Création automatique d'un détecteur adapté à la scène

Thierry Chesnais, Nicolas Allezard, Yoann Dhome, Thierry Chateau

Création automatique d'un détecteur adapté à la scène

Thierry Chesnais[1] Nicolas Allezard[1] Yoann Dhome[1] Thierry Chateau[2]

[1] CEA, LIST, Laboratoire Vision et Ingénierie des Contenus, BP 94, F-91191 Gif-sur-Yvette, France,
[2] Lasmea, UMR6602, CNRS, Université Blaise Pascal, Clermont-Ferrand, France

{thierry.chesnais, nicolas.allezard, yoann.dhome}@cea.fr
thierry.chateau@lasmea.univ-bpclermont.fr

Résumé

Cet article se place dans le cadre de la détection temps-réel de piétons à l'aide d'une caméra fixe non calibrée. Plus précisément, il s'agit d'adapter un classifieur au contexte d'une scène. L'approche développée ici repose sur une méthode offline semi-supervisée basée sur l'utilisation d'un oracle. Le rôle de ce dernier est de labelliser automatiquement une vidéo pour obtenir une base d'apprentissage spécialisée. Il est constitué de plusieurs détecteurs, chacun appris sur un signal différent (apparence, segmentation fond/forme, flot optique), dont les réponses sont ensuite fusionnées. Un détecteur final, contextualisé, est ensuite appris sur cette base. Cette méthode est totalement automatique et ne nécessite aucune connaissance a priori de la scène et peut donc être utilisée lors de la phase de déploiement d'un réseau de caméras.

Mots Clef

vidéosurveillance, détection de piétons, oracle.

Abstract

This article tackles the real-time pedestrian detection problem using a stationary uncalibrated camera. More precisely we try to specialize a classifier by taking into account the context of the scene. To achieve this goal, we introduce an offline semi-supervised approach which uses an oracle. This latter must automatically label a video, in order to obtain contextualized training data. The proposed oracle is composed of several detectors. Each of them is trained on a different signal: appearance, background subtraction and optical flow cues. Then we merge their responses and keep the more confident detections. A specialized detector is then built on the resulting dataset. Designed for improving camera network installation procedure, the presented method is completely automatic and does not need any knowledge about the scene.

Keywords

videosurveillance, pedestrian detection, oracle.

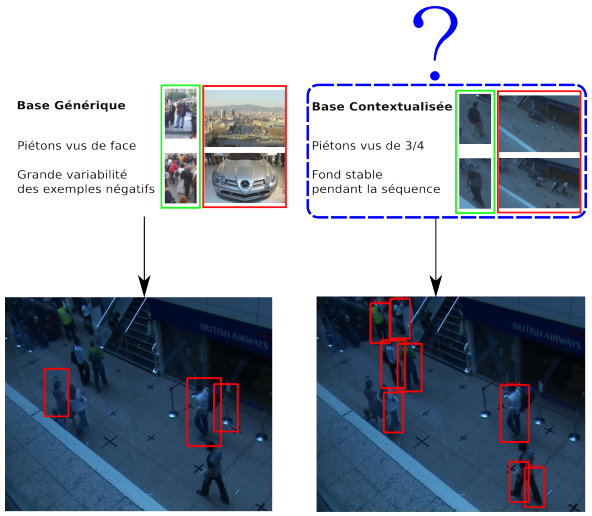


FIGURE 1 – Un classifieur appris sur une base générique (image de gauche) donne de moins bons résultats qu'un détecteur spécialisé (image de droite) lorsque le point de vue de l'ensemble d'apprentissage et celui de la détection diffèrent trop.

1 Introduction

En vision par ordinateur, la recherche d'algorithmes de détection d'objets, notamment de détection de piétons, fonctionnant en temps réel (10 images par seconde) et de manière robuste, reste encore aujourd'hui une tâche difficile. Les principales applications concernent la vidéosurveillance (observation de scènes à l'aide d'une ou plusieurs caméras fixes) ou bien les systèmes d'aide à la conduite (caméras embarquées dans un véhicule servant à repérer les humains aux abords de la route). Plusieurs articles récents présentent les dernières avancées dans ces domaines : [6] [10] [5]. L'une des principales difficultés rencontrées est la gestion de la grande variabilité d'apparence des piétons (tailles, occultations) ainsi que la variation des points de vue, différents d'une caméra à l'autre. Dans l'article nous nous plaçons dans le contexte de la vidéosurveillance avec

notamment l'utilisation d'une caméra fixe.

Les approches classiques pour faire de la détection d'objets reposent sur des méthodes d'apprentissage. On peut citer les séparateurs à vaste marge [18, 4] ou les méthodes de type boosting [8, 19]. Elles consistent à extraire des caractéristiques discriminantes piétons/non piétons à partir d'une base précédemment labellisée. Le détecteur compare ensuite les caractéristiques d'une nouvelle image avec celles de la base pour prédire la présence d'un piéton.

Mais, pour obtenir de bonnes performances en utilisant ces techniques, la base d'apprentissage doit être riche et posséder des caractéristiques similaires au contexte dans lequel le détecteur sera utilisé. Il est donc important que la base prenne au maximum en compte les spécificités de la scène. Or l'obtention d'une telle base, contenant des milliers d'exemples labellisés (objet ou non objet) et alignés (objet centré dans l'imagette), est une opération manuelle coûteuse. Il est donc impossible lors du déploiement d'un système de vidéosurveillance de constituer une base pour chaque caméra. Dans ce cas, l'approche retenue consiste généralement à n'utiliser qu'une base générique en espérant que le classifieur obtenu aura des performances acceptables (voir la figure 1).

Ces dernières années, plusieurs approches ont été proposées pour répondre au problème de la construction automatique de la base d'apprentissage afin d'exploiter au mieux les très nombreuses informations enregistrées par les caméras de vidéosurveillance. Les méthodes semi-supervisées sont une réponse. Elles utilisent des données labellisées mais aussi les grandes quantités de données non labellisées contenues dans la base d'apprentissage.

Les exemples de la base ainsi constituée, dite contextualisée car possédant de nombreuses observations spécifiques à la caméra étudiée, peuvent ensuite être incorporés dans le classifieur spécialisé de différentes manières :

– soit attendre d'avoir un ensemble important d'exemples et l'utiliser pour faire l'apprentissage en une seule fois, c'est le principe des **méthodes hors ligne (ou offline)** ;
– soit entraîner le classifieur sur les nouveaux exemples dès que ceux-ci sont disponibles, c'est le principe des **méthodes en ligne (ou online)** démocratisé dans le cadre du boosting par Grabner et al. [11].

Notre but est de proposer une nouvelle méthode semi-supervisée, fondée sur l'utilisation d'un oracle, pour fabriquer de manière automatique un classifieur adapté au contexte de la caméra. Nous avons choisi une approche offline pour l'apprentissage du détecteur pour deux raisons. Notre action se déroule lors de l'installation d'un réseau de caméras. Nous disposons donc du temps nécessaire pour obtenir un grand nombre d'exemples et en tirer profit. À l'inverse, lors de l'exploitation, la détection doit s'effectuer en temps réel. Faire un entraînement pendant cette phase nous prive de ressources utiles par ailleurs. Enfin, même si des méthodes online robustes [12] existent, ces dernières présentent un risque de dérive qui n'est pas compatible avec une utilisation dans la durée.

Dans cette étude, nous nous intéressons donc principalement au fonctionnement de l'oracle. Après avoir détaillé les principales méthodes semi-supervisées existantes, nous décrirons, dans la partie 3, notre stratégie de constitution de l'oracle. La partie 4 contient une évaluation de l'algorithme qui consiste en une analyse du comportement de l'oracle et en une comparaison des performances du détecteur spécialisé avec un classifieur de l'état de l'art.

2 Travaux reliés

Il existe de nombreuses familles de méthodes semi-supervisées. Les plus courantes sont le self-learning, le cotraining et l'utilisation d'un oracle.

Le **self-learning** [14] consiste à utiliser la réponse du classifieur pour labelliser un exemple. Les exemples pour lesquels le classifieur a une grande confiance sont ajoutés à la base d'apprentissage. Cette méthode est assez peu robuste car elle souffre d'un défaut de dérive. En effet, les exemples mal labellisés vont perturber les réponses pour les exemples suivants et risquent d'accentuer le phénomène. De plus si le seuil de confiance du classifieur est trop faible, de nombreux faux positifs vont se retrouver dans la base. À l'inverse avec un seuil trop haut, seuls les exemples parfaitement reconnus et donc apportant peu d'informations sont retenus.

Le **cotraining** introduit par Blum et Mitchell [3] est un formalisme dans lequel deux classifieurs sont entraînés en utilisant pour chacun des sous-parties indépendantes de la base de données. Par exemple dans [13], les auteurs entraînent deux classifieurs, l'un sur l'apparence et l'autre sur la soustraction de fond. Cet algorithme repose sur le fait que les classifieurs entraînés sur différentes parties des données doivent décerner le même label à une observation. Si un des classifieurs répond avec une confiance suffisamment élevée mais que le deuxième n'est pas sûr, alors l'observation est ajoutée dans la base de ce dernier. Pendant l'entraînement les classifieurs s'auto-améliorent. À la fin de la phase d'apprentissage, plusieurs classifieurs performants sont disponibles. Même si le fait d'utiliser deux classifieurs indépendants limite leurs dérives, les observations sont toujours labellisées directement grâce à leurs sorties. Le phénomène de dérive n'est donc pas exclu, les données étant rarement totalement indépendantes.

Les **approches avec oracle** utilisent une entité externe au classifieur. Celle-ci est chargée de labelliser tous les exemples non étiquetés avant de les ajouter dans la base. Le classifieur final et l'oracle sont totalement décorrélés éliminant le principal risque de dérive. La capacité de l'oracle à repérer les bons exemples sans faire d'erreur de label détermine directement les performances du système final. De nombreux oracles différents ont déjà été proposés dans la littérature. Wu et al. [20] utilisent un classifieur par parties

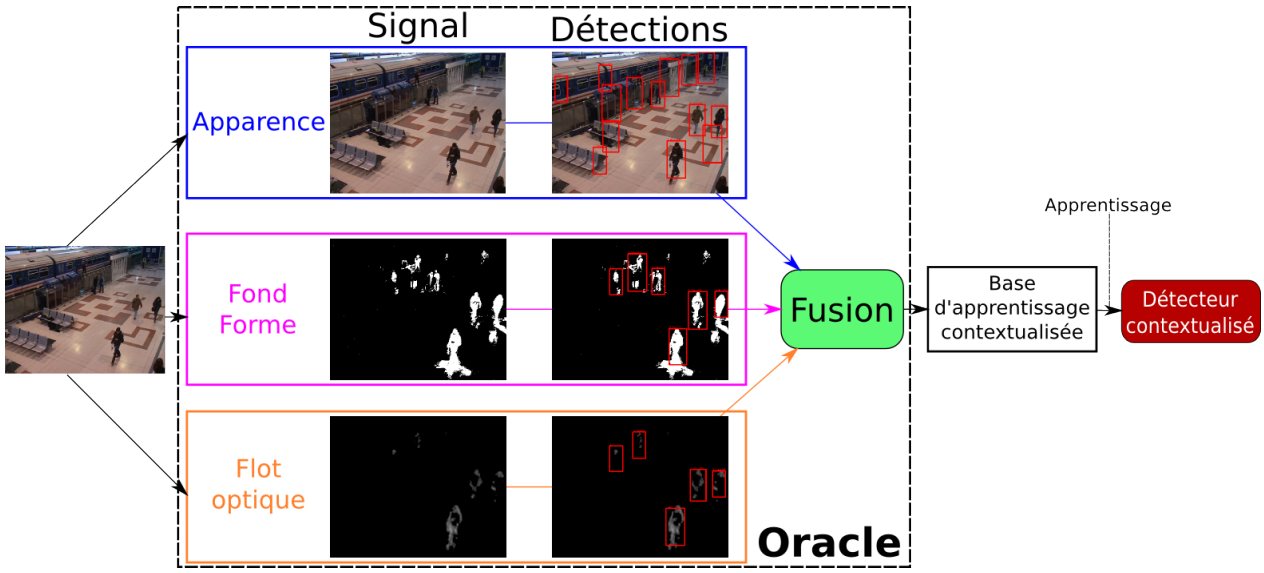


FIGURE 2 – **Schéma de fonctionnement de l'oracle.** L'oracle est constitué de trois classifieurs indépendants travaillant sur l'apparence, la segmentation fond/forme et le flot optique. Chacun fournit un ensemble de détections qui doivent être fusionnées (voir la figure 3) pour former une base d'apprentissage contextualisée. Le détecteur final est entraîné à partir des données contenues dans cette base.

basé sur l'apparence. Si l'oracle retrouve suffisamment de morceaux d'un piéton, alors l'exemple est ajouté à la base. Le problème de cet oracle est qu'il est constitué d'un seul classifieur n'analysant qu'un type de signal. À cela s'ajoute la difficulté de détecter des morceaux de piétons et de fusionner les réponses. Si l'oracle n'est pas performant sur la scène traitée alors le système ne fonctionne pas. Pour ajouter de la robustesse dans ce schéma, Stalder et al. [16] utilisent un oracle constitué de plusieurs étages. Chaque étage dépend directement de la sortie du précédent. La première étape consiste à détecter les piétons dans l'image. La deuxième partie initialise des trackers sur ces détections. Leur rôle est d'assurer la continuité spatio-temporelle entre les détections afin d'incorporer des exemples qui n'ont pas été détectés lors de la première étape. Cela permet, contrairement à l'oracle de Wu, de trouver des exemples difficiles pour le classifieur initial et donc importants pour la base contextualisée. Une dernière étape utilise le contexte 3D de la scène. Le principal inconvénient de ce système provient de sa structure : si un étage commet une erreur alors elle est obligatoirement transmise aux étages supérieurs pénalisant les performances de l'oracle.

Nous proposons donc un oracle fonctionnant de manière non séquentiel afin de gagner en robustesse.

3 Spécialisation du détecteur

Dans cette partie nous décrivons les différentes étapes de notre méthode pour fabriquer un oracle. À la manière du cotraining, ce dernier est constitué de plusieurs classifieurs travaillant sur des signaux indépendants. Contrairement à l'approche de [16] qui utilise un oracle séquentiel, notre approche permet, après une étape de fusion, de filtrer les mauvaises réponses fournies par chacun des signaux ce qui permet de construire une meilleure base d'apprentissage.

3.1 Oracle

Propriétés. Le rôle de l'oracle est d'annoter automatiquement une vidéo, c'est-à-dire de trouver des observations pertinentes et les labels correspondants.
L'oracle est donc un détecteur de piétons qui ne possède pas les mêmes propriétés que le détecteur final. Ce dernier doit non seulement être temps-réel, mais aussi détecter le maximum de piétons avec le minimum de faux positifs. En d'autres termes, il doit avoir un rappel [1] et une précision [2] élevés. En revanche dans le cas de l'oracle il est possible de relâcher certaines contraintes. Tout d'abord l'oracle n'est pas nécessairement temps réel puisque notre méthode est en deux temps, la première phase pouvant se dérouler sur une période relativement longue. De plus notre but étant de créer une base d'apprentissage, il n'est pas pénalisant d'omettre certains piétons pourvu que la vidéo soit assez longue. L'oracle peut donc avoir un rappel plus faible que le détecteur final. En revanche afin de minimiser le plus possible l'erreur de labellisation de la base contextualisée, l'oracle doit être le plus précis possible.

Constitution. Pour répondre à ce cahier des charges, nous avons décidé d'utiliser une combinaison de briques élémentaires (voir la figure 2). Une brique est constituée par un classifieur. À la manière du cotraining, nous entraî-

1. rappel $= \frac{\text{nombre de bonnes détections}}{\text{nombre de piétons}}$
2. précision $= \frac{\text{nombre de bonnes détections}}{\text{nombre de détections}}$

nons au préalable les détecteurs sur des signaux aussi indépendants que possible. Cela permet, lors d'une étape de fusion (voir la figure 3), de croiser les réponses de tous ces classifieurs afin de corriger les erreurs de l'un grâce à la sortie des autres.
Pour cet article nous avons testé trois signaux : l'apparence (gradient), la segmentation fond/forme [17] et le flot optique[2]. Chaque classifieur est basé sur un descripteur différent et donc sur une base d'apprentissage générique différente. Les trois fonctionnent en parallèle.

Création d'une base d'apprentissage contextualisée. Pour construire la base, il faut maintenant scanner des images du contexte à l'aide des trois classifieurs précédemment entraînés. Pour chaque position et échelle testées dans l'image nous obtenons un score de confiance par détecteur. Il est ensuite nécessaire de fusionner ces cartes de confiance. Or ces classifieurs étant a priori indépendants, les scores fournis par chacun d'eux ne sont pas comparables. Deux choix s'offrent à nous : travailler directement sur les cartes de confiance après normalisation pour les rendre comparables, ou bien travailler sur les détections fournies par les classifieurs après un algorithme de regroupement. Nous avons choisi cette dernière option. De la même manière que [4], nous utilisons un algorithme de clustering, le meanshift, pour regrouper toutes les boîtes qui ont un score positif. En sortie du regroupement, chaque détection possède un score. Celui-ci correspond à la somme des scores des boîtes qui ont contribué à ce regroupement. Nous obtenons pour une même image un ensemble de détections (boîte + score) par classifieur.

Les **exemples positifs** sont constitués par les observations pour lesquelles les classifieurs génériques ont une confiance suffisamment élevée en leurs labels. Cette étape de fusion est délicate car si elle est trop restrictive, des exemples difficiles donc intéressants risquent d'être oubliés. À l'inverse si elle est trop lâche, la base va être polluée par de nombreux faux positifs.
Une détection est incorporée dans la base uniquement si elle apparaît en sortie de plusieurs classifieurs. Le vote majoritaire est effectué comme expliqué sur la figure 3. Une première association exhaustive est réalisée entre les détections de l'apparence et celles de la segmentation fond/forme. Seules les boîtes d'apparence qui ont pu être appariées sont ajoutées à la base contextualisée. L'association est uniquement basée sur le recouvrement des boîtes. Comme [7], nous utilisons le critère de similarité suivant entre deux boîtes après regroupement : $\text{sim}_{\text{regroupés}}(B_1, B_2) = \frac{Aire(B_1 \cap B_2)}{Aire(B_1 \cup B_2)}$. Si la similarité entre deux boîtes est inférieure à 0.5 alors elles ne peuvent pas être appariées. Une deuxième association est ensuite réalisée entre les détections du flot optique et celles non associées de l'apparence. De la même manière les boîtes d'apparence appariées sont incorporées dans la base tandis que les boîtes non associées jusqu'ici sont définitivement écartées. Ce vote peut être vu comme la vérification des réponses de l'apparence par la segmentation fond/forme et le flot optique. En effet ces détecteurs fournissent une information de présence d'un piéton dans une zone étendue de l'image, tandis que celui sur l'apparence apporte une information de localisation précise. C'est la raison pour laquelle dans la fusion, nous privilégions les détections en provenance de l'apparence que nous vérifions avec les autres classifieurs.
Un dernier filtrage sur les scores des boîtes dans la base est effectué. Comme toutes les détections proviennent du classifieur appris sur l'apparence, les scores sont issus de cet unique détecteur et sont donc comparables. Pendant cette étape environ 50% des exemples, parmi les moins sûrs, sont supprimés de la base.

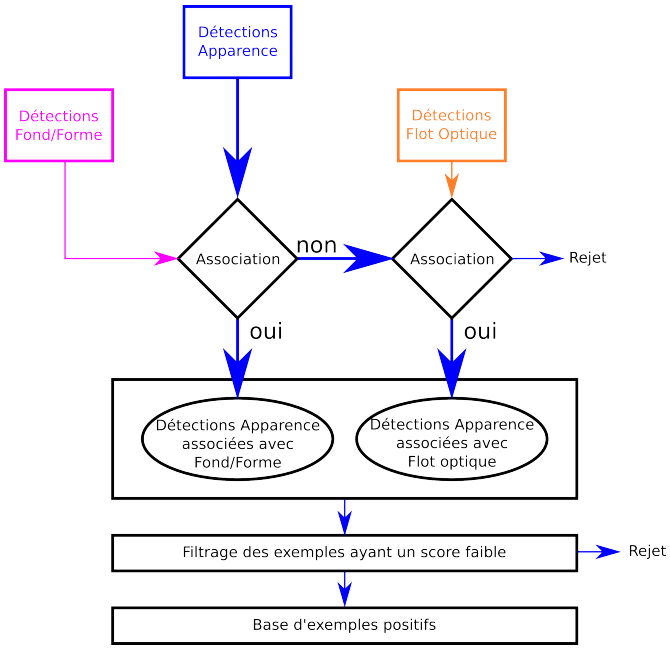


FIGURE 3 – Schéma de fonctionnement de la fusion des réponses des classifieurs au sein de l'oracle pour l'obtention de la base d'exemples positifs.

Une fois les exemples positifs obtenus, il faut générer les **exemples négatifs**. Notre stratégie consiste à tirer aléatoirement des boîtes dans toutes les images en évitant soigneusement les zones détectées précédemment. L'oracle ayant un rappel faible, tous les piétons ne sont pas détectés. Ils risquent donc de se retrouver dans la base des négatifs. Mais cela est peu probable du fait que les exemples négatifs sont beaucoup plus nombreux que les exemples positifs.
D'autre part, comme nous traitons une scène fixe, de nombreuses observations vont être semblables. Cela risque de conduire à une base qui n'est pas assez riche, avec un manque au niveau des exemples difficiles. Pour combler cette lacune nous avons décidé d'incorporer dans la base des exemples contenant des morceaux de piétons. Cela

revient à prendre une parcelle de l'image qui intersecte une détection fournie par l'oracle. Cependant les deux ne doivent pas trop se recouvrir pour ne pas vérifier le critère de similarité défini précédemment (c'est-à-dire : $\text{sim}_{\text{regroupés}}(B_{\text{piéton}}, B_{\text{négatif}}) < 0.5$).

3.2 Création du détecteur contextualisé

Pour créer un détecteur contextualisé, il suffit d'entraîner sur la base spécifique obtenue, un nouveau classifieur à l'aide de n'importe quelle méthode.
Une possibilité est de faire un apprentissage complet sur la base contextualisée contenant les trois signaux et de laisser l'algorithme de boosting choisir la combinaison des trois qui lui semble la plus pertinente. Cela aurait l'avantage de garder le maximum d'informations contenues dans la vidéo que l'on traite. Cependant dans nos expérimentations nous nous sommes rendus compte que le détecteur final obtenu avec cette approche avait des performances très proches d'un détecteur basé uniquement sur l'apparence, mais que le temps de calcul lors de la détection était significativement augmenté puiqu'il faut calculer tous les signaux. Nous avons donc opté pour un classifieur plus simple basé uniquement sur l'apparence.

4 Évaluations

Nous nous proposons maintenant de faire un bilan de cette méthode. L'étude est scindée en deux parties. Dans la première, nous étudions les caractéristiques de l'oracle présenté dans la partie 3. Dans la seconde, nous comparons les performances globales du détecteur ré-entraîné avec un détecteur de l'état de l'art et montrons qu'il est compétitif. L'algorithme a été testé sur différentes vidéos librement disponibles : PETS 2006 [3], PETS 2007 [4].

Nous évaluons notre système à l'aide de courbes précision-rappel en nous inspirant de la méthode décrite dans [1]. La précision est donnée par la formule $Pr = \frac{VP}{VP+FP}$ et le rappel par $R = \frac{VP}{P}$, où VP indique le nombre de vrais positifs, FP le nombre de faux positifs et P le nombre de positifs. Nous traçons les courbes représentant R en fonction de $(1 - Pr)$. Le point optimal est situé en (0, 1). À partir de ces mesures il est possible de définir la F-Mesure, $FM = 2.\frac{Pr.R}{Pr+R}$.
Le critère de similarité retenu entre la vérité terrain (VT) et une boîte (B) avant regroupement est le suivant :

$$\text{sim(VT, B)} = \frac{(VT_{cx} - B_{cx})^2}{(0.5 \times l(VT))^2} + \frac{(VT_{cy} - B_{cy})^2}{(0.5 \times h(VT))^2}$$

avec :
- cx et cy correspondant respectivement à l'abscisse et à l'ordonnée du centre de la boîte,
- $l(VT)$ et $h(VT)$ la largeur, respectivement la hauteur, de la boîte de vérité terrain.

3. http ://www.cvg.rdg.ac.uk/PETS2006/
4. http ://www.cvg.rdg.ac.uk/PETS2007/

Deux boîtes sont similaires si sim(VT, B) ≤ 1. Ce critère revient à définir un ellipsoïde autour du centre d'une boîte de la vérité terrain. Si le centre d'une détection est contenu dans l'ellipsoïde alors elle est considérée comme valable. Si deux détections sont associées à la même boîte de vérité terrain nous ne comptabilisons qu'une bonne détection. Les autres boîtes correspondent à de fausses détections.

4.1 Caractéristiques de l'oracle

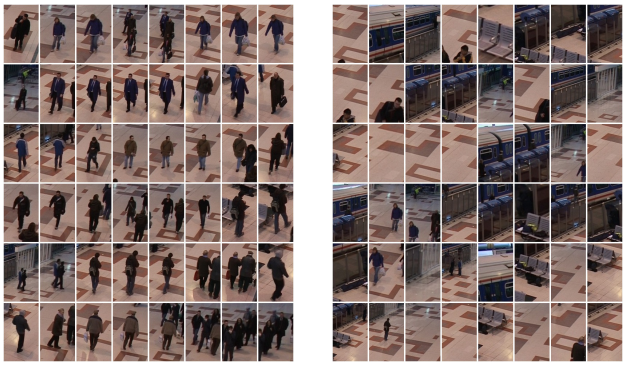
Dans ce paragraphe, nous étudions les caractéristiques de l'oracle pour vérifier qu'il correspond bien à nos attentes. Pour cela trois classifieurs génériques sont entraînés sur l'apparence, la segmentation fond/forme et le flot optique. Chaque classifieur est constitué de 400 itérations de boosting (Real-AdaBoost [9] [15]) sans cascade. Ne pas utiliser de cascade permet d'augmenter le rappel (plus de détections) au détriment de la précision (plus de faux positifs). Cette dernière est optimisée grâce à la fusion des classifieurs.
Le classifieur fonctionnant sur l'apparence utilise un descripteur basé sur le gradient. Nous avons utilisé le même descripteur pour le classifieur sur le flot optique : les composantes en x et en y du flot optique correspondent au gradient en x et en y de l'image d'apparence. En revanche, pour la segmentation fond/forme, nous avons décidé d'utiliser des ondelettes de Haar. Or ces dernières ne permettent pas de savoir si une zone homogène de l'image correspond à du fond ou à un objet. Pour combler cette lacune nous avons ajouté en complément des ondelettes la moyenne du bloc traité.
Nous entraînons le classifieur basé sur l'apparence à l'aide de la base de piétons de l'INRIA [5]. Cette dernière ne possédant pas d'information temporelle, les deux autres classifieurs ont été appris sur des bases construites spécialement. Le seuil de détection des classifieurs constituant l'oracle est fixé à 0 pour les deux séquences traitées.

PETS 2006. Ici nous travaillons sur la vue 4 du corpus PETS 2006 pour créer un classifieur. Nous avons choisi d'entraîner le détecteur final avec des exemples provenant de S2-T3-C et de tester les classifieurs, y compris ceux constituant l'oracle sur environ 1000 frames de S7-T6-B.

La figure 4 montre un échantillon de la base d'apprentissage obtenue après fusion des classifieurs constituant l'oracle. Pour les exemples positifs, la très grande majorité des imagettes correspond effectivement à un piéton. Cependant il existe deux problèmes principaux :
- Lorsque plusieurs piétons sont proches, le regroupement ne parvient pas toujours à les séparer correctement et à tendance à mal aligner l'exemple,
- La taille de l'imagette n'est pas toujours adaptée à l'objet.

En ce qui concerne les exemples négatifs, ils correspondent pour la plupart, comme nous le souhaitions, soit à des ob-

5. http ://pascal.inrialpes.fr/data/human/



(a) Exemples positifs

(b) Exemples négatifs avec morceaux de piétons

FIGURE 4 – Exemples choisis provenant de la base d'apprentissage du détecteur final pour PETS 2006

servations sans piétons, soit à des observations avec des morceaux de piétons.
Sur cette séquence, l'oracle obtient un rappel de 0.16 avec une précision de 0.99. Comme espéré, il possède une précision très importante. Celle-ci est obtenue sans connaissance a priori de la scène (comme le plan du sol ou un modèle 3D de la scène) et sans seuil à régler puisque tous les classifieurs de l'oracle ont un seuil de détection fixé à 0.

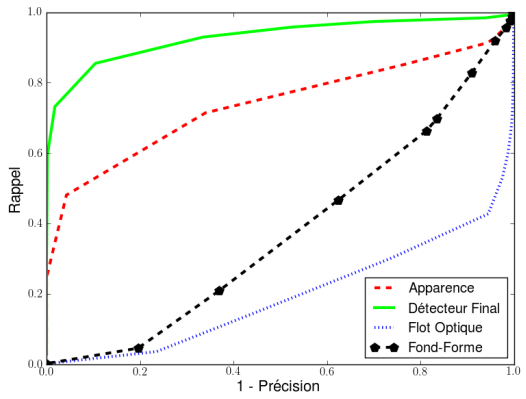


FIGURE 5 – Courbes précision-rappel sur la séquence PETS2006 pour les 3 classifieurs constituant l'oracle et pour le détecteur final

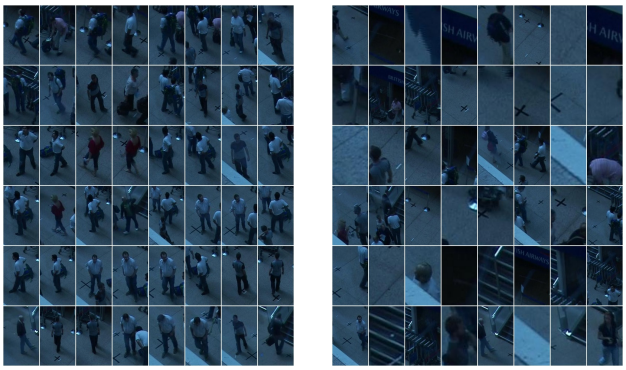
La figure 5 montre les courbes précision-rappel pour chaque classifieur constituant l'oracle ainsi que celle du détecteur final. Ce dernier n'utilise que l'apparence et il est entraîné de la même façon que le classifieur de l'oracle basé sur ce signal. Seule la base diffère entre les deux. 1800 exemples positifs et 8000 négatifs sont conservés lors du filtrage après la fusion des classifieurs et sont utilisés pour l'apprentissage.

TABLE 1 – Caractéristiques des détecteurs sur la séquence PETS 2006 - S7-T6-B - 4

	Rappel	Précision	F-Mesure
Apparence	0.71	0.66	0.69
Fond/Forme	0.47	0.38	0.42
Flot Optique	0.30	0.26	0.28
Oracle	0.49	**0.99**	0.65
Détecteur contextualisé	0.85	0.90	**0.87**

La table 1 indique la précision et le rappel de chaque classifieur pour le point où la F-Mesure est maximale. Remarquons que la précision des classifieurs appris sur la segmentation fond/forme et sur le flot optique est assez faible. Cela s'explique par le fait que ces classifieurs sont moins discriminants. Les détections sont plus dispersées autour de la cible et souvent deux cibles proches sont confondues après le regroupement. Comme expliqué lors de l'étape de fusion des classifieurs, ceux qui sont basés sur la segmentation fond/forme et sur le flot optique fournissent une information de présence d'un piéton dans une zone étendue de l'image, tandis que celui qui est basé sur l'apparence apporte une information de localisation précise.

PETS 2007. Le second corpus sur lequel nous avons testé notre méthode est PETS 2007. Nous travaillons sur la vue 3. Le point de vue de la caméra est très différent de celui qui a servi à créer la base d'apprentissage générique (piétons pris de face, bien droits). Dans le cas de PETS, les piétons sont observés de haut et sont généralement penchés.
La base spécialisée a été construite sur la séquence 3, tous les détecteurs sont évalués sur les 1000 premières images de la séquence 5.



(a) Exemples positifs

(b) Exemples négatifs avec morceaux de piétons

FIGURE 6 – Exemples choisis provenant de la base d'apprentissage du détecteur final pour PETS 2007

La figure 6 montre un échantillon de la base d'appren-

tissage obtenue après fusion des classifieurs constituant l'oracle.

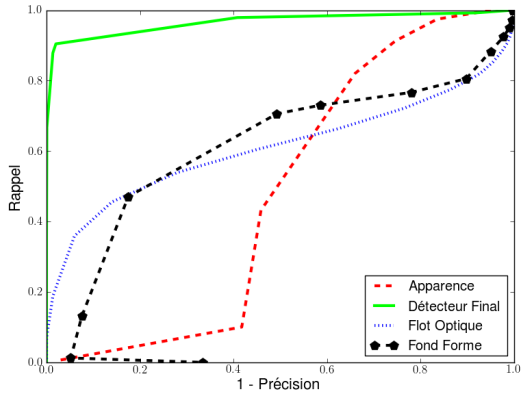


FIGURE 7 – Courbes précision-rappel sur la séquence PETS2007 pour les 3 classifieurs constituant l'oracle et pour le détecteur final

La figure 7 montre les courbes précision-rappel pour chaque classifieur constituant l'oracle ainsi que celle du détecteur final.

TABLE 2 – Caractéristiques des détecteurs sur la séquence PETS 2007 - S05 - 3

	Rappel	Précision	F-Mesure
Apparence	0.82	0.34	0.48
Fond/Forme	0.47	0.82	0.60
Flot Optique	0.54	0.72	0.62
Oracle	0.40	**0.99**	0.57
Détecteur contextualisé	0.90	0.98	**0.94**

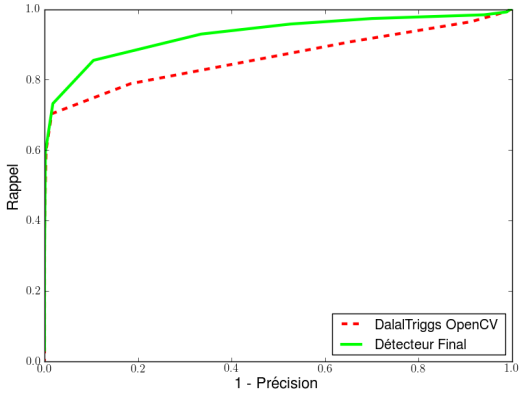
De la même manière que pour PETS 2006, la table 2 contient la précision et le rappel de chaque classifieur pour le point où la F-Mesure est maximale.
Contrairement au cas précédent, les classifieurs basés sur la segmentation fond/forme et sur le flot optique sont meilleurs que celui appris sur l'apparence. Cela s'explique par les deux faits suivants :

– Les exemples provenant de cette séquence et de la base générique ont des apparences trop différentes. Un classifieur basé uniquement sur ce signal offre donc de mauvaises performances.
– Il y a des groupes de personnes dans cette séquence. Les classifieurs qui ne sont pas discriminants (segmentation fond/forme et flot optique) ne sont donc pas trop pénalisés car une détection éloignée d'un piéton pourra être associée à celui d'à côté.

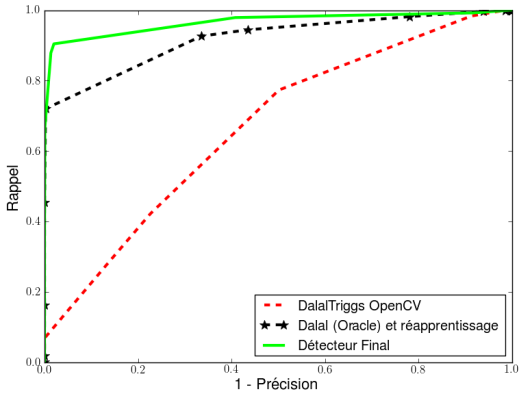
Comme les courbes le montrent, sur ces deux séquences, le détecteur spécialisé permet d'obtenir un meilleur rappel et une meilleure précision que le classifieur de l'oracle basé sur l'apparence.

4.2 Performances du détecteur spécialisé

Dans cette section nous comparons les performances du détecteur obtenu avec notre oracle et celles d'un détecteur de l'état de l'art. Nous avons retenu le détecteur de Dalal et Triggs [4] présent dans OpenCV. Nous appliquons la même méthode d'évaluation que précédemment. Pour prouver l'utilité de l'oracle présenté précédemment nous avons aussi ré-entraîné un classifieur à l'aide d'une base contextualisée extraite grâce au détecteur de Dalal et Triggs. Ce dernier fournit la positition des piétons tandis que les exemples négatifs sont obtenus aléatoirement. Le ré-apprentissage est le même dans tous les cas.



(a) PETS 2006



(b) PETS 2007

FIGURE 8 – Courbes précision-rappel sur les séquences PETS 2006 et 2007 pour notre détecteur final(vert), celui de Dalal et Triggs(rouge) et celui utilisant l'oracle basé sur le classifieur de Dalal (noir)

Les courbes de la figure 8 présentent les résultats des deux séquences retenues dans cette étude. Elles prouvent que lorsque la base d'apprentissage et la scène traitée sont trop

différentes, un détecteur contextualisé permet d'améliorer significativement les résultats.

5 Conclusion et Perspectives

Nous avons proposé une méthode semi-supervisée dont le but est de créer automatiquement un détecteur contextualisé. Pour cela nous avons fabriqué un oracle composé de plusieurs classifieurs, chacun travaillant sur un signal distinct. Un module de fusion est ensuite chargé d'agréger les réponses de chacun de ces classifieurs pour former une base d'apprentissage spécialisée. Cette dernière sert à l'entraînement d'un détecteur final qui intègre ainsi des informations du contexte.

Même si notre approche apporte des résultats satisfaisants, de nombreuses améliorations sont possibles.

- Comme nous l'avons remarqué, les classifieurs basés sur la segmentation fond/forme et le flot optique ne sont pas très précis. En effet, ils sont moins discriminants et ont donc tendance à fusionner des détections proches. Pour remédier à cela, il est possible d'utiliser des caméras calibrées afin de filtrer les sorties aberrantes. Néanmoins cela ajoute une étape manuelle lors de l'installation des caméras, raison pour laquelle nous ne l'avons pas pris en compte.
- Dans cette étude, nous avons choisi de construire un oracle à partir de trois signaux, mais il est tout à fait possible d'en choisir d'autres. Par exemple si nous disposions d'une tête stéréo, il serait envisageable d'apprendre un classifieur sur des cartes de disparités.
- Au niveau de l'apprentissage du détecteur final, nous avons décidé de traiter le problème à l'aide d'une méthode offline. Cependant il serait intéressant d'étudier le comportement de notre système couplé à un réapprentissage online. Cela aurait l'avantage de pouvoir mettre à jour le détecteur en cas de changement dans la scène (éclairage, éléments du décor...).

Références

[1] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *Pattern Analysis and Machine Intelligence*, 2004.

[2] M. Black. The Robust Estimation of Multiple Motions : Parametric and Piecewise-Smooth Flow Fields. *Computer Vision and Image Understanding*, 1996.

[3] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, 1998.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Int. Conf. on Computer Vision and Pattern Recognition*, 2005.

[5] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection : An evaluation of the state of the art. *Pattern Analysis and Machine Intelligence*, 2011.

[6] M. Enzweiler and D. M. Gavrila. Monocular pedestrian detection : Survey and experiments. *Pattern Analysis and Machine Intelligence*, 2009.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results. http ://www.pascal-network.org/challenges/VOC/voc2009/workshop/.

[8] Y. Freund and R. Schapire. A short introduction to boosting. *J. Japan. Soc. for Artif. Intel.*, 1999.

[9] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression : a statistical view of boosting. *Annals of Statistics*, 1998.

[10] D. Gerónimo, A. M. López, A. D. Sappa, and T. Graf. Survey of pedestrian detection for advanced driver assistance systems. *Pattern Analysis and Machine Intelligence*, 2010.

[11] H. Grabner and H. Bischof. On-line boosting and vision. In *Int. Conf. on Computer Vision and Pattern Recognition*, 2006.

[12] C. Leistner, A. Saffari, P. M. Roth, and Bischof H. On robustness of on-line boosting - a competitive study. In *Int. Conf. on Computer Vision - Workshop on Online Learning for Computer Vision*, 2009.

[13] A. Levin, P. Viola, and Y. Freund. Unsupervised improvement of visual detectors using co-training. *Int. Conf. on Computer Vision*, 2003.

[14] C. Rosenberg, M. Hebert, and H. Schneiderman. Semi-supervised self-training of object detection models. *IEEE Workshop on Applications of Computer Vision*, 2005.

[15] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 1999.

[16] S. Stalder, H. Grabner, and L. Van Gool. Exploring context to learn scene specific object detectors. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2009.

[17] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Int. Conf. on Computer Vision and Pattern Recognition*, 1999.

[18] V. N. Vapnik. *The nature of statistical learning theory*. 1995.

[19] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Int. Conf. on Computer Vision and Pattern Recognition*, 2001.

[20] B. Wu. Part based object detection, segmentation, and tracking by boosting simple feature based weak classifiers, 2008.