introduisons ici le concept d'*autoencodeurs discriminants*, qui permettent d'utiliser les données de deux classes (notées par la suite χ^+, pour les positifs et χ^- pour les négatifs) et apprennent une variété qui sera adaptée à la reconstruction des positifs tout en assurant une mauvaise reconstruction des négatifs. Ainsi, nous essayons de tirer également parti de l'information des exemples négatifs.
Soit $t(\mathbf{x})$ le label de l'exemple $\mathbf{x}$, avec $t(\mathbf{x}) \in \{-1, 1\}$ et $e(\mathbf{x})$ la distance de cet exemple à la variété, avec $e(x) = ||\mathbf{x} - \tilde{\mathbf{x}}||$. La nouvelle fonction à optimiser est :

$$L_d(\chi^+ \cup \chi^-) = \sum_{\mathbf{x} \in \chi^+ \cup \chi^-} \max\left(0, t(\mathbf{x}) \cdot (e(\mathbf{x}) - 1)\right) \quad (3)$$

qui n'est rien d'autre que la fonction de coût Hinge Loss utilisée dans de nombreux algorithmes de classifications

tels que le SVM. En pratique, nous utilisons une version légèrement différente de la Hinge Loss classique, comme proposé par [20] – la fonction standard étant $L_d = \sum_{\mathbf{x}\in\chi^+\cup\chi^-} \max(0, 1 - t(\mathbf{x}) \cdot e(\mathbf{x}))$ – plus adaptée à notre problème, étant donné que les erreurs de reconstruction sont toutes positives. En un sens, notre problème se rapproche plus des algorithmes d'apprentissage de métrique que de ceux de classification. Quand le minimum est atteint, les exemples positifs (resp. négatifs) ont une erreur de reconstruction plus petite (resp. plus grande) que 1.

Pour optimiser la fonction de coût, nous utilisons une rétropropagation de l'erreur. Repartant des équations de l'autoencodeur, nous avons $\mathbf{y} = h(W_{10}\mathbf{z})$ et $\mathbf{z} = k(W_{01}\mathbf{x})$. Comme proposé par [13], nous prenons comme fonction k la fonction identité, et une sigmoïde pour h. k étant l'identité, la transformation apprise de l'espace latent vers l'espace des descripteurs est linéaire. k est introduit pour conserver la généralité des équations. Pour simplifier les notations, notons $\mathbf{u}$ et $\mathbf{v}$ tels que : $\mathbf{u} = W_{10}\mathbf{z}$ et $\mathbf{v} = W_{01}\mathbf{x}$. L'objectif alors d'estimer les coefficients w_{ki} de W_{01} et W_{10} en minimisant $L(\chi^+ \cup \chi)$. La valeur optimal des w_{ki} vérifie : $\frac{\partial L}{\partial w_{ki}} = 0$, qui peut être résolue par une descente de gradient. Les dérivées partielles peuvent s'écrire :

$$\frac{\partial L}{\partial w_{ki}} = \frac{\partial L}{\partial \mathbf{e}^i} \cdot \frac{\partial \mathbf{e}^i}{\partial \mathbf{y}^i} \cdot \frac{\partial \mathbf{y}^i}{\partial \mathbf{u}^i} \cdot \frac{\partial \mathbf{u}^i}{\partial w_{ki}} \tag{4}$$

avec :

$$\frac{\partial \mathbf{e}^i}{\partial \mathbf{y}^i} = -1; \quad \frac{\partial \mathbf{y}^i}{\partial \mathbf{u}^i} = \frac{\partial h(\mathbf{u}^i)}{\partial \mathbf{u}^i}; \quad \frac{\partial \mathbf{u}^i}{\partial w_{ki}} = z^k \tag{5}$$

De plus, dans le cas d'une fonction sigmoïde, $\frac{\partial h(\mathbf{u}^i)}{\partial \mathbf{u}^i} = \mathbf{y}^i \cdot (1 - \mathbf{y}^i)$. Jusqu'ici, les équations sont identiques à la rétropropagation d'erreur classique. Ensuite, nous introduisons la fonction hinge loss avec :

$$\frac{\partial L}{\partial \mathbf{e}^i} = \begin{cases} \mathbf{e}^i & \text{si } t(\mathbf{x}) \cdot (e(\mathbf{x}) - 1) > 0 \\ 0 & \text{sinon} \end{cases} \tag{6}$$

Nous obtenons la descente de gradient suivante : $\Delta w_{ki} = -\eta\delta_i\mathbf{z}^k$, avec $\delta_i = \mathbf{e}^i \frac{\partial h(\mathbf{u}^i)}{\partial \mathbf{u}^i}$ si $t(\mathbf{x}) \cdot (e(\mathbf{x}) - 1) > 0$ et 0 sinon.

Pour la couche cachée (nous ne donnons les dérivations que pour une couche cachée, cependant les équations sont les mêmes pour les couches suivantes) :

$$\frac{\partial L}{\partial w_{lk}} = \frac{\partial L}{\partial \mathbf{z}^k} \cdot \frac{\partial \mathbf{z}^k}{\partial \mathbf{v}^l} \cdot \frac{\partial \mathbf{v}^l}{\partial w_{lk}} \tag{7}$$

Les deux derniers termes ne changent pas, cependant le troisième devient :

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

ce qui nous donne :

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

FIGURE 4 – Régions de 100×100 pixels centrées sur des voitures (base VeDAI). La taille, les reflets spéculaires, les ombres et occlusions rendent la tâche de détection difficile.

$$= -\sum_n e^n \frac{\partial(h(\mathbf{u}^n))}{\partial \mathbf{u}^n} w_{kn} = -\sum_n \delta(n) w_{kn} \tag{8}$$

L'incrément devient $\Delta w_{lk} = -\eta\delta_k\mathbf{x}^l$ avec $\delta_k = \frac{\partial h(\mathbf{v}^k)}{\partial \mathbf{v}^k} \sum_n \delta(n) w_{kn}$ si $t(\mathbf{x}) \cdot (e(\mathbf{x}) - 1) > 0$ et 0 sinon. Pour les deux incréments, η est le pas d'apprentissage. Différentes manières existent pour l'optimiser. Le lecteur intéressé pourra se référer à [19] pour plus d'information.

Ces équations peuvent être rendues plus robustes par l'ajout d'une marge w à l'équation $t(\mathbf{x}) \cdot (e(\mathbf{x}) - 1) > 0$, qui devient $t(\mathbf{x}) \cdot (e(\mathbf{x}) - 1) + w > 0$. w est déterminée par validation croisée.

Finalement, $e(x_{new})$, l'erreur de reconstruction pour un nouveau vecteur x_{new} peut être utilisée directement comme score de classification.

4 Expériences

La base VeDAI. Nous avons effectué la validation expérimentale sur une base conçue pour évaluer les algorithmes de détection de véhicules faiblement résolus, qui sera prochainement rendue publique. Elle contient plus de 1200 images (1024x1024 pixels, 3 couleurs), avec de nombreux fonds et véhicules différents (ainsi qu'illustré Figure 1). Ces images sont extraites du site Utah ARGC [1], plus précisément du 2012-HRO-6-inch-orthophotography set. La résolution des images est de 12.5×12.5 cm par pixel. Les voitures ont donc une taille aux alentours de 20×40 pixels. Leur détection est difficile à cause des occlusions, des réflexions, des ombres et des spécularités comme montré Figure 4. La variation intra-classe est importante. Nous avons utilisé un processus de validation croisée à 10 folds (lots). Les 1210 images sont réparties en 10 folds, chacun contenant 134 voitures réparties sur 121 images. Durant l'évaluation, 9 folds sont utilisés pour l'apprentissage, tandis que le dernier fold est utilisé pour le test. Chaque fold est utilisé une fois en tant qu'ensemble test.

Pipeline de détection. Notre algorithme se situe dans le cadre classique des algorithmes dits à fenêtre glissante, et utilise l'apprentissage de variété pour noter les fenêtres. Toutes les régions rectangulaires d'une proportion largeur/longueur donnée sont évaluées par notre classifieur.

1. http ://gis.utah.gov/

TABLE 1 – Résultats sur la base VeDAI

Détecteur	mAP		Rappel @ 0.01 FPPI		Rappel @ 0.1 FPPI		Rappel @ 1 FPPI	
Deformable Part Model [8]	60.5±4.2		13.4±6.8		31.4±5.8		74.5±4.5	
Descripteurs	HOG	LBP	HOG	LBP	HOG	LBP	HOG	LBP
SVM (1er étage)	58.9±3.5	52.9±3.5	13.2±5.1	10.5±6.3	30.4±3.9	30.8±3.9	72.1±4.1	63.2±5.0
SVM puis Standard AE	30.0±3.9	34.2±4.4	1.5 ±1.6	3.6±1.6	6.8 ±1.8	13.3±3.9	39.5±4.1	43.2±7.1
SVM fusion Standard AE	58.8±3.8	55.5±4.0	12.9±3.5	13.4±4.7	34.0±4.5	32.1±3.1	71.8±5.4	66.7±5.0
SVM puis AE Discriminant	68.0±4.2	59.2±3.6	**21.2±6.9**	14.5±4.5	46.7±6.8	39.2±4.0	78.7±3.4	70.8±4.3
SVM fusion AE Discriminant	**69.6±3.4**	**60.0±3.7**	20.4 ±6.2	**17.3±6.5**	**49.0±3.6**	**40.0±3.5**	**80.3±3.1**	**72.0±5.3**

En pratique, cela est fait par le biais d'une grille multi échelles sur l'image. Nous avons utilisé un pas de 8 pixels et un ratio entre les échelles de $2^{\frac{1}{10}}$, comme effectué dans [8] – on a un rapport $2^{1/10}$ pour la 1ère échelle, $2^{2/10}$ pour la 2ème *etc.* Comme le ratio largeur/longueur d'un véhicule varie selon son orientation, différents classifieurs ont été appris pour différents ratios. Ces ratios ont été déterminés par clustering des aires des positifs. 4 échelles ont été utilisées, la distance à la cible étant approximativement connue. Pour améliorer l'efficacité du détecteur, nous utilisons une cascade à deux étages. Le premier étage est constitué de 12 SVM linéaires basés sur des HOG ou des LBP (2 orientations $\times$ 6 ratios largeur/longueur), tandis que le second étage affine les scores de détection en utilisant 12 autoencodeurs discriminants (chacun apparié avec le SVM correspondant). Les détecteurs du premier étage utilisent toutes les données d'apprentissage, tandis que ceux du second étage utilise les négatifs difficiles obtenus après ce premier apprentissage. Les négatifs difficiles sont tous les exemples de la base de données que le SVM classe avec un bon score. Durant le test, les 12 SVM parcourent l'image. Seules les fenêtres avec un score supérieur à -1.0 sont gardées. Ensuite, comme habituellement dans un algorithme par fenêtre glissante, une étape de filtrage des non-maxima est effectuée. Nous avons utilisé une stratégie gloutonne simple, qui consiste à prendre les fenêtres de meilleur score et enlever les fenêtres qui recouvrent celles-ci dans un rayon égal à la moitié de la largeur de la fenêtre. Finalement, les fenêtres sélectionnées sont reclassées avec l'autoencodeur (standard ou discriminant). Notre algorithme peut prendre en entrée n'importe quel descripteur. Afin d'illustrer ce fait, nous avons effectué les expériences avec les descripteurs HOG et les descripteurs LBP. Nous avons combiné les deux scores issus des deux étapes de la cascade. Dans ce cas, le résultat final est obtenu par combinaison linéaire avec $\alpha S_{autoencodeur}(\mathbf{x}) + (1-\alpha)S_{SVM}(\mathbf{x})$, où α est fixé par validation croisée. La validation des paramètres (nombre de neurones, paramètre α), l'apprentissage se fait sur 8 des folds, et l'effet des paramètres est visualisé sur le 9ème fold. Une fois tous ces paramètres bloqués, un apprentissage avec les paramètres validés est effectué sur les 9 folds, et le résultat est issu d'un test sur le 10ème fold. Cette procédure est ensuite effectuée 10 fois pour obtenir les résultats. α et le nombre de neurones n'est pas le même pour chaque fold, mais en pratique ces paramètres sont proches.

Résultats. Nous avons mesuré la performance des détecteurs par la mean Average Precision (mAP) sur les 10 folds, ainsi que par la moyenne du taux de détection (rappel) pour 0.01 faux positif par image (FPPI), 0.1 FPPI et 1 FPPI. L'Average Precision est calculée par une extrapolation de la courbe précision-rappel en 11 points, tel que fait dans de nombreux benchmarks [7]. La moyenne et l'écart-type sont ensuite calculés sur les 10 folds. Les résultats ainsi obtenus sont donnés Table 1. Premièrement, nous avons observé que l'autoencodeur standard (n'utilisant que des exemples positifs), ne donne pas de bons résultats. Même combiné au premier étage de la cascade, il n'améliore pas significativement les performances du SVM. D'un autre côté, l'autoencodeur discriminant donne de bien meilleurs résultats que l'autoencodeur classique (+38.0 de mAP avec HOG, +25.0 avec LBP), et est significativement meilleur que le SVM (+9.1 de mAP avec HOG et +6.3 avec LBP), alors qu'il utilise les mêmes exemples d'entrainement. Pour les différents points de fonctionnement, l'autoencodeur discriminant gagne en performances. Le grand écart-type pour 0.01 FPPI rend la conclusion non fiable pour ce point, mais le gain est important pour les deux descripteurs. La combinaison du score de l'autoencodeur discriminant et du score du SVM donne des résultats légèrement meilleurs que l'autoencodeur discriminant seul, montrant ainsi que l'autoencodeur a conservé la majorité de l'information du premier étage de la cascade. Enfin, lorsqu'on compare avec le Deformable Part Model de [8] (en utilisant la release 5 et en désactivant l'apprentissage des parties du fait de la petite taille des objets), un des meilleurs algorithmes du moment, le gain est d'environ 10% de mAP. Ces résultats montrent clairement que non seulement les autoencodeurs discriminants donnent de meilleurs résultats que les autoencodeurs classiques, mais qu'ils permettent d'améliorer significativement les performances d'un SVM ou du DPM dans une tâche de détection de petits objets.

5 Conclusions

Cet article introduit le concept *d'autoencodeur discriminant* qui, en plus d'optimiser la reconstruction d'exemples positifs, éloigne les exemples négatifs de l'espace de reconstruction. Nous avons de plus montré comment apprendre de tels autoencodeurs, en s'inspirant de techniques d'apprentissage de métriques. Dans le contexte de la détection de petites cibles, nous avons montré que les autoenco-

deurs discriminants donnent de bien meilleurs résultats que les autoencodeurs classiques et permettent une amélioration significative des performances quand ils sont associés à un SVM linéaire.

Références

[1] D. Ackley, G. Hinton, and T. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive Science*, 9 :147–169, 1985.

[2] G. Carvalho, L. Moraes, G. Cavalcanti, and T. Ren. A weighted image reconstruction based on pca for pedestrian detection. In *International Joint Conference on Neural Networks*, 2011.

[3] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20 :273–297, 1995.

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

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE CVPR*, 2005.

[6] M. Enzweiler and D. Gavrila. Monocular pedestrian detection : Survey and experiments. In *IEEE PAMI*, volume 31, pages 2179–2195, 2009.

[7] M. Everingham, L. V. Gool, Williams, C. K. I., J. Winn, and A. Zisserman. The pascal voc challenge. *IJCV*, 88 :303–338, 2010.

[8] P. Felzenszwalb, R. Girshick, D. Mcallester, and D. Ramanan. Object detection with discriminatively trained part based models. In *IEEE PAMI*, volume 32, pages 1627–1645, 2009.

[9] R. Feraud, O. Bernier, J. Viallet, and M. Collobert. A fast and accurate face detector based on neural networks. In *IEEE PAMI*, volume 23, pages 42–53, 2001.

[10] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7 :179–188, 1936.

[11] G. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14 :2002, 2000.

[12] G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv :1207.0580*, 2012.

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

[14] H. Hotelling. Analysis of a complex statistical variable into principal components. *Journal of educational psychology*, 24 :417, 1933.

[15] A. Kembhavi, D. Harwood, and L. S. Davis. Vehicle detection using partial least squares. In *IEEE PAMI*, volume 33, pages 1250–1265, 2011.

[16] M. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *Am. Inst. of Chem. Engineers Jour.*, 37 :233–243, 1991.

[17] C. Lampert, M. Blaschko, and T. Hofmann. Beyond sliding windows : Object localization by efficient subwindow search. In *IEEE CVPR*, 2008.

[18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86 :2278–2324, 1998.

[19] Y. LeCun, L. Bottou, G. Orr, and K. Müller. Efficient backprop. In *Neu. net. : Tricks of the trade*. 1998.

[20] A. Mignon and F. Jurie. Pcca : A new approach for distance learning from sparse pairwise constraints. In *CVPR*, 2012.

[21] S. Munder. An experiment study on pedestrian classification. volume 28, 2006.

[22] A. Pentland. Viewbased and modular eigenspaces for face recognition. In *IEEE CVPR*, 1994.

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

[24] D. Rumelhart, G. Hinton, and R. Williams. Learning internal representations by error propagation. Technical report, DTIC Document, 1985.

[25] U. Rutishauser, D. Walther, C. Koch, and P. Perona. Is bottom-up attention useful for object recognition ? In *IEEE CVPR*, 2004.

[26] L. Saul and S. Roweis. Think globally, fit locally : unsupervised learning of low dimensional manifolds. *JMLR*, 4 :119–155, 2003.

[27] H. Seo and P. Milanfar. Visual saliency for automatic target detection, boundary detection, and image quality assessment. In *IEEE ICASP*, 2010.

[28] 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*, 35 :973–979, 2004.

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

[30] P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *IJCV*, 63 :153–161, 2005.

[31] X. Wang, T. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In *ICCV*, pages 32–39, 2009.

[32] X. Zhang, X. Gao, and T. Caelli. Parametric manifold of an object under different viewing directions. In *ECCV*, pages 186–199, 2012.

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