Contribution à la localisation et à la reconnaissance d'objets dans les images

Harzallah Hedi

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Mathématiques et Informatique**

Arrêté ministérial : 7 Août 2006

Présentée par

Hedi HARZALLAH

Thèse dirigée par **Cordelia Schmid**
et codirigée par **Fréderic Jurie**

préparée au sein **du laboratoire Jean Kuntzmann**
et de **l'école doctorale MSTII : Mathématiques, Sciences et Technologies de l'Information, Informatique**

Contribution à la localisation et à la reconnaissance d'objets dans les images

Thèse soutenue publiquement le **16 Septembre 2011**,
devant le jury composé de :

M., Jean Ponce
Professeur, ENS/INRIA, Président
M., Michel Devy
Professeur, LAAS/CNRS, Rapporteur
M., Michel Dhome
Professeur, LASMEA, Rapporteur
M., Pierre Pasquier
Ingénieur, MBDA France, Examinateur
M., Cordelia Schmid
Docteur, INRIA Grenoble, Directeur de thèse
M., Frédéric Jurie
Professeur, Université de Caen, Co-Directeur de thèse

Résumé

Cette thèse s'intéresse au problème de la reconnaissance d'objets dans les images vidéo et plus particulièrement à celui de leur localisation.

Elle a été conduite dans le contexte d'une collaboration scientifique entre l'INRIA Rhône-Alpes et MBDA France. De ce fait, une attention particulière a été accordée à l'applicabilité des approches proposées aux images infra-rouges.

La méthode de localisation proposée repose sur l'utilisation d'une fenêtre glissante incluant une cascade à deux étages qui, malgré sa simplicité, permet d'allier rapidité et précision. Le premier étage est un étage de filtrage rejetant la plupart des faux positifs au moyen d'un classifieur SVM linéaire. Le deuxième étage élimine les fausses détections laissées par le premier étage avec un classifieur SVM non-linéaire plus lent, mais plus performant. Les fenêtres sont représentées par des descripteurs HOG et Bag-of-words.

La seconde contribution de la thèse réside dans une méthode permettant de combiner localisation d'objets et catégorisation d'images. Ceci permet, d'une part, de prendre en compte le contexte de l'image lors de la localisation des objets, et d'autre part de s'appuyer sur la structure géométrique des objets lors de la catégorisation des images. Cette méthode permet d'améliorer les performances pour les deux tâches et produit des détecteurs et classifieurs dont la performance dépasse celle de l'état de l'art.

Finalement, nous nous penchons sur le problème de localisation de catégories d'objets similaires et proposons de décomposer la tâche de localisation d'objets en deux étapes. Une première étape de détection permet de trouver les objets sans déterminer leurs catégories tandis qu'une seconde étape d'identification permet de prédire la catégorie de l'objet. Nous montrons que cela permet de limiter les confusions entre les classes, principal problème observé pour les catégories d'objets visuellement similaires.

La thèse laisse une place importante à la validation expérimentale, conduites sur la base PASCAL VOC ainsi que sur des bases d'images spécifiquement réalisées pour la thèse.

Mots clés
Localisation d'objets • Classification d'images • Reconnaissance d'objet • Apprentissage machine • Apprentissage supervisé • Cascade de classifieurs • Contexte de l'image • Classifieurs non linéaires • Confusion inter-classes

Abstract

This thesis addresses the problem of object recognition in images and more precisely the problem of object localization.

It have been conducted in the context of a scientific collaboration between INRIA Rhônes-Alpes and MBDA France. Therefore, a particular attention was accorded to the applicability of the proposed approaches on infrared images.

The localization method proposed here relies on the sliding windows mechanism combined with a two stage cascade that, despite its simplicity, allies rapidity and precision. The first stage is a filtering stage that rejects most of the false positives using a linear classifier. The second stage prunes the detections of the first classifier using a slower yet efficient non-linear classifier. Windows are represented with HOG and Bag-of-words descriptors.

The second contribution of this thesis is a method that combines object localization and image categorization. This allows, on the one hand, to take into account context information in localization, and on the other hand, to rely on geometrical structure of objects while performing image categorization. This combination leads to a significant quality improvement and obtains performance superior to the state of the art for both tasks.

Finally, we consider the problem of localizing visually similar object categories and suggest to decompose the task of object localization into two steps. The first is a detection step that allows to find objects without determining their category while the second step, an identification step, predicts the objects categories. We show that this approach limits inter-class confusion, which is the main difficulty faced when localizing visually similar object classes.

This thesis accords an important place to experimental validation conducted on PASCAL VOC databases as well as other databases specifically introduced for the thesis.

Keywords
Object localization • Image categorization • Object recognition • Machine learning • Supervised learning • Cascade classifier • Image context • Non-linear classifier • Inter-class confusion

Table des matières

1

Introduction

Sommaire

Omniprésents, les ordinateurs effectuent à longueur de journée des tâches répétitives. Ils aident ainsi l'homme à manipuler d'énormes quantités de données, souvent même plus rapidement et plus précisément que lui.

Malgré cela, la capacité des ordinateurs demeure limitée lorsqu'il s'agit d'extraire automatiquement des informations d'images ou de vidéos, qui représentent pourtant des volumes de données extrêmement importants. Le traitement automatisé de ces données ouvrirait la voie à beaucoup d'opportunités.

La discipline qui vise à automatiser la compréhension des images, *la vision par ordinateur*, est une branche de l'intelligence artificielle dont l'objectif est précisément de permettre à une machine de comprendre ce qu'elle "voit" lorsqu'on la connecte à une ou plusieurs caméras. Nos travaux se situent au cœur cette discipline.

Cette discipline a connu un intérêt grandissant avec la montée en flèche du nombre d'images numériques grâce au développement de la vidéosurveillance et à la démocratisation des appareils photos numériques, des téléphones et des réseaux permettant leur échange (à peu près 5 Milliards de photos sont hébergées sur le site Flickr et 2.5 Milliards nouvelles photos par mois sont ajoutées au site Facebook [70, 71]).

Dans ce vaste domaine qu'est la vision par ordinateur, nous nous intéressons ici à la sous-discipline qu'est la reconnaissance d'objets, et plus particulièrement à la localisation de classes d'objets dans des images (par exemple localiser toutes les personnes présentes dans une image). Ce type de technique combine des approches

de traitement d'images et d'apprentissage automatique dans le but de mener à bien cette tâche de reconnaissance des objets présents dans les images.

L'amélioration des capacités des ordinateurs à effectuer cette tâche offrirait une multitude de possibilités en termes d'applications. Prenons l'exemple d'une collection de photos personnelles. De telles techniques permettraient de classer et de rechercher les photos plus facilement. Étendues à l'échelle du web, elles permettraient d'améliorer la recherche d'images qui pour le moment est principalement basée sur la recherche textuelle. Si nous prenons le cas d'un détecteur de personnes, celui-ci permettrait par exemple d'assister un opérateur dans des tâches de vidéosurveillance. Combiné à la reconnaissance d'actions, il permettrait de détecter des gestes suspects ou encore des malaises dans des maisons de retraite. Il est aussi possible d'imaginer un système de régulation automatique du trafic routier où le feu se mettrait automatiquement au vert si aucune autre voiture ou personne n'est présente au carrefour. Ou encore, des voitures intelligentes assistant le conducteur et l'informant des dangers imminents.

Les efforts consentis commencent à porter leurs fruits et des applications industrielles commencent à voir le jour. On peut citer la nouvelle génération de radars automatiques capables de reconnaitre si un véhicule est un camion ou une voiture pour appliquer la limitation de vitesse correspondante, où encore la reconnaissance et la reconstruction 3D des joueurs utilisés lors de la dernière coupe du monde de football.

Toutes ces applications, et bien d'autres, expliquent l'effort conduit par la communauté de la vision par ordinateur pour répondre à ce problème. Les travaux présentés dans ce manuscrit participent à cet effort.

La suite de ce chapitre présente plus en détails le but visé, les difficultés rencontrées et les contributions apportées par cette thèse.

1.1 But

Cette thèse s'intéresse au problème de la *localisation* (aussi appelée parfois *détection*) d'objets dans les images. En réalité, le terme de *localisation d'objets* peut désigner deux problèmes très différents : la localisation d'*instances* d'objets et la localisation de *catégories* d'objets. La localisation d'instances d'objets consiste à détecter et localiser le même objet dans des images différentes. La localisation de catégories d'objets consiste à détecter des objets différents mais appartenant à la même catégorie (par exemple détecter toutes les chaises présentes dans une image, quelque soit leur type). Nous sommes ici principalement intéressés par la localisation de catégories d'objets.

Notre but est de détecter des objets vus sous des conditions réalistes avec changement de pose, de point de vue, de taille, sur des fonds complexes et pour des caté-

gories ayant une forte variance intra-classe, sans nous restreindre à un type d'objet particulier. Ceci étant dit, nos travaux s'intéressent quand même particulièrement aux objets rigides. Ce choix est lié à la collaboration avec MBDA France dans le cadre duquel s'est déroulée cette thèse. Une autre contrainte qui en découle est que les détecteurs produits doivent être applicables (ou au moins extensibles) aux objets de petite taille ainsi qu'aux images infrarouges.

La position d'un objet dans l'image peut être définie au moyen de plusieurs critères (centre, boite englobante, ensemble des pixels de l'objet). Dans nos travaux, la position d'un objet est déterminée par sa boite englobante. Ceci constitue un compromis raisonnable entre précision et coût de génération des annotations nécessaires pour l'entrainement des détecteurs et pour leur validation.

Dans un deuxième temps, nous nous intéressons aux apports de la combinaison de techniques de localisation d'objets avec des techniques de catégorisation d'images. Ces deux problèmes, bien que proches, ont souvent été traités séparément. Nous voulons montrer ici qu'un réel gain peut être obtenu en les combinant.

Finalement, nous nous penchons sur le cas des classes d'objets très similaires qui, nous le montrerons, posent des problèmes spécifiques. Nous montrons que dans ce cas particulier la définition des classes (répartition des objets en classe et sous-classe) et donc des classifieurs appris, influence grandement les résultats et que cette définition doit être faite avec soin.

1.2 Difficultés

La tâche de localisation d'objets dans les images — et plus généralement la reconnaissance d'objets — présente plusieurs difficultés. La première réside dans la définition même des classes objets et de leurs modèles, qui est subjective et dépend de l'appréciation de la personne, de sa culture ou encore de son expertise. De plus, un objet peut non seulement être défini par son apparence mais aussi par sa fonction ou son interaction avec le monde extérieur. Il convient de noter que des objets ayant une apparence ou une fonction similaire peuvent être de classes différentes. De la même manière, des objets ayant des apparences différentes peuvent appartenir à la même classe, accentuant ainsi le problème de la variance intra-classes, très forte dans les images réalistes.

Dans ce genre d'images, les objets peuvent être partiellement masqués et apparaissent généralement sur des fonds complexes et variés. Un détecteur robuste devra alors être capable de détecter correctement les objets d'une même classe (pouvant pourtant avoir des apparences très diverses), sans pour autant commettre d'erreurs, et ce malgré un fond complexe. Il devra également être capable de ne pas les confon-

Fig. 1.1: Quelques objets problématiques. A droite, la chaise de l'image ne peut être reconnue comme telle que par sa fonction et non par son apparence. A gauche, le bus vert de l'image est peint sur le bus blanc.

Fig. 1.2: Variabilité des apparences des objets de classe "voiture".

dre avec d'autres objets, bien que les objets recherchés puissent être partiellement masqués et vus sous n'importe quelle condition d'illumination.

La difficulté de la localisation d'objets ne réside pas seulement dans ces problèmes à surmonter, mais aussi dans la difficulté d'exploiter certaines informations. On peut citer l'exemple de l'information contextuelle — a priori utile mais comportant une très grande variabilité — ou encore celui l'intégration et l'exploitation des informations liées à la variation des apparences avec les changements de points de vue.

1.3 Contexte

L'un des buts de la vision par ordinateur est de comprendre le contenu des images. Cette compréhension passe en particulier par la reconnaissance des objets contenus dans les images et par la compréhension des relations d'interaction entre les objets et la scène.

Fig. 1.3: Variabilité du fond avec des objets transparents : 3 images de vélos avec des fonds très différents auxquels le détecteur doit s'adapter.

La reconnaissance d'objets a suscité l'intérêt de la communauté scientifique dès les premiers pas de la vision par ordinateur. Nous pouvons citer les travaux fondateurs de Fischler et Elschlager [29] qui ont tenté de formaliser la connaissance d'un objet en le subdivisant en plusieurs parties puis en représentant ces parties par un modèle en constellation intégrant la relation entre les différentes parties et la flexibilité qu'elles peuvent avoir (voir figure 1.4 pour une illustration).

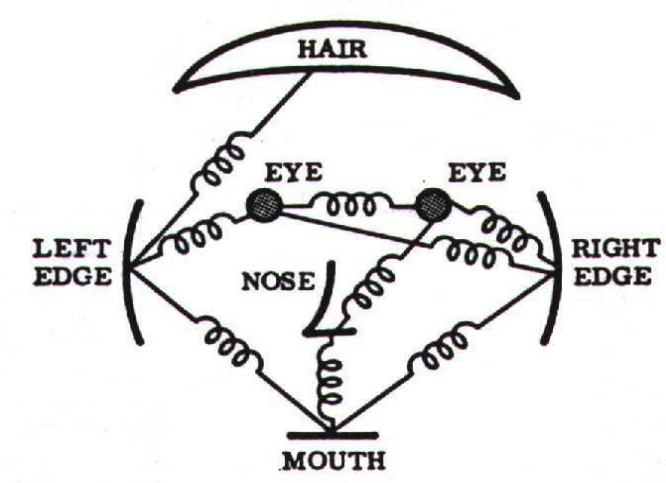


Fig. 1.4: Modèle utilisé par Fischler et Elschlager pour modéliser les visages (figure provenant de [29]).

Or la tâche s'est avérée rapidement difficile et les efforts se sont tournés vers des images plus simples ou encore dans des environnements contrôlés. Ainsi, durant les années 80 et le début des années 90 l'attention s'est portée sur des objets bien centrés, généralement avec des fonds uniformes, tels que les caractères et les visages [75]. L'inconvénient des méthodes résultantes est que les objets étaient souvent représentés par les images elles mêmes (représentation explicite au moyen d'exemples). Les modèles résultants n'étaient donc pas robustes aux transformations et variations rencontrées dans les images telles que les variations d'apparence et de point de vue,

et elles ne pouvaient pas s'étendre à des objets sur des fonds complexes ou partiellement masqués.

Les méthodes par apprentissage ont apporté des réponses à ces problèmes. Elles permettent de produire automatiquement des classifieurs en construisant des règles générales à partir de cas particuliers appelés exemples d'apprentissage. Parmi ces méthodes, nous pouvons citer les réseaux de neurones [4] ou encore les machines à vecteurs support (SVM) [79] particulièrement populaires en raison de leur bonne performance.

D'autre part, les techniques de description d'images ont progressé notamment avec l'apparition des *descripteurs locaux*. Ces derniers représentent une image ou un objet par une collection de vignettes centrées autours de points d'intérêt [67]. Les primitives qui décrivent ces vignettes ont fait l'objet d'un travail considérable qui a permis l'émergence de méthodes devenues aujourd'hui classiques. Les descripteurs locaux reposent généralement sur des histogrammes d'orientation du gradient [14, 56] ou encore des ondelettes de Haar [2, 82].

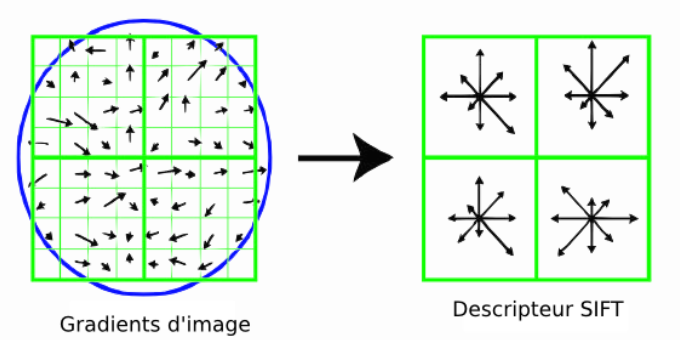


Fig. 1.5: Illustration du principe de calcul du descripteur SIFT, pour une grille 2×2 (figure de [56])

Une façon d'exploiter ces descripteurs locaux est la représentation par *sac de mots* ou *bag of words*, proposée par Csurka *et al.* [12]. Cette approche consiste à représenter l'image par l'histogramme des fréquences des différentes primitives (voir figure 1.6 pour une illustration). Dans ces travaux, Csurka *et al.* mettent en évidence la similarité entre la classification d'image et la classification de documents textuels [40], et proposent une méthode permettant d'effectuer la classification d'images en apprenant un classifieur SVM [79] sur des sacs de primitives.

Ces avancées en classification d'images ont été accompagnées d'avancées des dans les méthodes de localisation. Cela s'explique par plusieurs facteurs, notamment l'augmentation de la puissance de calcul et l'apparition de méthode ayant une faible complexité algorithmique. Ceci est compréhensible étant donné que les méthodes de localisation se basent, pour la majorité d'entre elles, sur la technique des *fenêtres*

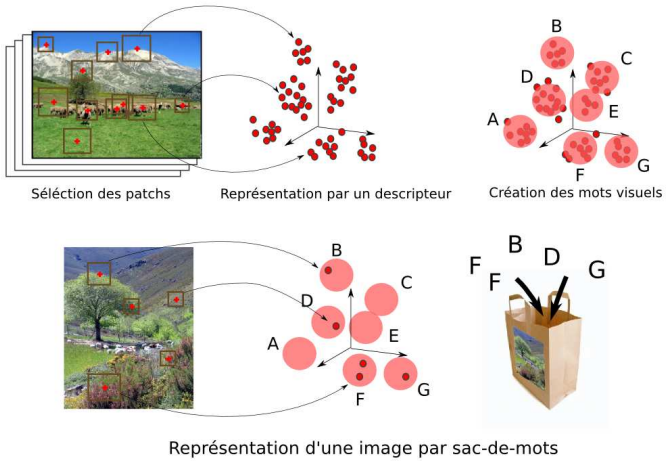


FIG. 1.6: Représentation par sac-de-mots. Un vocabulaire visuel est créé par un apprentissage non supervisé. Ensuite, une image est représentée par le nombre d'occurrences de chacun de ses mots visuels (figure extraite de [47]).

glissantes, particulièrement coûteuses en calculs. En ce sens, un des travaux les plus marquants est le détecteur de visage de Viola et Jones [82] qui a inspiré les détecteurs couramment utilisé dans les appareils photo numériques. Cette méthode se base sur une cascade de classifieurs où un descripteur de fenêtre est évalué successivement par une série de classifieurs qui peuvent choisir de le rejeter ou de continuer les traitements. La cascade, qui permet d'accélérer notablement l'opération de classification, a mis en évidence le potentiel des méthodes de classification par boosting (notamment avec adaboost), qui ont été largement repris par la suite [7, 38, 46, 86]. Une autre technique intéressante proposé par Viola et Jones [82] est celle des *images intégrales*. Celle-ci, introduite pour les primitives de Haar et étendue par la suite [46, 89], permet d'accélérer considérablement le calcul des primitives.

Récemment, Lampert *et al.* [45] ont traité le problème sous un angle différent en essayant de s'affranchir des fenêtres glissantes en se basant sur l'algorithme de *Séparation et Evaluation* (*Branch and Bound*). La méthode proposée permet de trouver le maximum global de la fonction de score tout en examinant un nombre restreint de fenêtres. Ceci dit, l'utilisation de cette technique impose une contrainte sur la fonction de score, et donc sur la nature du classifieur utilisée, et ne permet pas l'utilisation des classifieurs les plus performants.

Une autre série de travaux s'est penchée sur la représentation des objets. Notons les travaux de Dalal et Triggs [14] qui ont proposé un descripteur à base de histogrammes d'orientation des gradients (HOG), proche des SIFT, qui donne de très bonnes performances et a ravivé l'intérêt pour les descripteurs denses non quantifiés. Notons aussi les récents travaux de [83] qui combinent les descripteurs HOG

et LBP [62] et obtiennent de très bonnes performances tout en pouvant s'exécuter en temps réel.

D'autres travaux se sont intéressés à la modélisation des objets. Citons la méthode ingénieuse des *modèles implicites de forme* (*Implicit shape model*) proposée par Leibe *et al.* [49, 50] qui permet d'apprendre la relation entre les mots visuels et les masques de segmentation des objets, qu'il sera capable de retrouver dans une nouvelle image. Plus classique — mais non moins intéressants — sont les travaux de Felzenszwalb *et al.* [24]. Ceux-ci ont su tirer profit des récentes améliorations des techniques d'apprentissage et de représentation pour construire un modèle par partie particulièrement performant. Cette méthode a ravivé l'intérêt pour ce type de modèles qui étaient jusque là plus attractifs conceptuellement que pratiquement.

Ces avancées ont permis l'amélioration des systèmes de localisation et de traiter des objets toujours plus complexes et plus difficiles et ont poussé les chercheurs à explorer d'autres pistes. Les dernières années ont été marquées par la place de plus en plus importante que prend l'analyse de l'information contextuelle. Nous pouvons citer les travaux de [37, 73] qui modélisent le contexte présent dans l'image entière et essayent d'en faire ressortir les positions susceptibles de contenir l'objet. D'autre travaux se sont intéressés à modéliser et exploiter les relations entre les objets [8, 16, 44]. Des études récentes ont analysé empiriquement l'influence des différentes composantes de l'information contextuelle sur la localisation des objets [17].

1.4 Contributions

Comme nous venons de l'expliquer, le but premier de ce travail est la localisation d'objets dans des images réalistes. Un second objectif est d'étudier l'apport de localisation d'objet sur la tâche de catégorisation d'images. Dans ce contexte, les contributions apportées par ce travail sont les suivantes :

– Premièrement, nous proposons une méthode de localisation de classes d'objets efficace, basée sur un classifieur linéaire, inspirée des récents travaux constituant l'état de l'art. Nous effectuons ensuite une étude paramétrique des différents composants sur lesquels repose cette méthode. Cette étude a permis non seulement d'obtenir des détecteurs de bonne qualité, mais aussi une meilleure compréhension du fonctionnement du système et la détermination de ses paramètres les plus influents. Ces travaux sont présentés dans le chapitre 3.

– Notre deuxième contribution s'intéresse au problème de modélisation des objets. Nous proposons une cascade de deux classifieurs où le premier classifieur est celui proposé dans le chapitre 3 et le deuxième est un classifieur non-linéaire plus performant vis à vis des données à traiter. Cette cascade permet d'allier rapidité et

précision, améliorant considérablement la qualité des détections. Nos expérimentations incluent aussi des résultats intéressants, montrant par exemple l'influence de la taille des objets, des exemples tronqués ou encore le nombre d'exemples d'apprentissage. Les résultats obtenues par ce détecteur sur une base d'images fournie par MBDA France ont été publiés dans la conférence MCM-ITP [34] en Juin 2009. Cette étude est présentée dans le chapitre 4.

– La troisième contribution étudie la relation entre la localisation d'objets et la catégorisation d'images, qui ont souvent été considérées comme deux tâches distinctes. Nous proposons une méthode combinant les deux qui améliore les capacités et de l'une et de l'autre. Cette combinaison s'apparente à l'intégration de l'information contextuelle dans un sens et à l'exploitation de la structure géométrique des objets dans l'autre. Ces travaux, présentés dans le chapitre 5, ont été publiés dans la conférence ICCV 2009 [33] et ont été présentés en session orale. Les résultats préliminaires de localisation obtenus par cette méthode ont été soumis au PASCAL VOC challenge 2008 et ont obtenus la meilleure performance sur 11 classes parmi 20 et ont été présentés lors du Workshop PASCAL VOC Challenge [35].

– Notre quatrième contribution s'intéresse au problème de localisation de classes d'objets très similaires. Nous proposons une méthode en deux étapes : (a) une étape de localisation générique (localisation de tous les objets de toutes les catégories) (b) une étape de reconnaissance des objets localisés. Nous avons expérimenté cette méthode sur la base d'images de véhicules fournie par MBDA France. En procédant ainsi, non seulement nous améliorons la qualité des détections, mais nous sommes aussi capables de détecter la présence de la plupart des objets même si ces derniers sont mal identifiés par la suite. Nous montrons aussi que cette méthode peut être appliquée dans le contexte d'identification du modèle des voitures. Ces travaux, présentés dans le chapitre 6.

– Une dernière contribution est l'évaluation détaillée des méthodes proposées sur une grande base d'images fournie par MBDA France. La description de cette base ainsi que les résultats de l'évaluation sont présentés en Annexe (section classée "confidentiel" qui ne sera pas présente dans la version publique du manuscrit).

2

État de l'art et bases de données

Sommaire

Ce chapitre comporte trois sections distinctes. La première propose un état de l'art des méthodes de localisation d'objets dans les images, en se focalisant sur celles directement liées à la problématique de cette thèse. La seconde décrit les bases de données que nous utilisons dans nos travaux pour valider les méthodes proposées. Enfin, la troisième section présente les métriques utilisées pour réaliser les validations expérimentales.

2.1 État de l'art

Nous présentons dans cette section un état de l'art sur la localisation des objets dans les images. Nous commençons par donner une brève présentation des approches

historiques, puis nous nous attardons sur quelques unes des méthodes ayant le plus influencé le sujet. Ensuite, nous nous penchons sur les travaux réalisés pour résoudre deux sous-problèmes de la localisation à savoir l'exploration des images et la suppression des non maximas locaux. Finalement nous présentons des travaux récents qui se sont intéressés à l'utilisation d'informations contextuelles pour la localisation, c'est-à-dire qui utilisent tout ce qui entoure l'objet dans l'image.

2.1.1 Approches de localisation

Modélisation des apparences des classes d'objets

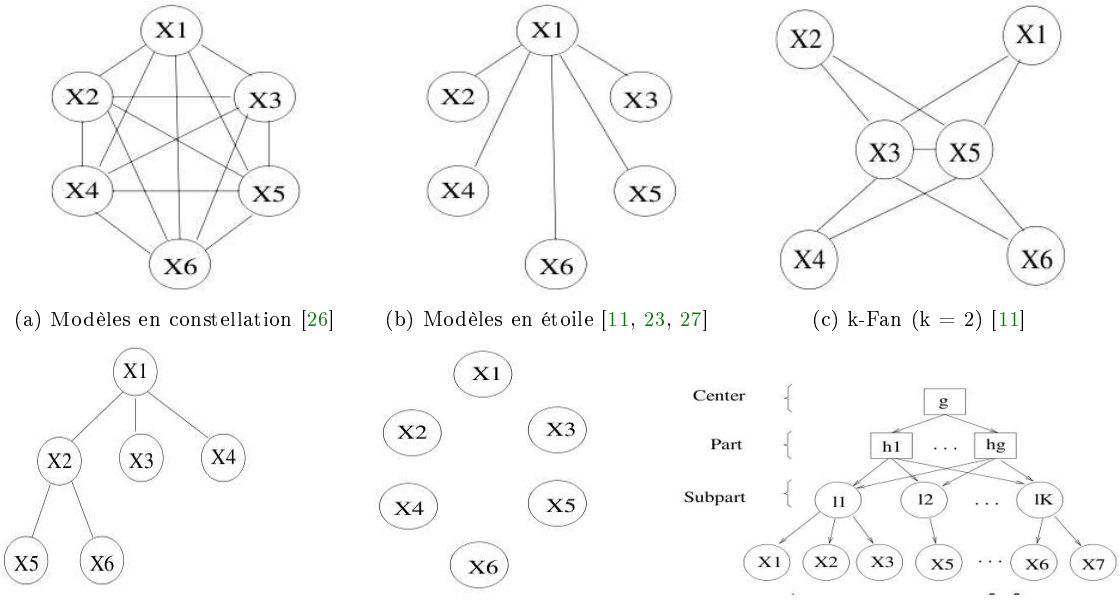
Localiser un objet dans une image n'est possible que si un modèle de ce qui distingue l'apparence de l'objet du fond (et des autres objets) existe et le permet. La modélisation des apparences des objets dans les images est donc au cœur des travaux sur la localisation.

Les modèles par parties furent parmi les premiers modèles proposés. Celui de Fischler et Elschlager [29], introduit en 1973, en est la référence. Ces modèles représentent les objets comme une collection de parties interconnectées. Le type et la force des connections entre ces parties diffèrent d'un modèle à l'autre. Cette connexion peut être formalisée explicitement, par des modèles paramétriques (comme des distributions gaussiennes [26, 59, 84]) ou via des méthodes non paramétriques comme dans [64]. Elle peut être aussi implicite comme dans le cas du modèle implicite de forme [49, 50] que nous détaillons plus loin.

L'organisation des parties et leur nombre ont fait l'objet de nombreux travaux. Fergus *et al.* [26] utilisent les modèles en constellation où toutes les parties des objets sont interconnectées et forment une clique. D'autres approches peuvent être utilisées comme les modèles en étoile [11, 23, 27]. Crandall *et al.* [11] ont proposé quant à eux, un modèle intermédiaire entre les constellations et les étoiles appelé *k-Fan* où la densité des connections est déterminée par le paramètre k (k=1 : modèle en étoile, k=Nombre de partie : modèle en constellation).

Les modèles par sacs de mots peuvent aussi être vues comme des modèles à parties où aucune relation géométrique entre les parties n'est modélisée. Nous montrons dans la figure 2.1 une illustration résumant la différence entre ces modèles.

Même si les modèles utilisant les relations géométriques entre parties paraissent particulièrement adaptés aux objets à parties, comme les humains ou les animaux, ils souffrent de leur de complexité et sont difficile à inférer. De ce fait, jusqu'à récemment, les modèles beaucoup plus simples comme les modèles rigides, obtenaient des performances nettement meilleures que celles obtenus par les modèles par parties.



(a) Modèles en constellation [26] (b) Modèles en étoile [11, 23, 27] (c) k-Fan (k = 2) [11]

(d) Modéles en arbre [25] (e) Modèles par sacs de mots [12] (f) Modèles hiérarchique [5]

Fig. 2.1: Exemples de modèles par parties (figures extraites de [9])

Un autre type de modèles particulièrement populaire dans le passé, surtout dans le contexte de la localisation d'instances d'objets, est celui du *template matching* [61]. Son principe est très simple : il consiste en la mise en correspondance d'un modèle de l'objet, qui n'est rien d'autre qu'une image de l'objet lui même, par convolution du modèle avec l'image. Même si des primitives différentes peuvent être utilisées pour s'affranchir de la représentation par pixel des images et de ses inconvénients, les modèles par template matching souffrent de leur sensibilité à la variance intra-classe et aux changements de points de vue. Pour palier à ces difficultés, des méthodes de *deformable template matching* ont été introduites [3, 39, 87]. Ces méthodes permettent la mise en correspondance de point extraits de l'objet d'intérêt avec ceux extraits de l'image à analyser. Même si ces méthodes arrivent à mieux modéliser les objets, elles ne peuvent que difficilement se généraliser à des objets complexes avec une forte variance intra-classe.

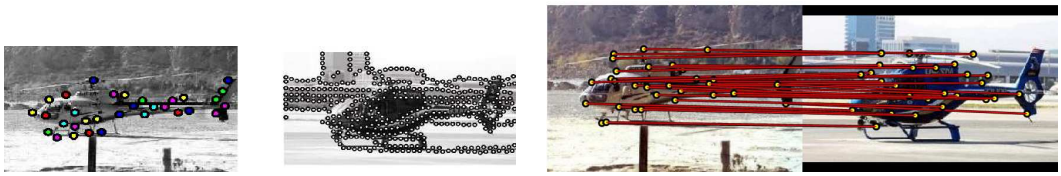


Fig. 2.2: Deformable template matching (figure extraite de [3])

Dans l'histoire récente de la localisation d'objets dans les images, une des méthodes qui a influencé le plus les travaux de recherche est celle de Viola et Jones [81]. Cette méthode repose sur plusieurs principes. Premièrement celui des classifieurs en cascade qui permet de classifier rapidement un nombre important de fenêtres et s'adapte donc parfaitement aux fenêtres glissantes. Nous y reviendrons dans le paragraphe dédié à l'exploration de l'image. La deuxième technique introduite par les travaux de Viola et Jones [81] est celle des images intégrales. Celles-ci permettent de calculer efficacement les primitives, quelles que soient leurs natures du moment où un calcul de sommes de valeurs contenues dans un rectangle est impliqué. Pour ce faire, une image intégrale II est calculée en partant de l'image originale I. La valeur de cette image intégrale en chaque point est la somme des valeurs des pixels contenus dans le rectangle définit par ce point et l'origine de l'image. Une fois l'image intégrale calculée, la somme des pixels contenus dans un rectangle est obtenue en seulement 4 opérations. Par exemple, la somme des valeurs des pixels contenus dans le rectangle D (voir figure 2.3) est calculée ainsi :

$$Sum(I(D)) = II(4) - II(3) - II(2) + II(1). \tag{2.1}$$

L'utilisation des images intégrales a été étendue au calcul d'autres descripteurs comme les HOG [46, 89], sous le nom d'histogrammes intégraux. Notons que l'utilisation des images intégrales perd son attractivité si le nombre d'images intégrales nécessaire devient trop important (par exemple lors d'une représentation par sac de mots avec une taille de vocabulaire de l'ordre de quelques milliers).

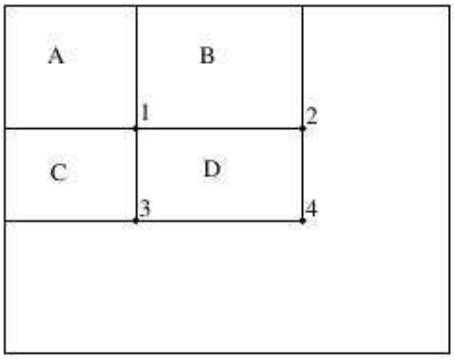


Fig. 2.3: Fonctionnement des Image intégrales : 4 opérations seulement sont nécessaires pour calculer la somme des valeurs des pixels contenus dans le rectangle D : $II(4) - II(3) - II(2) + II(1)$

L'algorithme d'apprentissage utilisé par Viola et Jones [81] (Adaboost [30]) permet aussi la sélection de primitives en choisissant les positions susceptibles de contenir le plus d'informations pertinentes. Le modèle ainsi obtenu est un modèle rigide où les positions des différentes primitives à extraire sont prédéterminées. Ce type

de modèles a été utilisé dans un bon nombre de travaux récents. Par exemple, Laptev [46], tout comme Viola et Jones [81], utilise un algorithme de sélection de primitives pour sélectionner les régions les plus discriminantes pour un objet donné. D'autres travaux [14, 28, 80] découpent la fenêtre en suivant une grille régulière pour obtenir plusieurs carreaux (appelés *blocs* ou *tiles*). Lazebnik *et al.* [48] quant a eux, découpent la fenêtre de description en une pyramide spatiale et utilisent par la suite une métrique adaptée à cette représentation qui donne plus de poids aux cellules les plus fines (voir figure 2.4 pour une illustration).

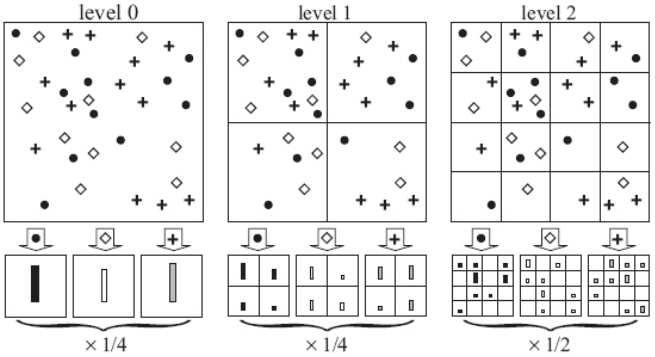
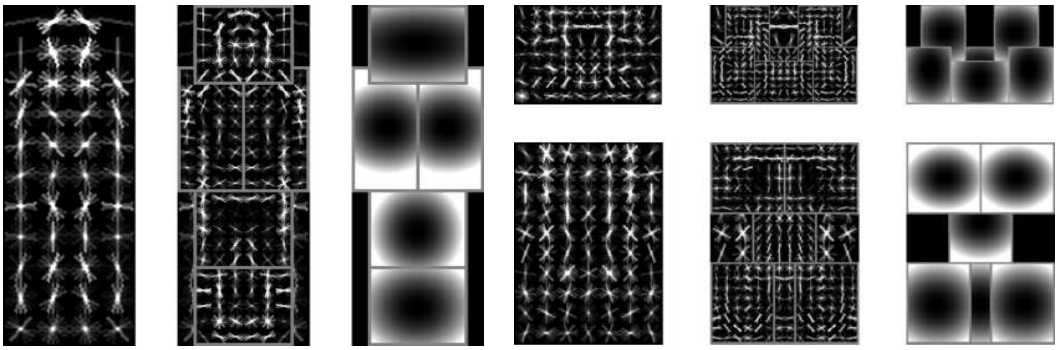


Fig. 2.4: Découpage par pyramide spatiale (figure de [48]).

Parmi les travaux récent basés sur la modélisation des objets en les subdivisant au moyen d'une une grille régulière et rigide, l'un des plus marquants a été celui de Dalal et Triggs [14]. Dans ces travaux, les auteurs proposent un nouvel algorithme de description d'images appelé HOG (Histogram of Oriented Gradients), largement inspiré des SIFT [56]. Les HOGs ne sont rien d'autre que des histogrammes grossiers d'orientations des gradients où 9 orientations différentes sont considérées. L'image de l'objet est découpée en carreaux en chevauchement et chacun est décrit par un HOG. La représentation finale est alors obtenue par concaténation de ces descripteurs. La méthode d'apprentissage utilisée par Dalal et Triggs est celle des machines à vecteurs support (SVM) [68, 78]. Pour renforcer le modèle appris, les auteurs apprennent un premier classifieur qui sera testé sur les images d'apprentissage. Les faux positifs aux plus hauts scores sont alors injectés dans l'ensemble des exemples négatifs qui seront utilisés pour apprendre le classifieur final. Cette méthode a permis d'obtenir d'excellents résultats qui ont inspiré un grand nombre de travaux.

Les travaux récents de Felzenszwalb *et al.* [23, 24] ont tiré partie de la qualité des descripteurs HOG d'une part et de l'avancée des méthodes d'apprentissage machine d'autre part pour construire un détecteur par partie particulièrement performant. Cette méthode se base sur un modèle en étoile où le centre est constitué d'un descripteur HOG grossier couvrant la totalité de l'objet et où les parties sont décrites par des HOGs plus précis (voir figure 2.5 pour une illustration). La position ainsi que

la variabilité de la position des différentes parties de l'objet sont apprises automatiquement. Pour ce faire, les auteurs proposent les *latent SVMs*, une généralisation des SVMs permettant de traiter des vecteurs à variables latentes. Les auteurs proposent aussi une approche pour chercher efficacement les exemples difficiles avec lesquels l'algorithme sera ré-entrainé. Cette méthode a suscité un regain d'intérêt pour les modèles par parties grâce aux excellents résultats obtenus [19, 20, 21]. Ces travaux confirment le bien-fondé de ces modèles et prouvent leurs supériorités lors de la détection d'objets flexibles, tels que les personnes.



(a) Modèle de personne

(b) Modèle de vélo vue de profil et vue de devant.

Fig. 2.5: Modèles de personnes et de vélos obtenus par la méthode de [24] (figure de [24])

Exploration des images

Une des difficultés majeures rencontrées lors de la localisation est que l'objet peut être présent à n'importe quelle position et échelle. Par conséquent, si une technique d'exploration basée sur des fenêtres glissantes est utilisée, le nombre de fenêtres à évaluer est énorme.

Comme nous l'avons mentionné plus haut, la méthode très populaire de Viola et Jones [81] a introduit le principe des classifieurs en cascade qui permet de classer rapidement un nombre important de fenêtres et convient donc bien aux fenêtres glissantes. Pour ce faire, les auteurs décomposent un classifieur fort en une série de classifieurs faibles, très rapides, et les arrangent en cascade. A chaque étage de cette cascade, un classifieur faible décide si la fenêtre contient ou non l'objet. Si oui, la fenêtre sera transmise au classifieur suivant dans la cascade, sinon elle sera considérée comme un négatif et les calculs sont arrêtés. La mesure la plus importante lors de l'apprentissage d'un classifieur en cascade est le "taux de faux négatifs" des étages de la cascade : la quasi totalité des instances d'objets doit être gardée alors

qu'un maximum de "vrai négatifs" doivent être rejetés dès les premiers étages de la cascade.

Au vu de l'économie de temps de calcul qu'offrent les cascades, celles-ci ont reçues beaucoup d'attention durant les cinq dernières années [6, 7, 18, 46, 89]. Ceci étant dit, les cascades ont de multiples limitations. L'entrainement d'une cascade est lent — il peut prendre des semaines entières [81] —, la détermination des bons taux de faux positifs et de détection à chaque étage est souvent empirique, et finalement, l'utilisation des cascades réduit forcement les performances globales du système.

Même si la plus part des travaux ne remettent pas en cause les fenêtres glissantes mais cherchent à s'y adapter, certains on essayé de proposer des alternatives.

Le modèle implicite de forme (ou *Implicit Shape Model*) introduit par Leibe *et al.* [49, 50], constitue une de ces alternatives. Ce modèle apprend pour chaque mot visuel présent sur un objet, sa position relativement au centre de l'objet, ainsi que le masque de segmentation. Ensuite, pour chaque image à traiter, un détecteur de points d'intérêt est utilisé pour extraire des mots visuels, et chaque mot visuel vote dans un espace de Hough pour les positions possibles de l'objet. L'exploration systématique de l'image est donc évitée. Plus récemment, cette méthode a été adaptée par [10, 80] pour générer des fenêtres susceptibles de contenir des objets, et ainsi remplacer là encore la technique des fenêtres glissantes. Comme le nombre de fenêtres obtenues est assez réduit, les auteurs de [10, 80] utilisent des algorithmes d'apprentissage plus sophistiqués, plus gourmands en temps de calcul mais aux capacités de modélisation nettement meilleurs. L'avantage de cette approche est qu'elle permet également de s'affranchir des fenêtres glissantes, en proposant des emplacements possibles d'une manière plus intelligente, et donc de concentrer toute la puissance de calcul disponible sur le traitement de fenêtres pertinentes. L'inconvénient est qu'elle reste dépendante de la qualité des points d'intérêt et des mots visuels extraits, et donc hérite de leurs limitations.

Une autre alternative appelée *Efficient Subwindow Search (ESS)* a été introduite par [45]. Cette technique permet d'explorer et de localiser la fenêtre ayant le plus haut score en utilisant une recherche par "Séparation et évaluation" (Branch and Bound). Cependant cette technique, qui a été introduite initialement pour un classifieur linéaire sur des sacs de mots sans normalisation de descripteurs, n'est applicable que dans certains cas et pour certaines fonctions de score, même si elle a été étendue ultérieurement à d'autres fonctions plus sophistiquées.

Notons que pour accélérer encore plus la procédure de localisation, les auteurs utilisent une image intégrale pour calculer la réponse du classifieur linéaire. Ainsi, le calcul du score d'une fenêtre se fait en 4 opérations élémentaires. Cette technique a été reprise par [80] pour constituer le premier niveau d'une cascade de classifieurs.

Suppression des non maximas

Comme la fonction de score est robuste aux légères translations et changements d'échelle plusieurs fenêtres à fort score sont généralement détectées autours d'un objet. En effet, même si une fenêtre est légèrement décentrée, plus grande ou plus petite que celle de l'objet présent, elle obtiendra tout de même un bon score. Cette même fenêtre sera considérée comme un faux positif lors de l'évaluation, car une et une seule fenêtre par objet est acceptée. Pour remédier à ce problème, un post-traitement appelé *suppression des non maxima* (ou *Non maxima suppression*) est appliqué pour sélectionner les meilleurs fenêtres parmi celles retournées.

La façon la plus triviale d'effectuer cette tâche est de fusionner les fenêtres qui se chevauchent, comme suggéré par [81]. Premièrement, les détections sont partitionnées en des sous-ensembles disjoints où deux détections sont réunies dans le même sous-ensemble seulement s'il y a un chevauchement entre elles. De chaque sous-ensemble, seule la fenêtre moyenne est gardée. Dalal *et al.* . [15] ont posé le problème comme celui d'une estimation de densité non paramétrique. Ils détectent les maxima de densité en appliquant un algorithme de *mean shift* sur 3 dimensions (position et échelle). Si cette densité dépasse un certain seuil, les maximas locaux sont retenus comme des détections.

Leibe *et al.* . [51] déterminent aussi les pics des scores en utilisant un estimateur mean-shift, mais ayant remarqué que plus la taille de l'objet augmente, plus les votes son éparpillés dans l'espace, ils suggèrent d'adapter la largeur du noyau à l'échelle traitée.

Dans le même esprit, Laptev [46] supprime les détections multiples en appliquant un algorithme de clustering sur les fenêtres et utilise la taille des clusters résultants comme mesure de confiance pour les détections.

Fritz *et al.* . [31] ont proposé quant à eux une méthode simple mais efficace pour la suppression des détections multiples. Ils acceptent la détection ayant le plus haut score et rejettent toutes celles qui se chevauche avec cette dernière (indice de Jaccard supérieur à 0.3) et ayant un score plus bas.

Plus récemment, Desai *et al.* . [16] ont suggéré d'incorporer la suppression des non maximas dans un modèle plus sophistiqué qui permet de prendre en compte, entre autre, la cooccurrence des objets et l'exclusion mutuelle entre les objets et ont montré que cette stratégie permet d'améliorer considérablement la qualité des détections. Cette dernière méthode s'inscrit dans une ligne de travaux portant sur l'utilisation de l'information contextuelle dans la reconnaissance d'objets, ligne qui fait l'objet de la section suivante.

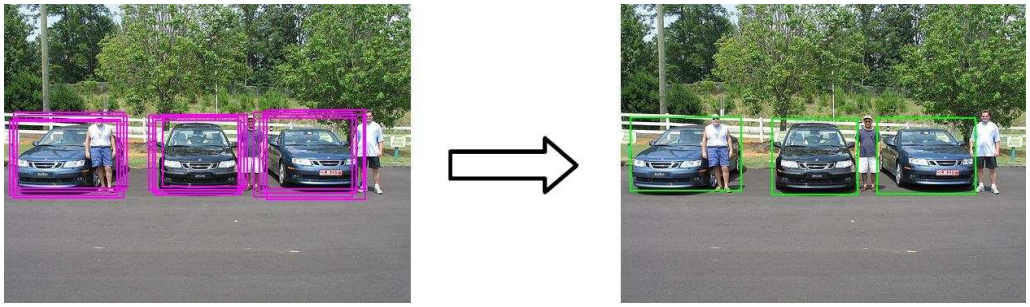


Fig. 2.6: Exemple de résultats avant et après la suppression des non maximas pour la détection de voitures.

2.1.2 Exploitation des données contextuelles

Une bonne compréhension du contenu des images nécessite l'exploitation de tous les éléments (c'est-à-dire l'objet recherché, mais aussi son contexte) dans la prise de décision. Par exemple, il serait surprenant de détecter une vache dans un arbre. Ceci étant dit, le contexte présente généralement une grande variabilité et est donc difficile à exploiter car souvent plus difficile à modéliser que les objets eux-mêmes.

La difficulté du sujet et les opportunités qu'il peut offrir, en fait un sujet de recherche d'actualité assez bien étudié ces dernières années. Aussi nous pouvons mentionner un bon nombre de travaux récents qui se sont penchés sur cette question.

Les travaux de Torralba [73] modélisent la relation entre le contexte et les propriétés de l'objet en se basant sur la corrélation entre les primitives de bas niveau sur toute la scène et les objets qu'elle contient. Cette modélisation permet de se focaliser prioritairement sur les endroits susceptibles de contenir l'objet. Hoiem *et al.* [37] modélisent aussi le contexte présent dans l'image en partant du même principe – c'est-à-dire que toutes les positions n'ont pas la même probabilité de contenir un objet – et construisent un modèle qui intègre l'interdépendance des objets, l'orientation des surfaces ainsi que le point de vue de l'appareil photo dans la détermination des endroits susceptibles de contenir les objets recherchés.

Une autre série de travaux se focalise sur la modélisation des relations inter objets. Carbonetto *et al.* [8] apprennent les relations spatiales entre les objets en se basant sur les labels textuels des images et ce dans un contexte de segmentation d'images. Kumar *et al.* [44] quand à eux proposent un modèle à 2 niveaux où le premier niveau capture les interactions entre l'objet et son proche voisinage et le deuxième les interactions entre régions plus éloignées. Notons aussi les travaux de [36, 85] qui consistent à segmenter d'abord l'image en régions labellisées puis à utiliser ces labels pour la classification des objets voisins.

Enfin, d'autres travaux ont essayé de combiner les résultats obtenus par plusieurs tâches de reconnaissance d'objets et notamment la catégorisation d'images et la

localisation d'objets. Citons les travaux de Li et Fei-Fei [52] qui proposent un modèle graphique d'évènements dans les images où un *événement* est un facteur latent conditionnant la génération d'objets de scènes et de catégories. Shotton *et al.* [69] proposent quand à eux une idée similaire dans le contexte de la segmentation d'images où les catégories les plus probables sont renforcées par la multiplication des distributions de classification de l'image globale et celle de la segmentation locale. S'interrogeant sur la pertinence des diverses sources d'information contextuelles, Divvala *et al.* [17] ont présenté récemment une évaluation empirique du rôle du contexte dans la localisation d'objets et ont évalué plusieurs de ces sources.

Malgré ces travaux, le problème reste difficile et peu de travaux sont arrivés à montrer une amélioration significative des performances due à l'utilisation d'informations contextuelles, sur des bases de données réalistes.

2.1.3 Localisation multi-vues / Localisation multi-classes

La question qui se pose lors de l'apprentissage est celle de savoir comment aboutir à des détecteurs performants malgré la variabilité intra-classe et la similarité inter-classes. Un tel classifieur s'avère difficile à créer, et l'approche la plus communément utilisée est de décomposer le problème en différents sous-problèmes plus faciles à résoudre. La plupart des méthodes de localisation décomposent les classes selon le point de vue et apprennent une série de détecteurs chacun spécialisés dans une vue (ou un groupe de vues) d'une classe d'objet et fonctionnant indépendamment [10, 33, 50, 80, 82]. Plusieurs auteurs ont essayé de proposer de meilleures solutions pour répondre au problème de détection multi-vues [22, 72, 74]. Ce problème reste d'actualité, surtout dans le contexte de détection de visages, où il est très important de détecter aussi bien les visages frontaux que les visages semi-frontaux ou latéraux. La plupart de ces approches se basent sur la stratégie "diviser pour régner" implémentée avec des détecteurs structurés en arbre. L'approche de Jones et Viola [41] consiste en la division de l'espace des poses en plusieurs sous-espaces de poses et l'entrainement d'un détecteur différent pour chacune de ces poses. Une étape initiale estime la pose du visage et l'oriente au détecteur correspondant. Li et Zhang [53] adaptent la cascade standard [82] en suivant une stratégie incrémentalement précise et complexe. Ils utilisent une cascade à 3 niveaux avec des partitions d'orientation différentes (de 1, 3 et 7 portions) appliquées successivement tant que la fenêtre n'est pas rejetée. Une stratégie similaire a été utilisée par Gavrila dans le contexte de détection de piétons [32]. Huang *et al.* [38] proposent aussi un détecteur de visage organisé en arbre. Une stratégie de description incrémentalement fine est appliquée pour diviser l'espace des visages en plusieurs sous-espaces. Chacun des nœuds de l'arbre est un classifieur multi-classes et multi-labels qui peut rejeter la fenêtre ou l'orienter vers un ou plusieurs nœuds suivants. Une des limitations de ce travail est que les sous-catégories doivent être définies par un opérateur. Plus récemment, Wu *et al.* [86] ont

proposé d'utiliser une méthode appelée *Cluster Boosted Tree method* pour constituer automatiquement avec un apprentissage par boosting de l'arbre de classifieurs.

D'autres auteurs se sont intéressés au problème de détection multi-vues dans un contexte plus général que celui de la détection de visage. Thomas *et al.* [72] ont proposé une méthode de détection d'objets générique avec des prises de vues quelconques. Cette approche est basée sur le modèle implicite de forme (ISM) [50]. L'idée principale est de connecter les mots visuels de chaque vue impliquant un transfert de votes entre les vues. Cependant, si plusieurs classes sont présentes, elles doivent être traitées séparément. L'aspect multi-classes du problème est donc ignoré.

Une autre possibilité de décomposition du problème serait, au lieu de partitionner les classes et apprendre un détecteur par vue, de redéfinir les classes en fusionnant les classes similaires. Cette procédure permettrais d'augmenter le nombre d'images d'apprentissage et de ce concentrer sur les différences entre ces classes et le fond. Plus généralement, on peut penser que le partage des indices entre les différentes classes peut aboutir à une amélioration notable de la qualité des détecteurs. Parmi les rares travaux qui se sont intéressés à ce problème, on peut citer les travaux de Torralba *et al.* [74] ainsi que ceux de Opelt et Pinz [63]. Même si ces approches utilisent des détecteurs distincts par classe, les primitives sont partagées entre les différentes classes permettant une détection plus rapide et plus précise.

2.2 Bases de données

Nous présentons dans cette section les bases d'images utilisées pour effectuer nos expérimentations. Nous utilisons premièrement les bases PASCAL VOC qui représentent une référence dans le domaine. Nous utilisons aussi une base d'images fournies par MBDA France dans le cadre de la collaboration entre l'INRIA et MBDA France. Finalement, nous présentons deux bases de véhicules (dites *base de voitures* et *base de voitures de PASCAL*) que nous avons annotées et que nous utilisons dans le chapitre 6.

2.2.1 Les bases PASCAL VOC

Les bases d'images PASCAL ont été crées et mises à disposition publiquement dans le contexte d'organisation de compétitions internationales, pour comparer les différentes méthodes de l'état de l'art. Elles sont à ce jour les bases d'images les plus répandues et constituent la référence dans les domaines de catégorisation d'images et de localisation d'objets. La compétition en question est le "PASCAL Visual Ob-

Classe	plane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow
Apprentissage	238	243	330	181	244	186	713	337	445	141
Test	112	116	180	81	139	97	376	163	224	69
Classe	table	dog	horse	moto	person	plant	sheep	sofa	train	tv
Apprentissage	200	421	287	245	2008	245	96	229	261	256
Test	97	203	139	120	1025	133	48	111	127	128

Tab. 2.1: Nombre d'images d'apprentissage et de test par classe pour la base PASCAL VOC.

ject Classes Challenge" (ou PASCAL VOC Challenge) qui est organisé chaque année depuis 2005 [1].

La base PASCAL VOC 2007 [19] est la plus récente des bases PASCAL dont les annotations sont disponibles. C'est donc cette base que nous utilisons ici. Elle contient les 20 catégories d'objets suivantes :
– Personnes.
– Animaux (bird, cat, cow, dog, horse and sheep).
– Véhicules (aeroplane, bicycle, boat, but, car, motorbike and train).
– Objets d'intérieur (bottle, chair, dining table, potted plant, sofa and TV/monitor).
Elle est constituée de 9963 images réparties en 5011 images d'apprentissage et 4952 images de test (voir le tableau 2.1 pour la distribution du nombre d'images par classe d'objets). Ces images ont été collectées sur internet, principalement à partir de Flickr, et annotées suivant des directives bien définies. Les objets sont présents dans des conditions réalistes avec des changements d'échelles, de points de vue ou encore d'illumination avec à chaque fois un nombre important de distracteurs issus du fond et des objets avoisinants (voir figure 2.7 pour un aperçu des images.) Pour chaque instance d'objet, une annotation est fournie avec la boite englobante l'objet (bounding box), la pose de l'objet (droite, gauche, avant, arrière ou autre) et finalement un indicateur permettant de savoir si l'objet est tronqué (partiellement visible). Quelques objets sont considérés comme "difficiles" et explicitement marqués comme tels. Ces objets ne sont pas pris en compte lors de l'évaluation des performances (selon le règlement du PASCAL VOC Challenge). Notons que nous utilisons aussi la base PASCAL VOC 2008 afin de comparer les approches développés aux approches les plus récentes de l'état de l'art.

2.2.2 Base MBDA

La base MBDA est une base de véhicules composée de 76000 images infrarouges séparées à parts égales entre images de test et d'apprentissage. Il y a 20 classes d'objets : 5 classes de véhicules légers, 3 classes de camions et 12 classes de poids

[1] http ://pascallin.ecs.soton.ac.uk/challenges/VOC/

Fig. 2.7: Exemples d'images de la base PASCAL VOC 2007.

lourds. Chaque objet est présent dans un nombre égal d'image de test et d'apprentissage. Les objets sont présents à des positions, des échelles et des points de vue différents. Les points de vue sont regroupés en 4 classes : gauche, droite, avant et arrière. Les images sont générées synthétiquement mais respectent des conditions réalistes avec fonds diverses et plusieurs conditions météorologiques. La figure 2.8 présente des exemples d'images utilisées pour l'apprentissage pour les 3 classes de camions présentes. Des exemples d'images de test sont montrés dans la figure 2.9. Cette base étant confidentielle, une présentation plus détaillée ainsi que les résultats de localisation obtenus avec les différentes méthodes développées lors de nos travaux sont donnés dans l'annexe A (confidentiel).

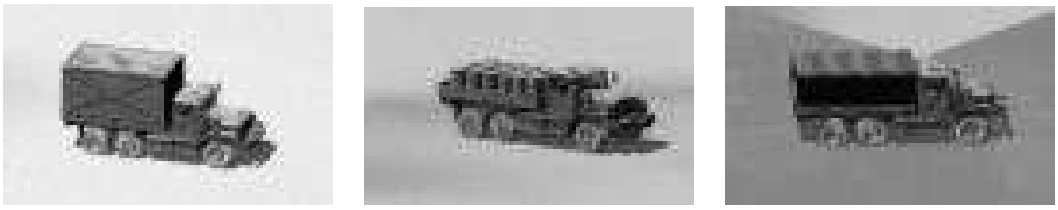


Fig. 2.8: Exemples d'images de camions de la base MBDA.

Fig. 2.9: Exemples d'images de test de la base MBDA.

2.2.3 Bases de véhicules

Pour évaluer la classification hiérarchique présentée dans le chapitre 6, nous avons crée les deux bases de données suivantes :

Base de voitures

La base de voitures est composée de 6000 images : 1000 images de voitures et 5000 de fond (ne contenant pas de voitures). Les images de voitures sont celles de 8 modèles différents, à savoir 106, 206, 407, AX, Clio, Logan, Punto et Qashqai. Ces images ont été collectées sur internet : le nom du modèle est recherché sur *Google images* pour obtenir une première liste d'images que nous filtrons manuellement par la suite. Pour ce qui des images de fond, nous utilisons des images de la base PASCAL VOC 2007 ne contenant pas de voitures. Chaque annotation renseigne une boite englobante, le point de vue et le modèle de la voiture. Cinq points de vue sont considérés (gauche, droite, avant, arrière, et "non spécifié"). Les voitures tronquées ainsi que celles de modèle inconnu sont marquées comme difficiles et ignorées lors de l'évaluation, similairement à la procédure suivi au PASCAL VOC challenge. La figure 2.10 montre des exemples d'images de la base de voitures.

Fig. 2.10: Exemples d'images de la base de voitures

Base des voitures de PASCAL

La base des voitures de PASCAL est une base crée à partir des images de voitures présentes dans la base PASCAL VOC 2007 que nous classifions en 3 catégories : "Voitures de sport", "Voitures 4x4" (contenant aussi les utilitaires) et "Autre" (voir figure 2.11 pour une illustration). Ceci résulte en respectivement 270, 186 et 794 objets d'apprentissage et 308, 239 et 654 objets de test. Nous gardons les informations fournies par les annotations initiales à savoir les marqueurs "Tronqué" et "Difficile" ainsi que l'information de point de vue. Notons que la définition des classes ici diffère de celle de PASCAL. Elle désigne les catégories "Voitures de sport", "Voitures 4x4" et "Voiture Autre" et non la catégorie "voiture".

2.3 Critères d'évaluation

Nous décrivons ici les mesures d'évaluation de performances utilisées lors de nos différentes expérimentations.

Sport 4x4 Autre

Fig. 2.11: Exemples d'images de la base des voitures de PASCAL

2.3.1 Critère d'appariement

Nous commençons par décrire le critère définissant une bonne détection, à savoir l'indice de Jaccard. Cette mesure permettant à l'origine de mesurer la différence entre deux ensembles, est utilisée ici pour mesurer le chevauchement entre d'une part la fenêtre détectée, et d'autre part la vérité terrain. Pour qu'une détection soit considérée correcte, il faut que l'indice de Jaccard entre la boite englobante prédite Bp et celle de la vérité terrain Bgt excède 50%. Cet indice est calculé comme suit :

$$Indice_{Jaccard} = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})}. \tag{2.2}$$

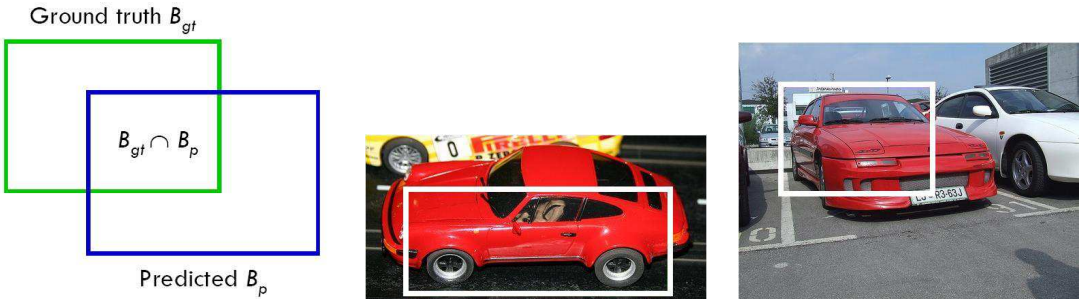


Fig. 2.12: A gauche, un schéma explicatif du calcul de l'index de Jaccard, au centre, une détection correcte avec un $Indice_{Jaccard} = 55\%$, et à droite une fausse détection avec un $Indice_{Jaccard} = 49\%$

Bien que considéré comme excessivement sévère par certains, ce critère est le plus répandu dans la littérature et c'est justement celui utilisé par le PASCAL VOC challenge.

2.3.2 Courbe de précision/rappel

La qualité de la localisation est généralement évaluée dans nos travaux par la courbe donnant la précision en fonction du rappel. Cette courbe permet de mettre en évidence deux aspects de la détection :

- le taux de précision ou la proportion de vrais positif dans les fenêtres retrouvées.
- le taux de rappel ou la proportion d'objets trouvés parmi tous les objets pertinents présents dans la base de données.

2.3.3 Courbe du rappel/taux de fausses alarmes

Une autre courbe utilisée pour l'évaluation de la qualité de la localisation est celle du rappel en fonction du taux de fausses alarmes. Cette courbe est souvent jugée plus intuitive et plus facile à lire et à interpréter. Comme pour la courbe précision/rappel, on fait varier le seuil de détection du système pour tracer le rappel en fonction du taux de fausses alarmes. Le taux de fausses alarmes peut être exprimé en nombre de fausses alarmes par images, ce que nous faisons dans ce manuscript.

2.3.4 Average Precision

Pour pouvoir statuer clairement si une méthode est meilleure qu'une autre et de combien, et ce surtout dans le contexte d'organisation de compétitions, il est nécessaire de pouvoir représenter les performances d'une méthode par une valeur numérique unique. Tous le long de ce manuscrit, nous suivons le critère du PASCAL VOC challenge et utilisons la mesure de *Average Precision* (ou AP). L'AP est une approximation de l'aire sous la courbe de précision/rappel. Pour obtenir l'AP, on calcule d'abord la précision interpolée en 11 valeurs de rappel différentes à savoir $r \in \{0, 0.1, ..., 0.9, 1\}$. La précision interpolée pour un niveau de rappel r donné est définie comme la précision maximale pour une valeur de rappel supérieur ou égale à r. Ce calcul est justifié par le fait qu'une augmentation de rappel est toujours la bienvenue si une augmentation de la précision s'en suit. L'AP est alors simplement la moyenne arithmétique de ces précisions interpolées. Ainsi, une forte valeur de l'AP requiert des valeurs de précision fortes pour tous les taux de rappel (voir fig 2.13).

Mean Average Precision

La *Mean Average Precision* (ou mAP) est la moyenne arithmétique des AP. Elle est utilisée généralement pour comparer les performances de deux méthodes sur un ensemble de classes.

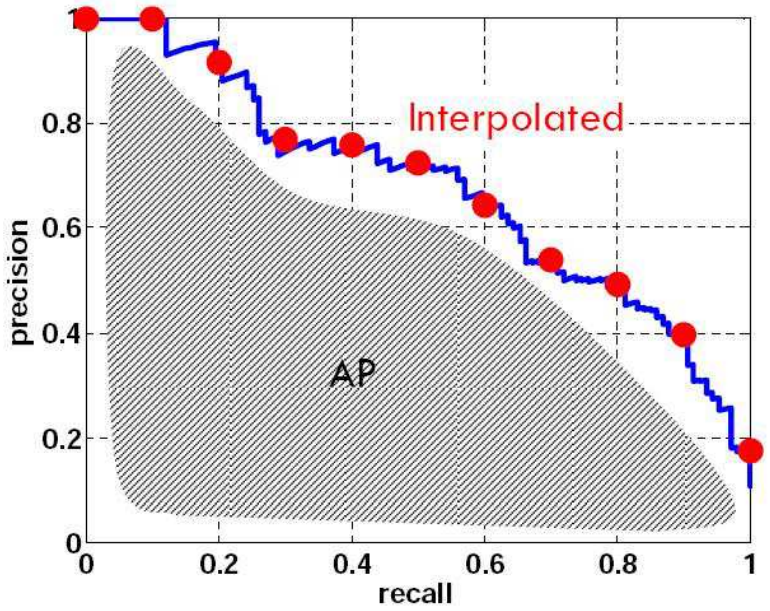


Fig. 2.13: Calcul de l'Average Precision.

2.3.5 Tests statistiques

Test de Friedman

Dans le cas où un nombre important d'expérimentations est effectué (comme dans le cas de notre étude paramétrique), la présentation exhaustive de tous les résultats devient déroutante et n'aide pas à tirer des conclusions. Pour présenter les résultats d'une manière plus synthétique et faire ressortir les paramètres importants, on utilise le *test de Friedman*. C'est un test statistique qui, en se basant sur les rangs, permet de dire si deux méthodes ou configurations sont significativement différentes pour un niveau de confiance donné. Dans notre cas, ce test se base sur les rangs obtenus par les différentes configurations sur les différentes classes et sépare ces configurations en des ensembles de configurations significativement différentes avec une certitude de 95%.

T-test

Dans le cas où les résultats présentent une certaine variance, il est difficile de mesurer précisément la différence de performances entre deux méthodes ou configurations. Pour remédier à ce problème nous utilisons ici un test statistique qui est le *t-test*. Ce test permet de mesurer la différence de performance entre deux méthodes, avec une certitude donnée, en utilisant les résultats obtenus avec différents runs. Dans notre cas, la différence de performance est montrée avec une certitude de 95%.

2.3.6 Matrices de confusion

Une erreur souvent commise par les systèmes de reconnaissance est la confusion entre classes. Pour évaluer quantitativement ces erreurs, nous utilisons les matrices de confusions. Pour un niveau de rappel donné, nous calculons la matrice de confusion qui montre le taux d'erreur dues aux confusions pour chacune des classes.

3

Proposition et étude d'une méthode efficace de localisation d'objets

Sommaire

Nous proposons dans ce chapitre une méthode de localisation efficace, largement inspirée des récents travaux de l'état de l'art. Celle-ci constitue le point de départ de nos travaux, et nous nous appuierons dessus plus tard pour développer d'autres contributions. Elle est basée sur un classifieur linéaire et des descripteurs HOG et Bag-of-Words, appliqués sur une fenêtre glissante.

Dans un premier temps nous donnons une description globale de la méthode ainsi qu'une présentation détaillée de chacun de ses constituants. Ensuite, nous effectuons une étude évaluant l'influence des différents paramètres de la méthode. Cette étude nous permettra non seulement d'obtenir un détecteur de bonne qualité, mais aussi une meilleure compréhension du fonctionnement du système et la mise en évidence de ses paramètres les plus influents.

3.1 Description de la méthode

Nous présentons dans cette section la méthode de localisation initialement développée et qui sera la base de nos travaux. Fondée sur un principe de détection par fenêtre glissante, cette méthode repose sur plusieurs composantes. La première est celle de la représentation des images, c'est-à-dire le calcule d'une signature à partir des pixels contenus dans une fenêtre de l'image. La deuxième composante est celle de l'apprentissage, où, à partir d'exemples positifs (morceaux d'images contenant l'objet à localiser) et négatifs (morceaux d'images montrant tout sauf l'objet à localiser) nous construisons automatiquement les modèles des objets. Finalement, la phase de la localisation où nous explorons les images de test et appliquons les modèles appris pour retrouver les objets présents. Nous décrivons dans ce qui suit brièvement l'architecture de notre détecteur, puis nous nous attardons dans les paragraphes suivants sur les détails de chacun de ces constituants.

Représentation des images Il s'agit dans cette étape de construire une signature représentant le contenu d'une région de l'image. Cette signature doit être telle que lorsque deux régions de l'image ont un contenu visuellement similaire, leurs signatures doivent être identiques ou proches.

Bien qu'il soit techniquement possible de construire une telle signature par concaténation des niveaux de gris des pixels de la région, une telle représentation ne serait pas adaptée aux algorithmes de reconnaissance. En effet, elle ne serait pas du tout invariante à des petites transformations géométriques (translations par exemple), ni aux changements de conditions d'illumination ou aux autres variations couramment rencontrées dans les images (bruit, occultations, etc.).

Or, une bonne représentation des images est cruciale dans l'obtention de bons algorithmes de reconnaissance ; toute faiblesse à ce niveau sera subie tout au long du processus de reconnaissance. De ce fait, une attention particulière doit être accordée à cette étape.

Dans notre approche, nous nous basons sur deux des meilleures méthodes de représentation de l'état de l'art, à savoir le descripteur *HOG (Histogram of Gradient)* [13] et le descripteur *Bow (Bag-Of-Words)* [12], que nous utilisons au sein d'un processus de localisation par fenêtre glissante.

Ces deux descripteurs sont complémentaires car représentent des caractéristiques différentes de l'image (l'un la texture, l'autre la forme). Ils sont tous deux utilisés pour représenter toutes les classes d'objets, laissant au classifieur la possibilité de choisir l'information la plus pertinente pour chaque classe.

Nous effectuons ensuite une étude de l'influence des différents paramètres de la représentation. Un paramètre important dans cette représentation est la capture des

informations spatiales. Celle ci est effectuée en découpant les fenêtres de description en plusieurs sous fenêtres appelées *carreaux*, qui seront représentés séparément.

Le descripteur final est obtenu en concaténant les descripteurs des carreaux. En procédant ainsi, nous faisons implicitement l'hypothèse que les objets traités sont rigides. Même si ceci n'est généralement pas le cas pour des images réalistes où il existe une variance intra-classe non négligeable, nous espérons que la variabilité spatiale sera capturée et modélisée par le classifieur. Cette hypothèse permet d'avoir des modèles plus simples à inférer. La qualité des résultats que nous obtenons confirme la pertinence de cette hypothèse. Notons aussi que ce choix est motivé par le fait que notre algorithme sera amené à détecter des objets de petites tailles où les modèles à parties ne sont pas adéquats.

Apprentissage des modèles A partir des descripteurs des images positives et négatives nous appliquons un algorithme d'apprentissage automatique pour constituer les modèles des objets. Nous utilisons ici, comme algorithme d'apprentissage, les machines à vecteurs support (SVM). Cet algorithme a prouvé son efficacité dans nombreux travaux de l'état de l'art [10, 15, 23, 80]. Il consiste à séparer les exemples positifs et négatifs par un hyperplan maximisant la marge entre ces exemples, tout en minimisant l'erreur de classification. L'opération de classification d'un vecteur (dans notre cas un descripteur de région) consiste à calculer la distance signée entre ce descripteur et l'hyperplan, cette distance étant aussi la mesure de confiance du classifieur.

Le choix de la méthode et des exemples d'apprentissage est particulièrement importante et influence fortement la qualité des modèles produits. Aussi nous accordons une attention particulière à ce choix en suivant une procédure itérative que nous décrivons dans la section 3.1.2.

Exploration des images Pour localiser les objets, notre approche se base sur la méthode désormais classique des fenêtres glissantes ([81]). Cette méthode consiste à parcourir l'image avec une fenêtre rectangulaire balayant ainsi toutes les positions et les échelles auxquelles les objets peuvent se trouver. Le classifieur préalablement appris est alors appliqué sur chacune de ces fenêtres pour en déterminer la nature.

Comme la mesure de confiance est robuste à de légères translations et changements d'échelles, plusieurs fenêtres à fort score sont généralement détectées autours d'un même objet. Pour palier ce problème, nous appliquons un post-traitement sur les résultats obtenus afin de filtrer les détections multiples. La procédure suivie à cet effet est décrite dans la section 3.1.3.

3.1.1 Représentation des images

Nous décrivons ici les descripteurs utilisés. Le premier descripteur est le descripteur HOG [13] qui encode la forme dans les images. Le deuxième est le descripteur BoW [12], qui consiste en l'histogramme des descripteurs SIFT quantifiés et qui permet de capturer la texture des images.

Ces deux représentations se basent sur un découpage spatial de la fenêtre, de manière à capturer l'information spatiale. Plusieurs types de découpage ont été envisagés et évalués. Nous présentons dans la section 3.2.2 les résultats obtenus.

Descripteur BoW

Nous proposons ici l'utilisation d'une représentation standard par BoW où chacun des carreaux est représenté par un histogramme de la fréquence d'apparition des mots visuels.

Les carreaux sont obtenus après découpage de la fenêtre suivant la méthode des pyramides spatiales proposée dans [48]. Cette méthode consiste à diviser récursivement la fenêtre en des régions de plus en plus fines, et à calculer le descripteur de chacune de ces régions séparément. Ces pyramides sont définies par le nombre de niveaux qu'elles contiennent. Le descripteur final est obtenu en concaténant les BoW de tous les carreaux.

Les régions (appelées aussi vignettes) sont extraites à toutes les positions et les échelles en suivant une grille dense (vignettes de 12×12 pixels, avec un pas de 6 pixels en hauteur et en largeur pour la première échelle et un pas de 1.2 pour l'échelle). Ces vignettes sont décrites par le descripteur *Opponent SIFT* [77] qui est une variante couleur du descripteur SIFT proposé par [56]. Le vocabulaire visuel est constitué à partir des vignettes extraites des images d'apprentissage moyennant l'algorithme de clustering K-means. Ensuite, chaque descripteur de vignette est quantifié en l'affectant au plus proche mot visuel du vocabulaire.

Descripteur HOG

En plus du descripteur BoW, nous proposons de représenter les objets par un descripteur HOG [13], qui comme le SIFT, est un descripteur reposant sur l'histogramme des orientations des gradients de l'image.

Nous proposons ici de découper les fenêtres de description en carreaux denses. Chaque carreau est alors décrit par un HOG et le vecteur de description finale est obtenu en concaténant les descripteurs des carreaux.

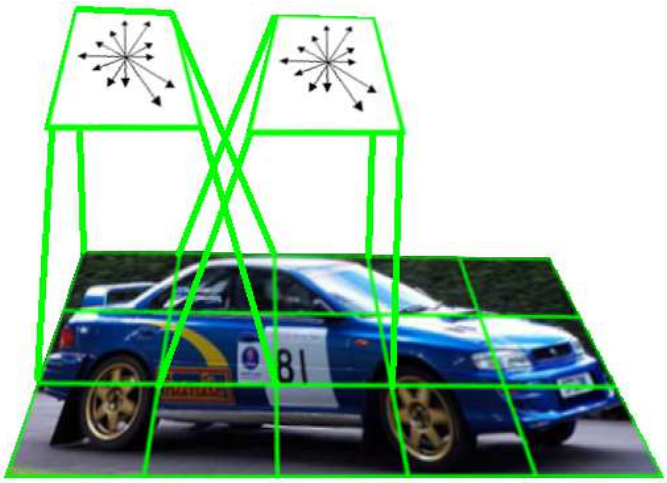


Fig. 3.1: Exemple de découpage spatial utilisé avec le descripteur HOG.

Dans le découpage utilisé, la fenêtre est divisée en un ensemble de carreaux denses en chevauchement ayant une taille fixe [14, 28]. En fonction du ratio largeur/hauteur moyen d'un objet et du nombre de carreaux souhaité T, nous tentons d'avoir des carreaux aussi carrés que possible. Nous aurons donc $round(\sqrt{\frac{TW}{H}})$ carreaux en largeur, et $round(\sqrt{\frac{TH}{W}})$ en hauteur. La résolution globale du découpage est contrôlée par le paramètre T. Nous appelons par la suite ce découpage *découpage adapté*, qui est déterminé par deux paramètres principaux : (a) le nombre total de carreaux et (b) le chevauchement entre les carreaux. Nous présentons dans notre étude paramétrique (section 3.2.2) plusieurs expérimentations en utilisant un nombre différent de carreaux, avec et sans chevauchement, et donnons des conclusions quand à l'influence de ces deux paramètres. Ce découpage adapté est aussi comparé au découpage dit *régulier* pour lequel le nombre de carreaux suivant la hauteur est le même que suivant la largeur.

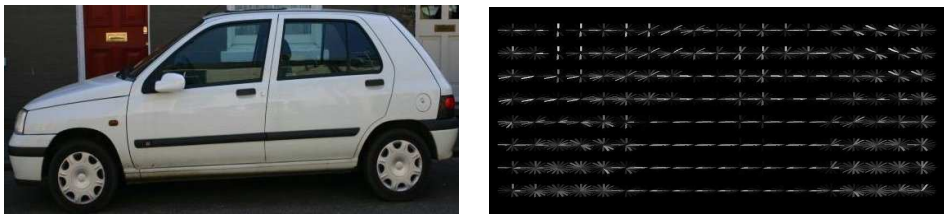


Fig. 3.2: Descripteur HOG (à droite) extrait de l'image de voiture (à gauche).

Notre implémentation des descripteurs HOG est inspirée de celle de Zhu *et al.* [89] qui proposent une implémentation rapide utilisant les histogrammes intégraux [66]. Nous utilisons la même approche : après avoir quantifié l'orientation du gradient à

chaque pixel moyennant une interpolation bilinéaire, nous calculons l'image intégrale pour chacune des orientations discrètes. Ainsi, un HOG est calculé en $4 \times$ *(nombre d'orientations)* opérations élémentaires. Notons que cette technique ne peut être combinée avec l'interpolation spatiale suggérée par [14] où le poids d'un gradient est affecté à plusieurs carreaux avoisinants.

Normalisation des descripteurs

La normalisation des descripteurs est une étape essentielle pour obtenir une robustesse face aux changements de contraste et aux changements d'échelle. Différentes normalisations des descripteurs sont possibles. Dalal et Triggs [14] ont expérimenté quatre types de normalisation : L2, L2 tronqué, L1 et L1 suivi de l'application de la racine carré. Ils ont remarqué que le L2 et le L1 suivi de la racine carré donnaient les meilleurs résultats, et qu'omettre l'étape de normalisation réduisait considérablement les performances dans le cas du descripteur HOG. Zhu *et al.* [89] ont aussi comparé les normes L1 et L2 et n'ont pas remarqué une différence notable entre les deux.

Nous avons choisi ici de travailler avec la norme L2 qui, elle-même, peut être utilisée de différentes manières. (a) Les histogrammes locaux (de chacun des carreaux) peuvent être normalisés indépendamment ou par groupes. (b) Une autre alternative est de normaliser globalement le descripteur entier. (c) Il est aussi possible de ne pas du tout normaliser le descripteur. Ces trois possibilités sont comparées dans notre étude paramétrique (voir section 3.2.2).

3.1.2 Apprentissage des modèles

Nous utilisons ici les SVMs linéaires comme méthode de classification. En plus de leurs bonnes performances, ces classifieurs réalisent les opérations de classification rapidement car il suffit pour cela de calculer un produit scalaire. Notons que nous apprenons un classifieur par ensemble de vues (vue de face, de profil, etc.), les vues avant et arrière ainsi que droite et gauche étant fusionnées.

Nous décrivons dans la section suivante la méthode retenue pour choisir les exemples d'apprentissage positifs et négatifs.

Exemples positifs

L'ensemble initial des exemples positifs est celui provenant des annotations fournies avec la base de données. Il s'agit de rectangles (boites englobantes) avec pour chaque boite le type d'objet ainsi que son orientation (vue de face, etc.).

A cet ensemble nous ajoutons des exemples générés artificiellement suivant la procédure proposée par Laptev [46]. Dans ces travaux l'auteur a essayé de remédier au manque d'exemples d'apprentissage en ajoutant des exemples obtenus en appliquant des petites transformations géométriques aux exemples originaux. L'auteur rapporte que cette technique offre un gain considérable en performance. Ici nous ajoutons des exemples obtenus en appliquant des translations, des changements d'échelle et une symétrie axiale (axe vertical passant par le milieu de la fenêtre) aux exemples initiaux. Ainsi, non seulement nous augmentons notre ensemble d'apprentissage mais nous acquérons une robustesse face à ce même genre de transformations. Nous évaluons dans l'étude paramétrique l'influence de cette procédure sur la qualité des détections.

Exemples négatifs

Les exemples négatifs sont obtenus via un processus itératif. Premièrement, les objets autres que celui à localiser sont introduits comme exemples négatifs (en utilisant là encore les informations de vérité terrain données avec la base). A cet ensemble, nous ajoutons des fenêtres du fond sélectionnées aléatoirement à partir des images d'apprentissage.

Nous apprenons alors un classifieur initial avec ces exemples. Ce classifieur est ensuite appliqué sur les images d'apprentissages au moyen d'un algorithme de fenêtre glissante. Les faux positifs, ou *exemples difficiles*, sont alors ajoutés à l'ensemble d'exemples négatifs et un nouveau classifieur est appris. Cette procédure est répétée plusieurs fois pour obtenir le classifieur final.

3.1.3 Suppression des non maximas

Pour éliminer les détections multiples, nous appliquons un algorithme de suppression des non maximas. Celui-ci se base sur une approche itérative proche de celle de Fritz *et al.* [31]. La détection ayant le plus haut score est sélectionnée et toutes celles qui chevauchent cette dernière y sont affectées et enlevées de l'ensemble des détections. Seuls les chevauchement ayant un indice de Jaccard (voir paragraphe 2.3.1) supérieur à 0.4 sont considérés. Cette procédure est répétée jusqu'à ce que toutes les détections soient traitées.

Ayant observé que les détections sont généralement concentrées autours des vrais positifs alors que les faux positifs se retrouvent généralement éparpillés, le score final affecté à une fenêtre est modifié selon la formule suivante :

$$Score = S_{\text{sélectionnée}} + \log(\sum_{i=1}^{Nb_{\text{affectées}}} S_i), \tag{3.1}$$

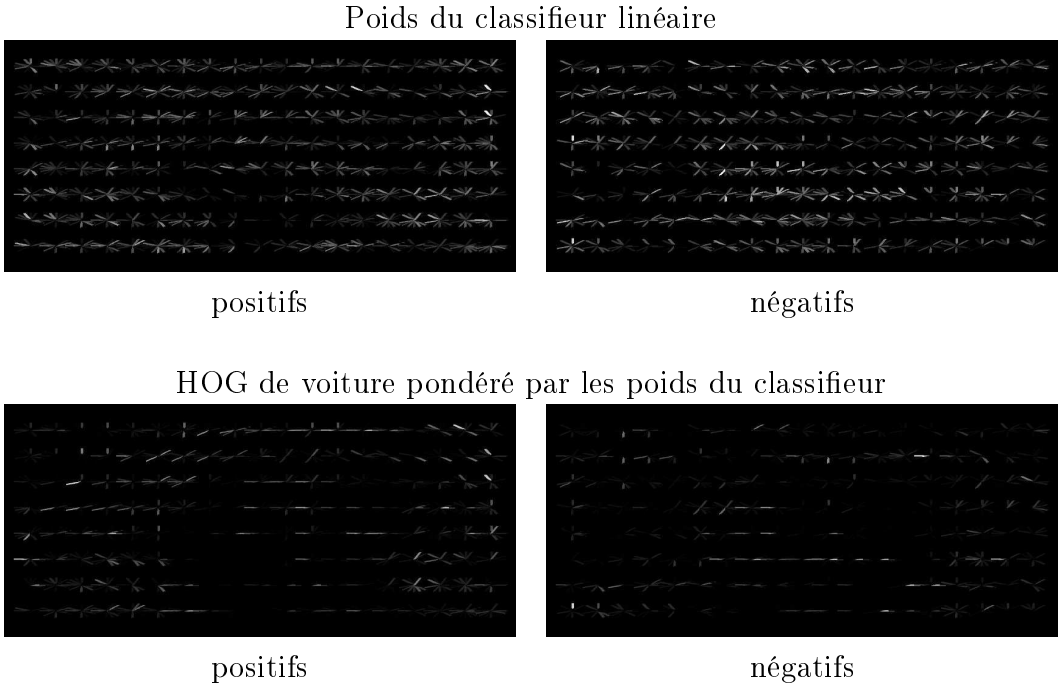


Fig. 3.3: Classification des HOG : La première ligne montre les poids du vecteur W du classifieur linéaire appris pour les voitures vues de profil. La deuxième ligne montre le HOG de la voiture de la figure 3.2 pondéré par ces poids.

où $S_{\text{sélectionnée}}$ est le score de la fenêtre sélectionnée et S_i celui de la $i^{\text{ème}}$ fenêtre qui y est affectée. En procédant ainsi, nous prenons en considération le nombre et le score des détections avoisinantes.

3.2 Étude paramétrique

Dans cette section nous présentons les résultats de l'étude paramétrique réalisée sur notre méthode. Notre motivation est animée par deux buts distincts. Premièrement, celui de démontrer que notre méthode atteint des performances similaires à l'état de l'art sur des bases de données parmi les plus difficiles. Deuxièmement, celle de déterminer quels sont les paramètres les plus importants de notre système (ceux responsables des bons résultats observés).

Cette section est organisée comme suit. Nous commençons par présenter le protocole expérimental, nous présentons ensuite l'étude des paramètres liés à la représentation des images, et enfin, nous présentons l'étude de ceux liés à l'apprentissage des détecteurs.

3.2.1 Protocole expérimental

En raison du nombre important de paramètres de notre système, l'évaluation exhaustive de toutes les combinaisons de paramètres serait très coûteuse en temps de calcul, voire impossible.

Pour mener à bien notre étude paramétrique, nous avons déterminé expérimentalement une configuration que nous considérons au départ comme proche de l'optimale et que nous appelons *configuration de référence* (voir tables 3.1 et 3.9). Nous évaluons alors l'influence de chaque paramètre en mesurant l'impact du changement de la valeur de référence de ce paramètre sur la performance globale.

Nous ne rapportons ici que les résultats pour les paramètres les plus influents. Donner un nombre trop grand de valeurs numériques n'aiderait pas le lecteur à tirer des conclusions et serait déroutant. Nous avons donc préféré nous appuyer sur des tests statistiques, à savoir le *test de Friedman* 2.3.5 et le *T-test* 2.3.5 pour présenter une vue synthétique de l'influence des différents paramètres.

Les autres critères d'évaluation utilisés ici sont décrits dans la section 2.3. Notons aussi que les performances présentées ici sont exprimées en mAP, qui reflète la qualité moyenne des performances sur l'ensemble des classes.

La base d'images sur laquelle nous effectuons notre étude paramétrique est la base PASCAL 2007 [19] qui contient des images de 20 classes prises dans des conditions réalistes.

3.2.2 Représentation des images

Nous étudions dans cette section l'influence des différents paramètres des descripteurs. Le premier paramètre étudié pour chacun de ces deux descripteurs est celui du découpage spatial de la fenêtre de détection. Nous évaluons ensuite l'influence de l'élargissement de la fenêtre de détection sur les performances. Puis nous évaluons l'importance de la normalisation des descripteurs et finalement nous montrons la complémentarité des deux descripteurs utilisés.

La table 3.1 montre la configuration de référence utilisée pour la représentation d'images.

Paramètres du descripteur HOG

Évaluation du découpage spatial Le découpage spatial que nous utilisons pour le descripteur HOG est le découpage adapté décrit dans la section 3.1.1. Nous utilisons aussi une grille régulière et comparons les résultats obtenus par chacune de ces deux approches.

Paramètre		Valeur par défaut
HOG	Découpage spatial	grille adaptée, entre 80 et 200 carreaux
	Nb orientations	16
	Chevauchement	oui
	Orientation range	2π
	Interpolation	non
BoW	Découpage spatial	Pyramide spatiale à 3 niveaux
	Taille du vocabulaire	100
	Point d'intérêt	Dense : pas spatial 6px, pas d'échelle 1.2
Normalisation		L2, par carreau
Élargissement		4px

Tab. 3.1: Configuration de référence pour la représentation d'images comprenant les paramètres du descripteur HOG, du BoW, de la normalisation et de l'élargissement de la fenêtre.

Nous construisons plusieurs configurations avec un nombre de carreaux allant de 40 à 350, pouvant se chevaucher ou non, avec des grilles régulières ou adaptées. Ces configurations sont évaluées au moyen du détecteur préalablement décrit et en utilisant la configuration de référence. Les résultats obtenus par ces différentes configurations sont comparés par le *test de Friedman* (voir définition section 2.3.5) pour obtenir des groupes de configurations significativement différents.

Notons que pour pouvoir évaluer l'effet du chevauchement tout en gardant la même résolution, les configurations à carreaux non chevauchants sont obtenues à partir des configurations à carreaux chevauchants. Pour ce faire, les carreaux redondants (ceux qui couvrent des régions déjà couvertes) sont éliminés. De ce fait, le nombre de carreaux indiqué est supérieur (de l'ordre de 4 fois) au nombre réel de carreaux présents. Dans ces cas, le nombre indique plutôt la résolution et est donné à titre indicatif.

Dans la table 3.2, nous montrons les groupes de configurations significativement différents obtenus par le test de Friedman. Nous obtenons 12 groupes de configurations équivalentes en termes de performances. La différence de performance de ces configurations est présentée dans la figure 3.4, où nous montrons avec un *T-test* le gain obtenu par chaque configuration par rapport à la plus mauvaise configuration (dénoté A) en termes de mAP. La différence de performance entre la meilleure et la plus mauvaise configuration est de 7%. Cette différence est très significative et montre l'importance du choix du découpage spatial. Nous pouvons aussi remarquer que la plupart des configurations du groupe de tête utilise une grille adaptée et que pour la totalité, il y a un chevauchement entre les carreaux. De ces observations nous pouvons tirer trois conclusions principales :

	Rang Moyen	Groupes											
non chev, réguli, 49 carreaux	19.0	A											
non chev, adapt, 40 carreaux	18.5	A	B										
non chev, adapt, 60 carreaux	17.9	A	B	C									
non chev, réguli, 121 carreaux	14.5		B	C	D								
chevauch, adapt, 40 carreaux	14.3		B	C	D								
chevauch, réguli, 49 carreaux	13.8			C	D	E							
non chev, adapt, 120 carreaux	13.0				D	E	F						
non chev, adapt, 140 carreaux	12.9				D	E	F						
non chev, adapt, 80 carreaux	12.6				D	E	F	G					
non chev, adapt, 200 carreaux	11.8				D	E	F	G					
non chev, réguli, 225 carreaux	11.3				D	E	F	G	H				
non chev, adapt, 160 carreaux	11.0				D	E	F	G	H				
non chev, réguli, 361 carreaux	10.2				D	E	F	G	H	I			
non chev, adapt, 100 carreaux	9.9					E	F	G	H	I			
chevauch, réguli, 361 carreaux	8.8						F	G	H	I	J		
chevauch, adapt, 60 carreaux	8.7						F	G	H	I	J		
chevauch, réguli, 121 carreaux	8.5							G	H	I	J	K	
chevauch, réguli, 225 carreaux	7.2								H	I	J	K	L
chevauch, adapt, 80 carreaux	6.1									I	J	K	L
chevauch, adapt, 100 carreaux	5.4										J	K	L
chevauch, adapt, 140 carreaux	4.6										J	K	L
chevauch, adapt, 200 carreaux	4.3											K	L
chevauch, adapt, 120 carreaux	3.9												L
chevauch, adapt, 160 carreaux	2.8												L

Tab. 3.2: Groupes de configurations obtenus par le test de Friedman : la configuration optimale reposes sur un découpage adapté, avec du chevauchement entre les cellules et un nombre de carreaux proche de 150.

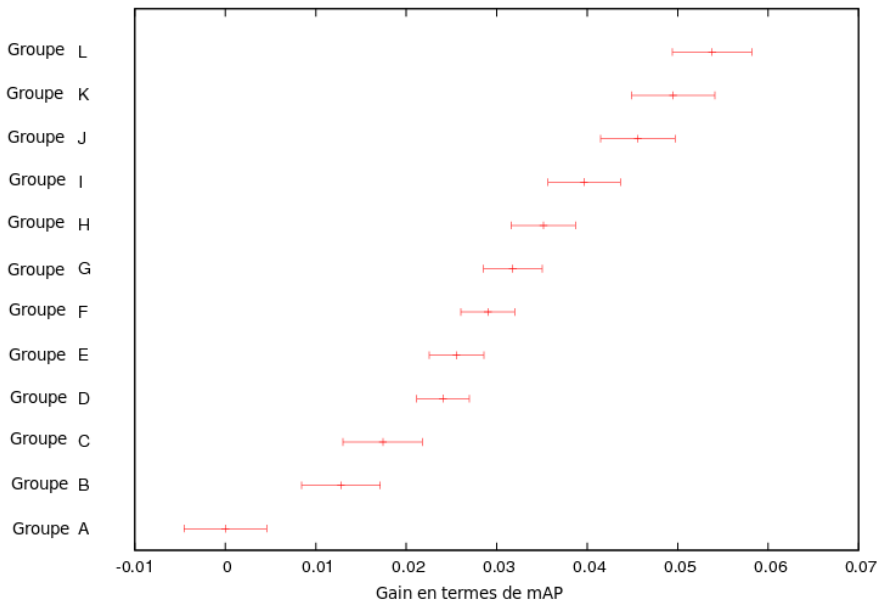


Fig. 3.4: Influence du découpage spatial sur les HOGs : Gain obtenu par les différents groupes du test de Friedman par rapport au plus mauvais groupe (A).

1. Bien que l'optimisation du nombre de carreaux pour chaque classe d'objets donne des résultats légèrement meilleurs, une grille qui contient à peu près 150 carreaux produit une très bonne représentation de l'objet.
2. Le chevauchement entre les carreaux est primordial à la bonne représentation des objets. En effet, nous observons une amélioration allant jusqu'à 3% en introduisant le chevauchement.
3. L'utilisation des grilles adaptées s'avère importante même si ce paramètre est moins influent que les précédents.

Orientation des HOGs Nous évaluons ici l'influence du nombre d'orientations discrètes (ou *bin*) utilisées lors du calcul du descripteur HOG, ainsi que de l'influence de la prise en compte ou non du signe de l'orientation du gradient sur la qualité du descripteur.

Le nombre de bins correspond à la résolution angulaire utilisée. Un nombre trop élevé de bins correspond à une description précise de l'image mais à partir de laquelle le classifieur aura du mal à généraliser. A l'opposé, un nombre trop faible d'orientations donnera une description grossière contenant peu d'information.

En ce qui concerne le signe de l'orientation du gradient, lorsqu'il est ignoré, cette orientation est comprise entre 0 et π. Dans le cas où le signe est pris en compte, elle est comprise entre 0 et 2π. Pour donner un exemple, les 3 images de la figure 3.5 auront la même signature avec la première méthode alors que chacune aura une signature différente avec la deuxième. Il est donc clair qu'ignorer le signe du gradient permet d'apporter une certaine robustesse au descripteur (exemple d'une image d'une personne portant une chemise noire et un pantalon blanc et la même personne portant une chemise blanche et un pantalon noir) au prix d'une perte d'information.

Pour pouvoir évaluer l'influence de chacun de ces deux paramètres, nous faisons varier le nombre de bins de 8 à 32 en considérant à chaque fois les cas où le signe de l'orientation du gradient est pris en compte ou est ignoré. Les résultats de cette évaluation sont présentés dans le tableau 3.3 où nous présentons le mAP, l'amélioration du mAP obtenues, ainsi que l'intervalle d'incertitude dans cette amélioration avec une confiance de 95% (obtenu avec le T-test).

Nous observons d'abord que les HOGs à 16 bins s'avèrent meilleurs de 3% que ceux à 8 bins, alors qu'ils sont équivalents à ceux à 32 bins. Nous pouvons donc conclure que ce nombre de bins constitue un compromis acceptable entre précision et généralisation. Pour ce qui est des orientations signées, les résultats doivent être comparés à résolution angulaire équivalente (par exemple 16 bins orientés contre 8 bins non orientés). Nous remarquons qu'elle donne toujours de meilleurs résultats (+3.3% pour la configuration de référence). Nous pouvons donc conclure que d'une part cette information est discriminante, et que d'autre part, le nombre d'exemples présents

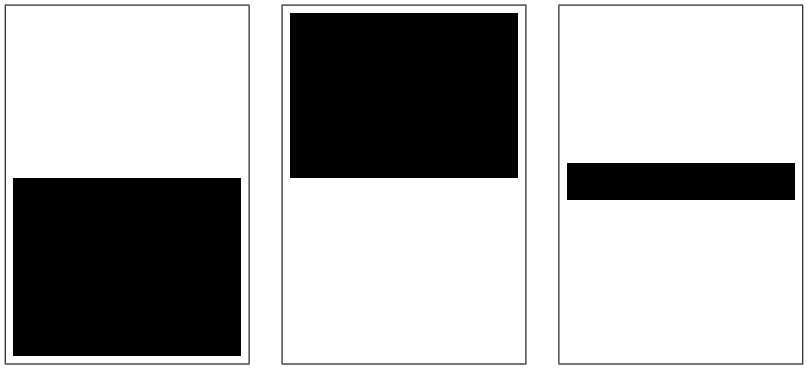


Fig. 3.5: Influence du signe de la prise en compte du signe de l'orientation du gradient : exemple d'imagettes qui obtiendraient la même signature en ignorant le signe et qui auraient des signatures différentes en le prenant en compte.

Configuration	mAP	Amél. moyenne	Confiance
8 ori. intervalle $[0 \cdots \pi]$	11.3	-3.3	$\pm$ 0.58
8 ori. intervalle $[0 \ldots 2\pi]$	11.5	-3.2	$\pm$ 0.57
16 ori. intervalle $[0 \cdots \pi]$	12.3	-2.3	$\pm$ 0.68
16 ori. intervalle $[0 \ldots 2\pi]$ (ref)	14.6	0.0	$\pm$ 0.11
32 ori. intervalle $[0 \cdots \pi]$	13.3	-1.3	$\pm$ 0.71
32 ori. intervalle $[0 \ldots 2\pi]$	14.4	-0.2	$\pm$ 0.49

Tab. 3.3: mAP et le gain en mAP pour différents paramètres de la quantification de l'orientation du gradient.

est suffisant pour pouvoir se passer de la robustesse apportée par des orientations non signées.

Paramètres du descripteur BoW

Concernant le descripteur BoW, nous évaluons l'influence du découpage spatial et fixons la taille du vocabulaire visuel à 100 mots visuels.

Découpage spatial des BoW Nous utilisons ici un seul type de découpage spatial, à savoir la pyramide spatiale, qui s'est montré très performante avec ce type de descripteurs [48]. Le paramètre que nous considérons ici est celui du nombre de niveaux de la pyramide. Nous faisons donc varier ce paramètre est calculons la mAP obtenue à chaque fois. Les résultats sont présentés dans le tableau 3.4.

Configuration	mAP	Amélioration moyenne	Confiance
Pyramide de niveau 0	2.9	-12.1	± 2.23
Pyramide de niveau 1	7.6	-7.4	± 1.19
Pyramide de niveau 2	13.3	-1.7	± 0.61
Pyramide de niveau 3 (ref)	15.0	0	± 0.22
Pyramide de niveau 4	13.3	-1.7	± 0.53

Tab. 3.4: mAP et le gain en mAP pour différents nombres de niveaux de la pyramide spatiale.

Nous remarquons que l'utilisation d'un seul niveau donne des résultats médiocres. Cette configuration correspond à l'implémentation de base des BoW, qui n'incorpore aucune information spatiale de l'objet.

Ceci montre l'extrême importance de cette information spatiale pour la localisation. L'utilisation d'un niveau de plus porte la mAP à 7.6% et la meilleure performance est en utilisant 3 niveaux avec une mAP de 15%. Notons que l'utilisation de plus que 3 niveaux détériore les résultats. En effet, en faisant ainsi nous augmentons considérablement la taille du descripteur (de 8500 à 34100 dimensions). L'algorithme d'apprentissage se trouve donc noyé dans les détails et tend à s'adapter trop aux exemples d'apprentissage (phénomène de sur-apprentissage).

Lorsqu'il est comparé au HOG (voir table 3.5), le descripteur BoW obtient des résultats légèrement supérieurs. D'un autre coté, la meilleure configuration pour les BoW est plus coûteuse en termes de temps de calcul et d'espace mémoire (descripteur à 8500 dimensions contre $\approx$ 2500 dimensions pour le HOG). Dans la mesure où ces deux réprésentations sont différentes, il n'est pas surprenant que la plus adaptée à une classe d'objets dépende de la classe considérée (voir table 3.8). Ceci suggère que la combinaison des deux descripteurs peut être bénéfique. Nous montrerons dans le paragraphe 3.2.2 l'influence de cette combinaison.

Élargissement de la fenêtre de description

Nous évaluons ici l'effet de l'élargissement des fenêtres de description sur les performances. Certains travaux montrent que cela aide à la construction d'une meilleure représentation des objets [14]. Nous pouvons l'expliquer par deux raisons : la prise en compte du contexte avoisinant l'objet, ou encore l'ajout des contours de l'objet qui pouvaient êtres exclus du fait d'une fenêtre trop ajustée à ses bords.

Pour évaluer l'influence de cette étape, nous comparons les résultats obtenus sans élargissement avec ceux obtenus par les deux méthodes d'élargissement : (a) élargissement constant de quelques pixels ou (b) élargissement proportionnel à la taille

de la fenêtre. La première méthode permet de mettre l'accent sur les contours de l'objet alors que la deuxième se concentrera plus sur le contexte entourant l'objet. Pour l'élargissement par pourcentage nous considérons des élargissements d'au maximum 20%. Au delà nous dépassons la frontière de l'image pour un nombre trop important d'objets.

HOG			
Élargissement par	mAP	Amélioration moyenne	Confiance
Sans élargir	12.7	-1.9	± 0.46
4 pixels (ref)	14.6	0.0	± 0.11
8 pixels	14.3	-0.3	± 0.36
10 %	14.2	-0.4	± 0.31
20 %	14.2	-0.4	± 0.36
BoW			
Elargissement par	mAP	Amélioration moyenne	Confiance
Sans élargir	15.3	0.3	± 0.54
4 pixels (ref)	15.0	0.0	± 0.22
8 pixels	15.1	0.1	± 0.58
10 %	15.6	0.6	± 0.62
20 %	14.7	-0.3	± 0.59

Tab. 3.5: mAP et le gain en mAP pour différentes configurations d'élargissement.

Le tableau 3.5 montre les résultats obtenus avec ces différentes configurations pour les descripteurs HOG et BoW. Nous remarquons que l'élargissement a peu d'effet sur le descripteur BoW. Une explication possible est que l'élargissement est déjà présent de part la construction des BoW. En effet, un patch est inclus dans le BoW si son centre est dans la fenêtre de description (sans exclure les patchs dont une partie est hors de la fenêtre).

D'un autre coté, l'élargissement apparait comme un paramètre important pour le descripteur HOG avec un gain de 1.9%. En revanche, toutes les méthodes d'élargissement semblent être équivalentes ce qui suggère que l'information la plus importante capturé par l'élargissement est le contour des objets.

Normalisation des données

La normalisation permet d'acquérir une robustesse contre les changements d'illumination et d'échelles. Nous évaluons ici son influence sur les performances de notre détecteur. Nous utilisons ici la norme L2 qui est la mieux adaptée au noyau linéaire utilisé par le SVM. Différentes configurations sont considérées, à savoir :

- Pas de normalisation.
- La normalisation globale où le descripteur entier (la concaténation des descripteurs de carreaux) est normalisé.
- La normalisation locale où chaque descripteur de carreau est normalisé seul. Cette normalisation offre une meilleure résistance aux changements d'illumination. En effet, si un point est extrêmement lumineux, il n'affectera que le carreau où il est présent. En revanche, l'inconvénient (comparé à la précédente) est que cette normalisation nous fait perdre le lien entre les descripteurs des carreaux.

HOG			
Configuration	mAP	Amélioration moyenne	Confiance
Sans normalis.	02.3	-12.3	± 1.40
Globale	12.8	-1.8	± 0.65
Par carreau (ref)	14.6	0.0	± 0.11
BoW			
Configuration	mAP	Amélioration moyenne	Confiance
Sans normalis.	13.4	-1.6	± 0.50
Globale	12.7	-2.3	± 0.87
Par carreau (ref)	15.0	0.0	± 0.22

Tab. 3.6: mAP et le gain en mAP pour différentes normalisations.

Le tableau 3.6 montre les résultats obtenus avec ces différentes configurations. La normalisation s'avère être une étape nécessaire avec le descripteur HOG, au vu des résultats très mauvais obtenus sans normalisation.

Le descripteur BoW quand à lui, donne des résultats acceptables même sans normalisation. Ceci est dû au fait que les SIFT sont déjà normalisés avant d'être quantifiés par le biais du vocabulaire visuel. Sans la normalisation, le descripteur perd uniquement sa robustesse par rapport aux changements d'échelle.

Nous pouvons aussi remarquer que la normalisation locale donne de meilleures performances que la normalisation globale et ce pour les deux descripteurs HOG et BoW. Une explication possible est la meilleure robustesse contre les changements d'illumination qu'offre cette normalisation.

Combinaison du descripteur HOG et du descripteur BoW

Bien qu'ils soient tous les deux basés sur les mêmes informations élémentaires, à savoir les gradients de l'image, les descripteurs HOG et BoW représentent les images de manières différentes. Ceci est montré dans la figure 3.6 où nous mesurons l'intersection entre les ensembles de vrais positifs retournés par un détecteur basé sur

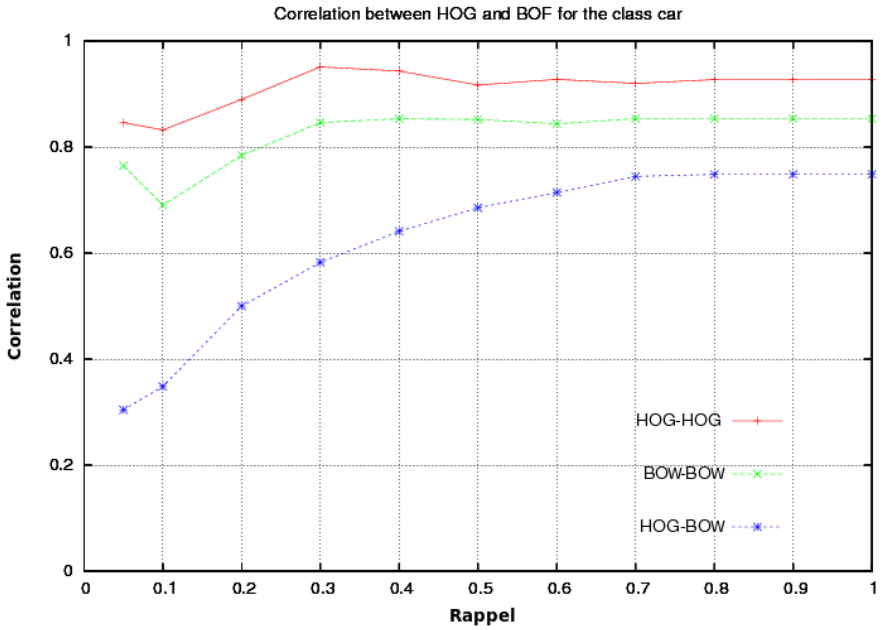


Fig. 3.6: Intersection entre les vrais positifs obtenus HOG et BoW. Les résultats présentés ici sont relatifs à la classe "car", mais des résultats similaires ont été observés pour les autres classes.

le descripteur HOG et un autre basé sur le descripteur BoW pour différentes valeurs du rappel. Pour mieux interpréter les résultats présentés, nous donnons à titre de référence les mêmes courbes obtenues pour deux détecteurs basés sur les HOG et deux détecteurs basés sur les BoW (La différence entre ces détecteurs vient de la variabilité introduite par la sélection aléatoire d'exemples négatifs). Nous remarquons que les vrais positifs obtenus avec la combinaison des HOG et des BoW sont différents et ce surtout pour les faibles rappels. Ceci vient du fait que les meilleures détections obtenues par l'un et par l'autre sont significativement différentes. Ces résultats confirment l'intérêt qu'il y a à combiner les deux descriptions.

Descripteur	mAP	Amélioration moyenne	Confiance
HOG (ref)	14.6	0.0	± 0.11
HOG+BoW	17.6	3.0	± 0.77
BoW (ref)	15.0	0.0	± 0.22
HOG+BoW	17.6	2.6	± 0.79

Tab. 3.7: mAP et le gain en mAP en combinant HOG et BoW.

Le résultat de la combinaison des descripteurs est présenté dans le tableau 3.7 où nous montrons l'amélioration obtenue ainsi que la confiance qu'il est possible d'avoir

dans cette amélioration. La combinaison est faite ici par simple concaténation des descripteurs. Cette combinaison offre un gain significatif de 3% lorsqu'elle est comparée au HOG et de 2% en la comparant au BoW. Les résultats détaillés de cette combinaison sont donnés dans le tableau 3.8. Nous observons que les objets déterminés par leur forme obtiennent généralement de meilleurs résultats ave le HOG (bicycle, car, bus) mais que les résultats des deux détecteurs restent tout de même assez proches. Nous observons aussi que la combinaison offre, à quelques exceptions près, une nette amélioration des performances.

	HOG	BoW	HOG+BoW
aeroplane	10.0	16.9	22.4
bicycle	27.8	21.2	30.5
bird	04.7	04.9	03.3
boat	00.6	04.8	01.8
bottle	11.4	07.3	11.2
bus	31.7	25.2	26.4
car	33.9	28.4	36.7
cat	02.6	06.9	06.0
chair	10.1	09.8	11.1
cow	14.9	10.3	14.3
diningtable	09.7	06.7	10.9
dog	01.8	06.9	07.6
horse	28.1	30.5	33.8
motorbike	22.6	26.6	27.2
person	12.2	13.1	14.7
pottedplant	09.9	09.4	09.8
sheep	10.0	12.5	15.1
sofa	04.3	12.1	14.7
train	19.3	17.0	22.4
tvmonitor	26.1	28.5	32.2
Moyenne	14.6	15.0	17.6

Tab. 3.8: Résultats pour toutes les classes de la base PASCAL 2007 obtenus avec les différentes méthodes de représentation d'images exprimées en mAP.

3.2.3 Sélection des exemples négatifs et positifs

Dans cette section nous évaluons l'influence de la sélection des exemples d'apprentissage.

Comme pour les paramètres de représentation, nous montrons dans le tableau 3.9 les paramètres de référence de sélection des exemples d'apprentissage.

Paramètre		Valeur par défaut
Exemples positifs		12 par annotation symétrie, translation et chgt. d'échelle
Ex. négatifs	Nb réentrainement	4
	Fond	4 par positif
	Ex. difficiles	Nb Ex. fond / num. itération

Tab. 3.9: Configuration de référence pour l'apprentissage du détecteur définissant le nombre de réapprentissage et la méthode de sélection des exemples.

Exemples positifs additionnels

Une technique parfois utilisée [10, 46] pour élargir l'ensemble d'apprentissage est d'y ajouter de nouveaux exemples par symétrie, translation et changement d'échelle par rapport à l'annotation fournie. Ceci permet non seulement d'avoir plus d'exemples d'apprentissage mais aussi d'acquérir une robustesse contre ces mêmes transformations. Nous avons appliqué cette technique, et étonnamment, nous n'avons pas constaté d'amélioration de performance. La même méthode appliquée à la base d'images PASCAL 2006 produit au contraire une amélioration de 3% en termes de mAP[1]. Nous avons aussi essayé de faire varier le pas de la fenêtre glissante et avons observé que pour des pas plus importants, les détecteurs appris avec les exemples transformés donnent de meilleurs résultats.

Exemples négatifs additionnels

Lorsque les exemples d'apprentissage sont sélectionnés aléatoirement ils ont de fortes chances d'être loin de la frontière de classification 'idéale'.

Pour éviter ce problème, et à défaut de pouvoir utiliser tout les exemples négatifs possibles, nous suivons une procédure itérative pour la sélection d'exemples négatifs difficiles sur lesquels le détecteur initialement appris se trompe et qui seront utilisés pour réentrainer le détecteur. Nous présentons dans la table 3.10 les résultats obtenus sans et avec plusieurs itérations de réentrainement. Notons que pour une comparaison équitable, tous les détecteurs ont été entrainés avec le même nombre d'exemples d'apprentissage, seule leur provenance change (sélection aléatoire ou sélection des exemples difficiles).

[1] ceci est le seul résultat présenté sur la base PASCAL 2006, tous les autres étant obtenus sur la base PASCAL 2007

La procédure de réentrainement utilisée s'avère primordiale pour l'obtention d'un bon détecteur. Une seule itération suffit pour obtenir une amélioration de 7.7% de mAP. Avec une ou deux itérations de plus, les détecteurs sont améliorés. Au delà, les performances se stabilisent. Pour montrer qualitativement l'effet de cette

Nb Itération	mAP	Amélioration moyenne	Confiance
Sans ré-entrain.	4.7	-9.8	± 1.9
1 itération	12.5	-2.1	± 0.59
2 itération	12.9	-1.7	± 0.46
3 itération	14.6	0	± 0.36
4 itération (ref)	14.6	0	± 0.11

Tab. 3.10: mAP et gain en termes de mAP pour différents nombres de réentrainement.

procédure de réentrainement, nous présentons dans la figure 3.7 des exemples de faux positifs que nous avons réussi à éliminer grâce à la procédure de réentrainement. Nous observons que ces exemples sont de plus en plus difficiles : les exemples éliminés à la première itération apparaissent pour l'œil humain comme extrêmement simples à rejeter (ceci explique les mauvais résultats obtenus sans réentrainement). A partir de la deuxième itération, les exemples deviennent de plus en plus difficiles à éliminer.

3.2.4 Effet du nombre d'exemples d'apprentissage

Nous évaluons dans cette section l'influence du nombre d'exemples d'apprentissage sur la qualité de la localisation. Pour ce faire, nous apprenons plusieurs détecteurs (utilisant uniquement les HOG) en prenant à chaque fois une sous partie de la base d'apprentissage. Les résultats obtenus sont présentés dans la table 3.11.

Comme attendu, nous remarquons que l'augmentation du nombre d'exemples d'apprentissage améliore notablement les performances. Remarquons que le mAP ne semble pas saturer avec le nombre actuel d'exemples ce qui suggère que les performances peuvent être nettement améliorées en augmentant le nombre d'exemples d'apprentissage de la base.

3.3 Conclusion

Ce chapitre a décrit une méthode de localisation efficace développée durant cette thèse. Cette méthode se base sur les récents travaux de l'état de l'art et combine les techniques les plus avancées et les plus adaptées à notre problématique. Une étude

Fig. 3.7: Exemple de fausses détections éliminées par notre procédure de réentrainement : la difficulté des fausses détections croit avec les itérations.

paramétrique approfondie a été menée pour déterminer la valeur des paramètres à utiliser par la suite afin d'obtenir de bonnes performances. Elle a permis, par exemple, de souligner l'importance d'une bonne représentation d'images, et notamment de certains aspects très influents, comme le type de découpage spatial, le chevauchement entre les carreaux, ou encore la normalisation des descripteurs. D'autres paramètres sont apparus comme extrêmement importants comme le choix des exemples d'apprentissage, surtout des exemples négatifs obtenus itérativement. Cette étude a permis une compréhension approfondie du fonctionnement du système et a également permis de mettre en avant quels étaient les paramètres les plus influents.

	10%	25%	50%	75%	100%
aeroplane	9.2	9.3	6.3	3.2	10.0
bicycle	16.8	20.4	19.2	25.5	27.8
bird	0.9	0.2	0.3	0.6	04.7
boat	0.8	0.1	1.4	0.3	00.6
bottle	9.3	10.2	10.7	11.2	11.4
bus	20.8	26.3	26.7	28.3	31.7
car	29.0	32.3	31.4	31.7	33.9
cat	0.2	0.6	0.2	0.2	02.6
chair	0.3	9.5	9.8	10.0	10.1
cow	1.2	9.9	12.4	12.2	14.9
diningtable	1.4	2.8	4.8	7.0	09.7
dog	9.4	0.5	0.7	0.6	01.8
horse	19.2	15.4	17.2	24.8	28.1
motorbike	12.0	14.8	17.6	17.4	22.6
person	5.8	8.8	8.7	12.5	12.2
pottedplant	9.2	9.4	9.4	9.8	09.9
sheep	1.1	9.8	4.2	4.8	10.0
sofa	0.7	5.2	4.0	3.7	04.3
train	13.2	15.0	20.7	16.2	19.3
tvmonitor	22.9	23.5	24.0	25.7	26.1
mAP	9.2	11.2	11.5	12.3	14.6

Tab. 3.11: Performances en fonction du nombre d'exemples.

4

Localisation à deux niveaux

Sommaire

La méthode proposée dans le chapitre précédent a le mérite de la simplicité et de l'efficacité. L'efficacité provient de l'utilisation de méthodes de représentations HOG et BoW combinées à un classifieur SVM linéaire, qui ensembles permettent un coût algorithmique limité.

En revanche, les expérimentations conduites avec ce détecteur – et bien que les performances atteintes soient proches de celles de l'état de l'art – sont quelque peu décevantes. Certaines des fausses détections peuvent sembler faciles à rejeter pour un humain.

Il nous a semblé que la limitation principale provenait du classifieur SVM linéaire qui ne possède pas un pouvoir discriminant suffisant. En effet, les frontières linéaires (hyperplan dans l'espace de représentation) peuvent difficilement s'accommoder de problèmes difficiles tels que ceux traités dans cette thèse (grande variance intra-classe et une grande similarité inter-classes). Cependant l'utilisation d'un classifieur plus puissant n'est envisageable que si un mécanisme de cascade lui est associé. L'utilisation d'un tel classifieur de manière systématique pour chaque position de fenêtre serait prohibitive.

Pour ces raisons, nous proposons une méthode de localisation inspirée de la méthode présentée au chapitre précédent, mais qui intègre un mécanisme de cascade et un classifieur plus performant.

4.1 Description de l'approche

L'approche proposée combine deux classifieurs SVM, l'un linéaire et l'autre non linéaire, combinés au sein d'une cascade à deux étages.

Nous commençons par décrire le classifieur non linéaire puis présentons les mécanismes de la cascade.

4.1.1 Classifieur SVM non linéaire

Le classifieur utilisé dans notre approche est un SVM non linéaire. Ce choix est motivé par les excellents résultats obtenus par ce type de classifieurs dans la tâche de catégorisation d'images [58, 76, 88]. Les SVM non linéaires utilisent des noyaux non linéaires qui exploitent une technique appelée *kernel trick* pour travailler dans un nouvel espace de représentation où les données sont linéairement séparables. C'est dans cet espace, qui peut avoir un nombre de dimensions infini, que l'hyperplan séparant les exemples positifs des négatifs va être calculé. Transposé vers l'espace de représentation d'origine, cet hyperplan n'est plus linéaire et peut donc s'accommoder à la difficulté du problème tout en gardant l'aspect positif provenant du principe de la maximisation de marge.

Plus formellement, le classifieur utilisé est un SVM non linéaire avec un noyau *RBF* χ^2 (Chi carré) qui s'écrit :

$$K(x, v) = exp(-\gamma.d_{\chi^2}(x, v)) \tag{4.1}$$

où γ est un paramètre du noyau, et d_{χ^2} est la distance χ^2 définie par :

$$d_{\chi^2}(x, v) = \sum_{i=1}^{N} \frac{(x_i - v_i)^2}{|x_i + v_i|}. \tag{4.2}$$

La valeur de γ est définie par l'heuristique de Zhang *et al.* [88] comme suit :

$$\gamma = \frac{Nb^2}{\sum_{i=1}^{Nb} \sum_{j=1}^{Nb} d_{\chi^2}(x_i, x_j)}, \tag{4.3}$$

où Nb est le nombre d'exemples d'apprentissage et x_i le $i^{\text{ème}}$ exemple. Notons qu'un noyau différent est utilisé pour chacun des deux descripteurs (HOG et BoW). Ces deux noyaux sont alors combinés par multiplication. Le noyau final s'écrit donc :

$$K(x, v) = K_{HOG}(x_{HOG}, v_{HOG}) \times K_{BoW}(x_{BoW}, v_{BoW}). \tag{4.4}$$

Cette méthode permet de mieux combiner différents canaux d'information qu'une concaténation classique, car à chaque canal est associé un poids qui lui est propre.

Les exemples d'apprentissage utilisés pour entrainer le classifieur sont les mêmes que ceux utilisés pour entrainer le classifieur linéaire, auxquels sont ajoutés un grand nombre d'exemples négatifs extraits aléatoirement du fond. En effet, à défaut de pouvoir suivre la même procédure itérative d'ajout d'exemples difficiles (pour des raisons de temps de calcul), nous ajoutons des exemples du fond (ici 70 000 exemples) en espérant que l'ensemble final contiendra assez d'exemples pour refléter la difficulté du problème traité.

4.1.2 Architecture du système

Le désavantage majeur des SVM non linéaires est le temps de calcul requis pour la classification d'un vecteur. Si, comme expliqué dans le chapitre précédent (paragraphe 3.1.2), la fonction de classification des SVM linéaires peut être factorisée pour être réduite à un produit scalaire et une addition, ce n'est pas le cas des SVM non linéaires. La fonction de classification requiert dans ce cas le calcul des distances entre le vecteur à classer et tous les vecteurs supports. Dans notre cas, les classifieurs non linéaires contiennent chacun de l'ordre de 2000 vecteurs supports. Ceci implique qu'une opération de classification sera à peu près 2000 fois plus coûteuse avec un classifieur non linéaire qu'avec un classifieur linéaire. En pratique, l'utilisation des SVM non linéaires s'avère impossible à combiner avec l'approche des fenêtres glissantes sans utiliser de cascade.

Pour remédier à ce problème le nombre de fenêtre doit être réduit. La technique la plus utilisée dans ce cas est celle des cascades, qui permet de rejeter rapidement un grand nombre de faux positifs dès les premiers étages.

Nous avons opté ici pour une cascade à deux niveaux. Le classifieur du premier niveau n'est rien d'autre que le classifieur linéaire préalablement décrit. Le classifieur du deuxième niveau est le SVM non linéaire. La figure 4.1 donne une vue globale du système ainsi obtenu.

Bien que simple, cette méthode permet d'allier la rapidité du classifieur linéaire et les excellentes performances du classifieur non linéaire. En effet, le classifieur linéaire (ou *classifieur de filtrage*) s'occupe de rejeter la plupart des faux positifs tandis que les quelques fenêtres qui restent seront traitées par le classifieur non linéaire (ou *classifieur de scoring*) qui sera chargé de leur affecter une note de confiance.

Cette cascade minimaliste nous permet d'éviter les cascades plus complexes qui, rappelons le, ont de multiples limitations comme la lenteur de l'entrainement (qui peut durer des semaines entières [81]), ou encore le choix des taux de faux positifs et de détection à chaque étage qui sont souvent empiriques.

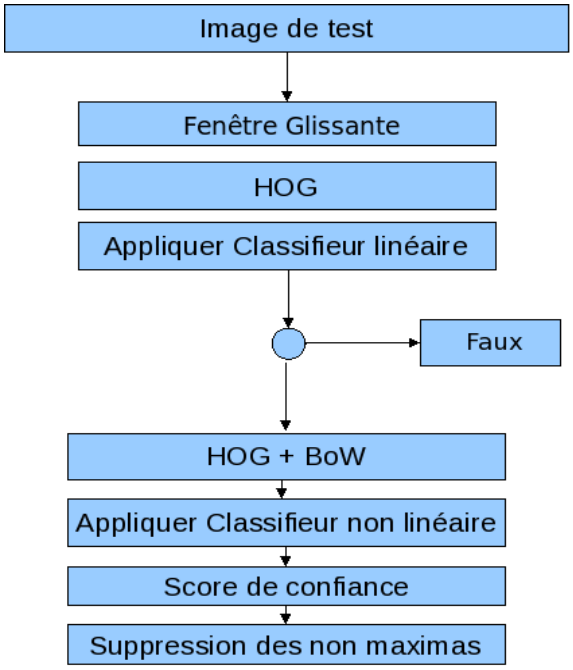


Fig. 4.1: Vue globale de l'approche à deux étages

Notons que pour le classifieur de filtrage, nous n'utilisons que le descripteur HOG. En effet, le but de ce classifieur n'est pas d'obtenir les meilleures performances possibles mais de rejeter le plus de fenêtres négatives possible tout en gardant les positives. Comme le descripteur HOG est beaucoup plus rapide que le descripteur BoW, cela nous permet d'économiser un temps de calcul considérable. Nous montrerons dans la section expérimentale ci-après le bien-fondé de ce choix.

4.2 Résultats expérimentaux

Dans cette section nous présentons des résultats expérimentaux obtenus avec la cascade à deux étages décrite plus haut. Nous commençons par une présentation des résultats globaux obtenus en appliquant cette procédure. Puis nous présentons des statistiques sur ces résultats afin d'apporter au lecteur une meilleure compréhension des raisons des réussites ou des échecs de notre algorithme.

La base de données d'images sur laquelle nous effectuons toutes nos expérimentations est la base PASCAL 2007 (pour plus de détails sur cette base, voir 2.2.1).

4.2.1 Résultats globaux

La table 4.1 montre le gain obtenu grâce à l'introduction du classifieur non linéaire (second niveau de la cascade). Cette table présente les résultats obtenus en utilisant le classifieur de filtrage (premier niveau) seul et après introduction du second niveau. Dans les deux cas, nous donnons les résultats obtenus avec les 3 différentes combinaisons de descripteurs (HOG, BoW et la combinaison des deux). Pour l'étape de filtrage, comme nous l'avons dit précédemment, nous utilisons uniquement le descripteur HOG et nous gardons pour chaque image les 200 meilleures fenêtres (celles ayant le score le plus haut).

	Linéaire			χ^2		
	HOG	BoW	HOG+BoW	HOG	BoW	HOG+BoW
plane	10.0	16.9	22.4	18.4	29.8	33.8
bike	27.8	21.2	30.5	39.5	33.3	43.0
bird	04.7	04.9	03.3	09.8	11.1	09.7
boat	00.6	04.8	01.8	02.0	04.2	09.6
bottle	11.4	07.3	11.2	18.2	09.5	18.7
bus	31.7	25.2	26.4	42.2	39.7	41.9
car	33.9	28.4	36.7	47.5	42.3	50.4
cat	02.6	06.9	06.0	02.5	14.4	15.0
chair	10.1	09.8	11.1	13.6	12.7	14.6
cow	14.9	10.3	14.3	22.1	20.4	23.9
table	09.7	06.7	10.9	10.5	13.3	15.1
dog	01.8	06.9	07.6	10.7	15.5	15.4
horse	28.1	30.5	33.8	43.5	40.5	48.2
moto	22.6	26.6	27.2	34.6	37.6	41.7
person	12.2	13.1	14.7	14.5	16.8	20.2
plant	09.9	09.4	09.8	11.7	11.4	16.1
sheep	10.0	12.5	15.1	12.7	19.8	21.2
sofa	04.3	12.1	14.7	14.2	18.8	20.3
train	19.3	17.0	22.4	31.8	34.4	29.1
tv	26.1	28.5	32.2	37.2	35.6	38.2
mAP	14.6	15.0	17.6	21.9	23.1	26.3

Tab. 4.1: Résultats globaux de localisation incluant les résultats avec et sans le classifieur non linéaire, pour toutes les combinaisons de descripteurs.

Le gain obtenu par l'introduction du classifieur non linéaire est très important : +7% pour le HOG, +8% pour le BoW et +9% pour la combinaison des deux descripteurs. Pour quelques classes le gain avoisine les 15% comme pour les chevaux (+14.4%), les motos (+14.5%) ou encore pour les voitures (+13.7%). Les meilleures

performances sont obtenues avec la cascade à deux étages utilisant la combinaison des HOG et des BoW (mAP de 26.3%). En utilisant une seule de ces représentations, les performances décroissent de plus 3%.

Notons aussi que, même si la combinaison des descripteurs HOG et des BoW permet d'apporter un gain avec le classifieur linéaire, celle-ci n'améliorera pas la qualité de filtrage du classifieur. En effet, l'objectif du classifieur de filtrage est de fournir une petite liste de fenêtres qui contient le plus de positifs possible. Pour évaluer un classifieur de filtrage, nous pouvons donc calculer le rappel maximum qu'il est possible d'obtenir lorsque seules sont gardées les fenêtres à plus haut score. Nous présentons dans la figure 4.2 la courbe du rappel maximum en fonction du nombre de fenêtres retournées par le classifieur de filtrage, et ce pour chacune des 3 combinaisons de descripteurs (HOG, BoW et la combinaison des deux). Nous remarquons que les trois courbes sont très similaires ce qui amène à conclure que le descripteur le moins coûteux (à savoir le HOG) est suffisant pour effectuer la première passe de filtrage.

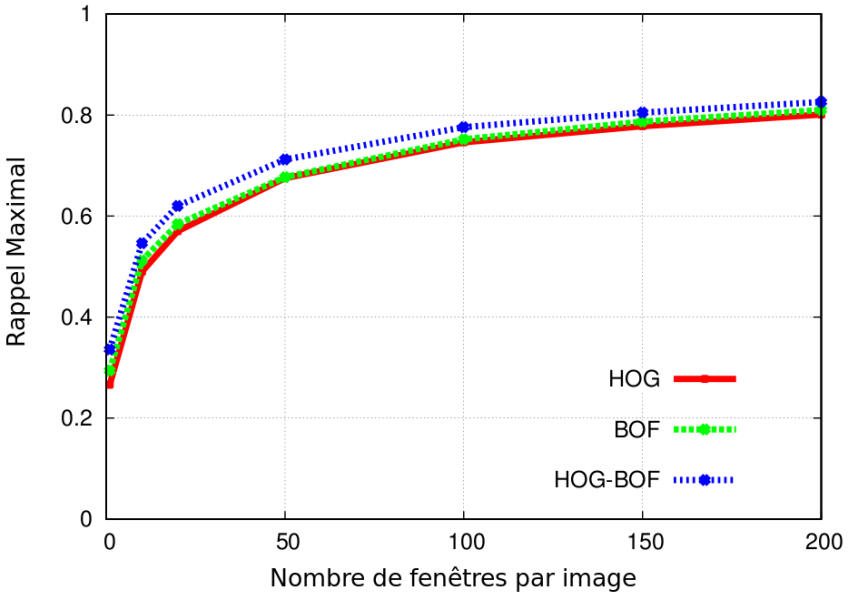


Fig. 4.2: Rappel moyen maximal en fonction du nombre d'images retournées par la phase de filtrage

Cette observation est également confirmée par la figure 4.3 qui, couplée à la figure 4.2, montre qu'à partir d'un certain taux de rappel (ici 0.6%) les exemples retournés sont trop difficiles et par conséquent, noyés dans les faux positifs qui deviennent de plus en plus nombreux.

Cette courbe montre aussi l'efficacité du classifieur de filtrage. En effet, à partir des quelques 65 000 fenêtres données par la procédure des fenêtres glissantes, une cin-

quantaine est suffisante pour obtenir une performance quasi-optimale (l'utilisation de 200 fenêtres au lieu de 50 donne un gain de 0.5% en termes de mAP).

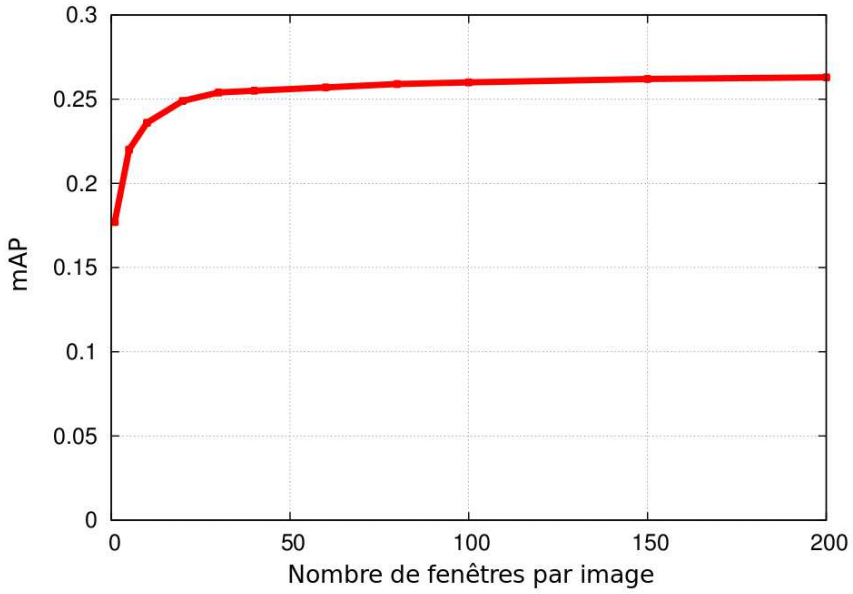


Fig. 4.3: mAP obtenu en fonction du nombre de fenêtres retournées par la phase de filtrage.

Notre détecteur final donne d'excellents résultats sur la base PASCAL 2007. Nous comparons dans la table 4.2 les résultats obtenus par ce détecteur (dénoté cascade) à ceux obtenus durant le PASCAL VOC challenge 2007 [19] ainsi qu'à l'approche récente de Felzenszwalb *et al.* [23] [1].

Comparée aux méthodes participantes au challenge, notre méthode obtient des performances globalement supérieures : meilleures performances sur 13 classes parmi 20 et 9.3% d'amélioration en termes de mAP par rapport à la meilleure approche (participant pour toutes les classes).

Comparée à l'approche de Felzenszwalb et al., notre approche obtient un mAP équivalent mais obtient la meilleure performance sur 12 classes parmi 20. Notons que les méthodes *INRIA_Norm.* et *INRIA_PlsCls* sont les méthodes avec lesquelles nous avons participé au PASCAL VOC challenge 2007 et par lesquelles nous avons obtenu la mention honorable. La méthode INRIA_Norm. correspond à une version moins élaborée du détecteur linéaire avec le descripteur HOG. La méthode INRIA_PlsCls, quant à elle, ajoute à la méthode INRIA_Norm. des informations émanant du contexte d'une façon similaire à celle décrite dans le chapitre 5.

[1] Nous présentons les résultats tels que publiés sur leur site internet. Ces résultats sont ceux obtenus sans utilisation d'information de contexte. Nous nous comparerons leur approche combinant l'information du contexte plus tard dans le chapitre 5.

	Résultats du challenge									Après challenge	
Classe	Darmst.	INRIA_Norm.	INRIA_PlsCls	IRISA	MPI_Cent.	MPI_ESS.	Oxford	TKK	UoCTTI	UoCTTIUCI	Cascade
plane	-	09.2	13.6	-	06.0	15.2	26.2	18.6	20.6	27.8	33.8
bike	-	24.6	28.7	28.1	11.0	15.7	40.9	07.8	36.9	56.1	43.0
bird	-	01.2	04.1	-	02.8	09.8	-	04.3	09.3	01.4	09.7
boat	-	00.2	02.5	-	03.1	01.6	-	07.2	09.4	14.6	09.6
bottle	-	06.8	07.7	-	00.0	00.1	-	00.2	21.4	25.7	18.7
bus	-	19.7	27.9	-	16.4	18.6	39.3	11.6	23.2	38.3	41.9
car	30.1	26.5	29.4	31.8	17.2	12.0	43.2	18.4	34.6	47.0	50.4
cat	-	01.8	13.2	02.6	20.8	24.0	-	05.0	09.8	15.1	15.0
chair	-	09.7	10.6	09.7	00.2	00.7	-	02.8	12.8	16.4	14.6
cow	-	03.9	12.7	11.9	04.4	06.1	-	10.0	14.0	16.7	23.9
table	-	01.7	06.7	-	04.9	09.8	-	08.6	00.2	23.1	15.1
dog	-	01.6	07.1	-	14.1	16.2	-	12.6	02.3	11.0	15.4
horse	-	22.5	33.5	28.9	19.8	03.4	-	18.6	18.2	44.4	48.2
moto	-	15.3	24.9	22.7	17.0	20.8	37.5	13.5	27.6	37.3	41.7
person	-	12.1	09.2	22.1	09.1	11.7	-	06.1	21.3	35.5	20.2
plant	-	09.3	07.2	-	00.4	00.2	-	01.9	12.0	14.0	16.1
sheep	-	00.2	01.1	17.5	09.1	04.6	-	03.6	14.3	16.9	21.2
sofa	-	10.2	09.2	-	03.4	14.7	-	05.8	12.7	19.3	20.3
train	-	15.7	24.2	-	23.7	11.0	33.4	06.7	13.4	31.9	29.1
tv	-	24.2	27.5	25.3	05.1	05.4	-	09.0	28.9	37.3	38.2
mAP	-	10.8	15.1	-	09.4	10.1	-	08.6	17.1	26.4	26.3

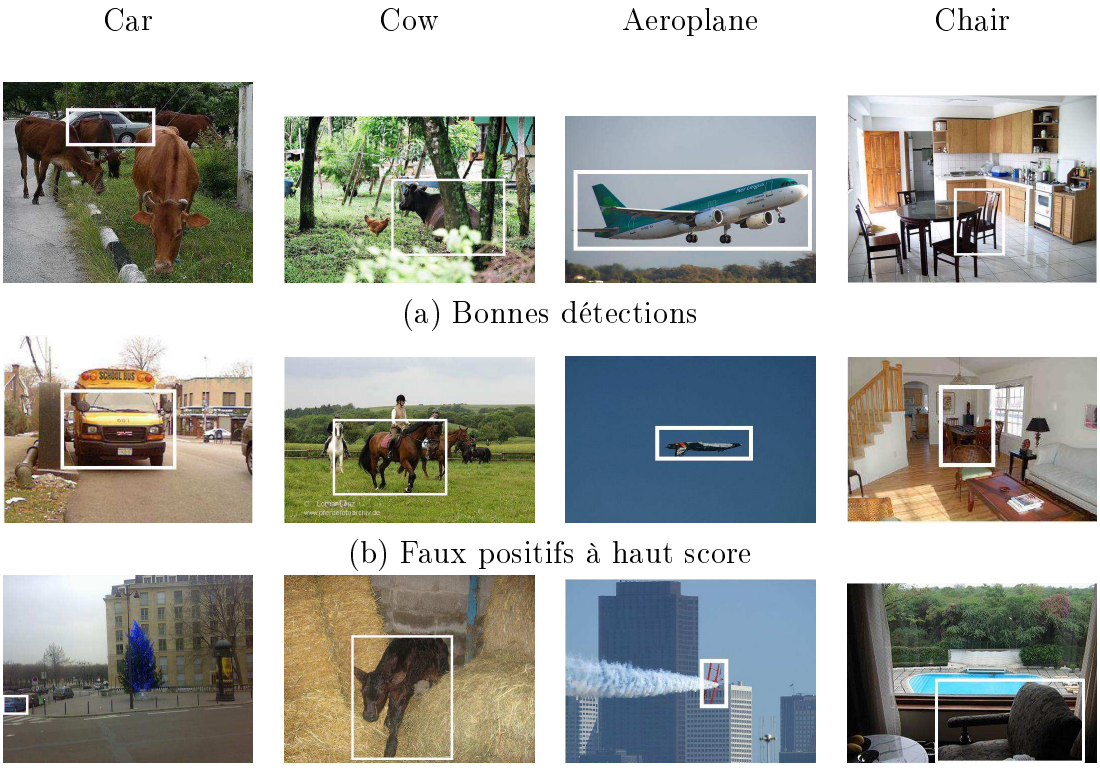
Tab. 4.2: Comparaison avec l'état de l'art : notre méthode obtient des résultats nettement meilleurs que les méthodes ayant participé au PASCAL VOC challenge 2007, et des performances comparables à celles récemment publiées.

Nous présentons dans la figure 4.4 des résultats qualitatifs de notre détecteur[2] qui donne une meilleure idée sur la difficulté de la tâche et les erreurs récurrentes. La première ligne montre des bonnes détections à haut score, la deuxième montre des exemples de faux positifs à haut score et la troisième et dernière ligne montre des exemples de faux négatifs (objets que notre algorithme n'a pas réussi à détecter). Nous remarquons qu'en ce qui concerne les faux positifs, les principales sources d'erreurs sont les confusions avec d'autres classes similaires (1er et 2ème exemples de faux positifs) et les détections imprécises (3ème exemple). Pour ce qui est des faux négatifs, les sources de problèmes sont la petite taille des objets (1er exemple), les points de vue ou poses non usuelles (pas ou peu présentes dans l'ensemble d'apprentissage) (2ème et 3ème) ou encore les objets tronqués (1er et 4ème).

4.2.2 Statistiques sur les résultats

Pour mieux interpréter les résultats de détection présentés ci-dessus, nous donnons dans ce paragraphe des statistiques qui permettent une meilleure compréhension des forces et des faiblesses de notre détecteur.

[2]Plus de résultats peuvent être retrouvés ici : http://lear.inrialpes.fr/people/harzallah/explore/



(a) Bonnes détections

(b) Faux positifs à haut score

(c) Faux négatifs

Fig. 4.4: Exemples de détections pour les classes car, cow, aeroplane et chair. (a) Exemples d'objets correctement détectés. (b)Exemples de faux positifs ayant un haut score. (c) Exemples d'objets que notre approche n'est pas capable de détecter.

Nous commençons par mesurer l'impact des limites décrites dans le paragraphe précédent. Pour ce faire, nous présentons :
- Les performances en fonction de la taille des objets
- Les performances en fonction de la précision de la fenêtre
- Des statistiques sur la confusion inter-classes
- Une mesure de l'influence des objets tronqués

Enfin, nous nous intéressons à l'effet de la détection multi-vue et à l'effet de la taille de l'ensemble d'apprentissage.

Performances en fonction de la taille

Nous avons observé sur les résultats qualitatifs, présentés dans la section précédente, que la taille des objets est la cause de la majorité des faux négatifs. Nous évaluons ici quantitativement l'influence de ce paramètre sur la performance de notre détecteur.

Pour ce faire, nous présentons dans la figure 4.5 la mAP (sur toutes les classes) obtenue pour différents intervalles de tailles d'objets. Notons que nous ne réap-

prenons pas notre détecteur mais que nous limitons l'ensemble des objets de test à ceux appartenant à l'intervalle choisi. Nous définissons la taille ici comme la moyenne de la largeur et de la hauteur du rectangle englobant l'objet.

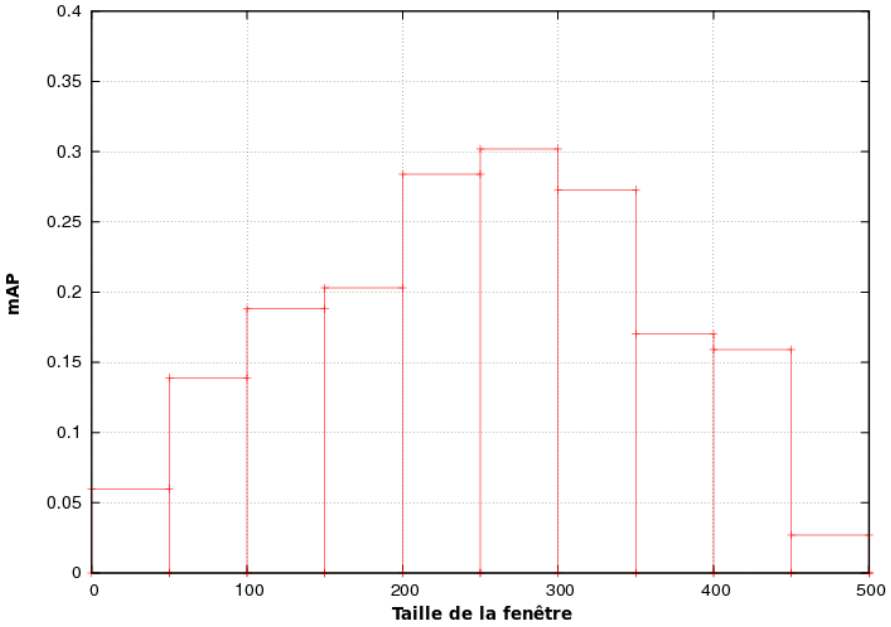


Fig. 4.5: mAP en fonction de la taille des objets

Nous observons d'abord que les meilleures performances sont obtenues pour des objets ayant une taille entre 200 et 350 pixels. En dessous de 200 pixels, notre algorithme obtient des performances acceptables. Sous le seuil de 50 pixels, les performances obtenues sont mauvaises. Ceci s'explique par le peu d'information que contiennent ces fenêtres, ne permettant pas à l'algorithme de décider de la présence d'un objet ou non. D'un autre coté, au delà de 350 pixels les performances baissent et de très mauvaises performances sont obtenues pour des fenêtres ayant une taille entre 450 et 500 pixels. Ceci peut paraitre surprenant, mais l'explication est qu'à partir d'une certaine taille les objets sont très souvent tronqués et sont donc visuellement loin des exemples utilisés pour l'apprentissage.

Effet de la précision de localisation

Une des causes des faux positifs est l'imprécision des détections. En effet, quelques détections, bien que bien situées sur l'objet, ne vérifient pas le critère d'appariement avec la vérité terrain (indice de Jaccard $2.3 < 0.5$) et sont donc considérées comme des faux positifs.

Nous évaluons dans ce paragraphe l'influence de ce critère sur les performances obtenues. Nous rapportons dans la figure 4.6 la courbe représentant la mAP en

fonction du seuil d'acceptation pour l'indice de Jaccard. Nous présentons aussi dans la figure 4.7 l'histogramme de la précision des détections.

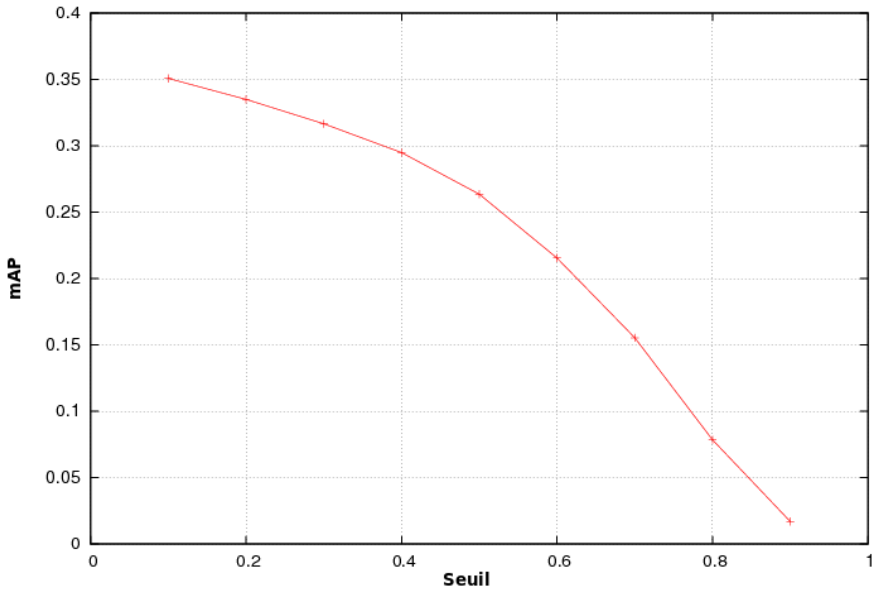


Fig. 4.6: mAP en fonction du seuil d'acceptation pour l'indice de Jaccard.

Nous observons qu'avec un critère plus tolérant, nous obtenons une amélioration notable de mAP. Par exemple, pour un seuil d'acceptation fixé à 0.3, on obtiendrait une amélioration de 5% en termes de mAP. Ceci s'explique par le fait que plus de 15% des détections ont un taux de chevauchement entre 0.3 et 0.5. Ceci est causé par la grande variabilité intra-classe qui peut exister ainsi que par le fait qu'un seul ratio d'aspect (le ratio moyen) est utilisé pour chaque détecteur de point de vue.

Influence des instances tronquées

La détection des objets tronqués reste un des grands challenges de la tâche de localisation d'objets. Peu d'études se sont penchées sur le sujet [83], mais il reste évident que les modèles rigides, tel que celui utilisé ici, sont moins adaptés que les modèles à parties ou encore les modèles à forme implicite (Implicit Shape Model).

Nous mesurons dans ce paragraphe l'influence des exemples tronqués sur les performances de notre approche. La table 4.3 présente l'AP obtenu en ignorant les exemples marqués comme tronqués (les faux négatifs ainsi que les bonnes détections sont ignorés). Rappelons que les annotations fournies par la base PASCAL incluent des indicateurs pour les exemples tronqués et difficiles. Nous observons qu'un gain significatif de 6% en termes de mAP est obtenu. Plus intéressant encore, pour

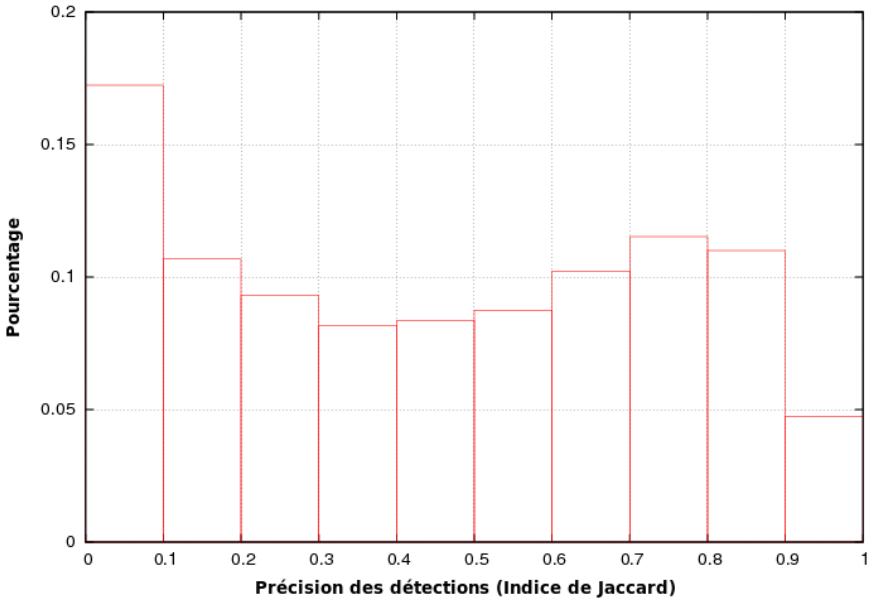


Fig. 4.7: Histogramme de l'indice de Jaccard entre les détections et la vérité terrain

Fig. 4.8: Exemples d'images de voitures tronquées de la base PASCAL 2007

quelques classes, le gain obtenu avoisine les 20% (19% pour les voitures, 16% pour les bus).

	mAP	bicycle	bottle	bus	car
Tronqués ignorés	32.0	54.4	22.4	57.7	70.6
Base entière	26.3	43.0	18.7	41.9	51.1

Tab. 4.3: Résultats obtenus en incluant et ignorant les exemples tronqués.

En effet, les objets tronqués sont particulièrement difficiles à détecter (voir Figure 4.8). Premièrement ils peuvent représenter une partie minuscule de l'objet réel (par exemple, morceau d'une roue de voiture) et par conséquent être très différents les uns des autres et très différents des exemples d'apprentissage. Deuxièmement, ils peuvent avoir des ratios d'aspect très improbable pour des objets entiers. Finalement, nous n'utilisons pas les exemples tronqués lors de l'apprentissage car ceux-ci n'ont pas une apparence consistante. L'utilisation de ces exemples lors de l'apprentissage peut fausser le détecteur et faire en sorte que plusieurs parties d'un même objet soient détectées. Ceci a été vérifié expérimentalement. En effet, nous avons essayé d'inclure les objets tronqués dans l'ensemble d'apprentissage et avons observé que cela engendre une baisse de performances.

Confusion entre les classes

Nous examinons ici une autre source d'erreurs de notre algorithme à savoir les confusions entre objets. En effet nous avons observés que de nombreuses fausses détections sont causées non pas par des détections sur le fond mais plutôt par la détection d'autres objets. Ceci est compréhensible sachant que plusieurs objets de la base se ressemblent fortement.

Pour mesurer quantitativement l'effet de la confusion, nous présentons dans la table 4.4 la matrice de confusion entre objets. Nous ne considérons ici que les détections obtenues pour un rappel de 30%.

La première remarque émanant de cette table est que la plupart des faux positifs proviennent tout de même sur le fond. Ceci dit, pour quelques classes, une bonne partie des faux positifs sont dues à la confusion. Par exemples, nous pouvons remarquer que 9% des détections de chats apparaissent sur des chiens, 9% des détections de trains sont des bus, ou que 11.8% des détections de vaches sont des chevaux et 13.4% des moutons. On peut aussi remarquer que les vélos et les motos sont souvent confondus.

D'une manière générale, on peut dire qu'il y a deux groupes d'objets souvent confondus entre eux : les animaux et les véhicules. Les véhicules peuvent eux même être

		Classe détecté									
		plane	bike	moto	train	bus	car	tv	boat	bottle	chair
Détecteur de	plane	56.7	0.7	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0
	bike	0.0	85.6	2.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	moto	0.0	12.1	73.5	0.8	0.0	0.8	0.0	0.0	0.0	0.0
	train	0.0	0.0	0.0	50.3	9.0	2.4	0.0	0.6	0.0	0.0
	bus	1.3	0.0	0.0	2.6	80.8	1.3	0.0	0.0	0.0	0.0
	car	0.0	0.0	0.0	0.0	1.1	97.3	0.0	0.0	0.0	0.0
	tv	0.0	0.0	0.0	0.0	0.0	0.0	63.0	0.0	0.0	0.0
	cat	0.1	0.1	1.3	0.1	0.0	0.0	0.1	0.0	0.0	0.3
	boat	0.2	0.1	0.2	0.4	0.3	1.4	0.1	0.3	0.0	0.1
	bottle	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.1	11.9	0.1
	chair	0.0	0.0	0.1	0.2	0.3	0.3	0.5	0.0	0.2	4.7
	bird	0.1	0.3	0.3	0.1	0.0	0.2	0.1	0.1	0.1	0.2
	cow	0.0	0.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.4
	dog	0.1	0.9	1.0	0.0	0.0	0.2	0.0	0.1	0.1	0.4
	horse	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sheep	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	table	0.2	0.2	0.1	0.4	0.1	0.9	0.1	0.0	0.0	0.9
	plant	0.0	0.3	0.2	0.0	0.0	0.2	0.1	0.1	0.2	0.4
	person	0.1	0.2	0.5	0.0	0.0	0.1	0.0	0.1	0.4	0.2
	sofa	0.0	0.3	0.0	0.0	0.0	3.8	0.3	0.3	0.0	1.6

		Classe détecté										
		cat	bird	cow	dog	horse	sheep	table	plant	person	sofa	Fond
Détecteur de	plane	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	41.3
	bike	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.4	0.0	8.5
	moto	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.5	0.0	8.3
	train	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	37.7
	bus	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	14.1
	car	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.6
	tv	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	37.0
	boat	0.1	0.1	0.1	0.1	0.1	0.0	0.1	0.1	0.5	0.1	95.3
	bottle	0.0	0.0	0.0	0.1	0.1	0.0	0.0	0.3	3.0	0.0	84.0
	chair	0.1	0.1	0.3	0.1	0.3	0.1	0.1	0.1	1.8	0.3	90.0
	cat	12.3	1.7	0.2	9.2	0.7	1.0	0.3	1.0	4.7	0.9	65.2
	bird	0.3	0.7	0.6	0.7	0.7	0.4	0.0	0.2	3.7	0.0	91.1
	cow	0.0	1.2	29.7	7.3	11.8	13.4	0.0	0.0	0.8	0.0	34.6
	dog	2.8	1.7	4.1	8.8	4.6	2.5	0.1	0.2	7.7	0.2	63.7
	horse	0.0	0.0	1.7	2.6	89.7	1.7	0.0	0.0	0.0	0.0	4.3
	sheep	1.0	1.6	7.1	3.0	3.2	14.5	0.0	0.2	1.6	0.2	66.7
	table	0.4	0.1	0.0	0.1	0.1	0.0	2.5	0.0	0.9	0.6	91.9
	plant	0.3	0.1	0.1	0.3	0.2	0.0	0.1	3.9	2.6	0.1	90.7
	person	0.2	0.4	0.4	0.5	0.8	0.3	0.0	0.2	21.7	0.0	73.3
	sofa	1.9	0.0	0.0	1.9	0.0	0.0	3.2	0.6	1.6	22.8	61.2

Tab. 4.4: Matrice de confusion pour un taux de rappel de 30% : chaque ligne représente les résultats d'un détecteur et montre le taux de positifs, de confusion et de faux positifs parmi ses détections.

divisés en deux groupes, les vélos et les motos d'une part, et les trains, les bus et les voitures de l'autre part.

Remarquons aussi qu'il existe une confusion considérable entre le groupe vélos et motos avec les personnes. Ceci est dû au fait que ces objets apparaissent souvent ensemble (personne sur vélo ou moto). Finalement, remarquons que les objets à faible précision sont souvent confondus avec les personnes. Ceci est dû au grand nombre de personnes dans la base de données.

Détection multi-vues

Dans ce paragraphe, nous nous penchons sur l'effet de l'utilisation de détecteurs multiples (un par vue) sur la qualité des résultats. Pour ce faire, nous comparons les résultats obtenus avec cette méthode à ceux obtenus en construisant un seul détecteur pour toutes les vues. Pour donner une meilleure idée de l'importance de chaque détecteur de vue, nous présentons aussi les résultats obtenus par chacun des détecteurs de point vue séparément. Ces résultats sont donnés dans le tableau 4.5. Rappelons que pour chacune des images de la base PASCAL, l'information de point de vue est donnée (5 point de vue existent : "Left", "Right", "Rear", "Frontal" et "Unspecified") et que nous construisons 3 détecteur par objet, un pour la vue "Unspecified", un autre regroupant les vues "Left" et "Right" et un dernier regroupant les vues "Rear" et "Frontal".

	Toutes classes confondues	bicycle	car	tvmonitor
Left+Right	12.2	31.1	24.7	09.7
Rear+Front	10.3	10.2	28.3	34.6
Unspecified	15.2	15.9	42.9	29.6
Mixés	26.3	43.0	51.1	38.2
Général	23.6	27.1	48.8	41.3

Tab. 4.5: Évaluation de l'effet du détecteur multi-vues : à quelques exceptions près la séparation des vues donne toujours de meilleurs résultats.

Nous observons que la séparation des vues permet d'obtenir de meilleures performances (+3% en termes de mAP). Ce gain est plus ou moins important selon les classes : pour les vélos par exemple, le gain est de 16%. Par contre, pour quelques classes comme pour les TV, la tendance est inversée et le détecteur général donne de meilleurs résultats.

Ces résultats permettent aussi de mettre en avant la difficulté des vues. Pour les vélos par exemple, le détecteur de profil donne de meilleurs résultats que le détecteur général. Par contre le détecteur avant/arrière donne des résultats médiocres. Ceci s'explique par le peu d'informations discriminantes pour les vues avant/arrière comparées à la vue de profil. Le contraire se produit pour les TV où la vue avant/arrière est clairement plus facile que celle de profil.

Effet du nombre d'exemples d'apprentissage

Nous passons aux dernières expérimentations, qui se penchent sur l'effet du nombre d'exemples d'apprentissage sur la qualité des détecteurs. En effet, dans un contexte

où la performance est la motivation principale, on pourrait se demander si l'augmentation du nombre d'exemples d'apprentissage ne donnerait pas les mêmes gains qu'une méthode nettement plus sophistiqué.

Pour évaluer l'effet du nombre d'exemples, nous apprenons 4 ensembles de détecteurs avec des sous-ensembles des images d'apprentissage en plus des détecteurs originaux utilisant toutes les images d'apprentissage. Nous rapportons dans la table 4.6 les résultats obtenus par chacun de ces détecteurs.

	10%	25%	50%	75%	100%
aeroplane	21.2	26.9	29.4	31.1	33.8
bicycle	32.1	37.5	38.0	41.5	43.0
bird	3.1	10.0	6.0	8.4	09.7
boat	4.8	9.4	5.0	2.4	09.6
bottle	11.0	13.2	16.4	18.0	18.7
bus	32.1	38.1	42.5	42.7	41.9
car	40.4	45.1	47.4	49.5	50.4
cat	6.4	6.3	13.4	11.8	15.0
chair	10.6	12.4	13.7	14.6	14.6
cow	5.9	16.7	16.6	24.2	23.9
diningtable	10.5	10.8	13.4	15.2	15.1
dog	11.8	13.1	13.8	14.8	15.4
horse	36.5	41.1	43.4	47.0	48.2
motorbike	28.6	33.1	36.2	40.4	41.7
person	14.0	16.7	18.2	19.0	20.2
pottedplant	10.0	11.3	12.3	13.7	16.1
sheep	15.8	19.0	19.6	21.4	21.2
sofa	6.4	13.8	19.8	20.1	20.3
train	26.2	28.5	33.0	35.8	29.1
tvmonitor	31.0	36.0	36.2	37.0	38.2
Mean	17.9	22.0	23.7	25.4	26.3

Tab. 4.6: Effet de la taille de l'ensemble d'apprentissage : l'utilisation de moins d'exemples peut baisser considérablement les performances

Nous remarquons qu'en termes de mAP, l'augmentation du nombre d'exemples d'apprentissage induit nécessairement un gain en performances. Plus intéressant encore, la performance ne semble pas saturer avec 100% des exemples d'apprentissage. Ceci implique que l'agrandissement de cet ensemble apporterait un gain considérable. Notons tout de même que pour certaines classes (train, chaise), peu ou pas d'améliorations sont observées à partir d'un certain nombre d'exemples.

Idéalement, les expériences avec les sous-ensembles d'apprentissage devraient être répétées plusieurs fois en travaillant à chaque fois avec un nouvel ensemble choisi aléatoirement. Cependant, en raison du temps de calcul requis, ces expériences ne sont effectuées qu'une seule fois. Même si ceci n'affecte que faiblement les résultats en termes de mAP, les résultats par classe peuvent quant à eux souffrir de la variance des exemples d'apprentissage. Ceci est observé par exemple pour la classe de trains où les performances obtenues avec 50% et 70% des exemples sont supérieures à celles obtenues avec tous les exemples.

4.2.3 Temps de calcul

Nous présentons dans ce paragraphe des statistiques sur les temps de calcul nécessaires à notre algorithme pour s'exécuter. Les temps présentés ici correspondent à des calculs effectués sur une machine Pentium 4 à 3Ghz avec une image de taille 500x375 pixels (taille type pour les images de la base PASCAL 2007).

La table 4.7 montre les temps de calcul nécessaire à la phase de filtrage puis à la phase de scoring. Nous remarquons que la phase la plus coûteuse est celle de scoring (deuxième étage de la cascade) même si elle ne traite que 200 fenêtres par images. Le temps de calcul par image total requis pour obtenir les résultats présentés est de 1 minute 12 sec. Ce temps peut être facilement réduit à 27 secondes, sans perte notable de performances, en ne traitant que 50 fenêtres par le classifieur de scoring. Notons tout de même que notre implémentation n'a pas été optimisée et que ces temps peuvent donc certainement être réduits.

	Méthode	Nombre de fenêtre considérée	Temps de calcul	
			Total	Par fenêtre
Filtrage	HOG	65K	12s	≈ 2ms
	BoW	65K	26s	≈ 4ms
	HOG + BoW	65K	38s	≈ 6ms
Scoring	HOG	200	20 sec	≈ 100ms
	BoW	200	40 sec	≈ 200ms
	HOG + BoW	200	60 sec	≈ 300ms

Tab. 4.7: Temps de calcul pour les différentes phases de notre approche de localisation.

4.3 Conclusion

Dans ce chapitre, nous avons proposé une nouvelle méthode de localisation d'objets dans les images. Cette méthode, composée d'une cascade à deux étages de classifieurs, allie rapidité et bonnes performances. Le premier étage de la cascade est un classifieur linéaire qui malgré sa rapidité et sa capacité à rejeter la plupart des faux positifs, ne permet pas de bien modéliser les objets. Le deuxième étage de la cascade est un classifieur non linéaire, très lent, mais qui donne d'excellents résultats de classification. Nous avons ensuite présenté de multiples résultats expérimentaux permettant de mieux comprendre les forces et les faiblesses de l'algorithme. Notons que nous avons remporté le PASCAL VOC challenge 2008 [35] avec une version légèrement amélioré de cette approche qui incorpore des informations contextuelle, obtenant les meilleures performances sur 11 classes parmi 20. Nous reviendrons plus en détail sur cette méthode dans le chapitre 5.

5

Vers une méthode combinant localisation d'objets et classification d'images.

Sommaire

Dans les chapitres précédents, nous nous sommes intéressés à la localisation des objets dans les images et avons développé un algorithme de localisation atteignant des performances au niveau de l'état de l'art. Dans ce chapitre, nous nous intéressons conjointement à une autre tâche importante de la vision par ordinateur, à savoir la classification d'images. Nous cherchons à mettre au point une méthode qui combine les approches de localisation présentées dans les chapitres précédents avec une méthode de classification d'images, pensant que les deux pourraient être complémentaires.

Dans la première section de ce chapitre, nous présentons les raisons qui peuvent motiver le développement d'une telle combinaison d'approches. Puis la section 5.2 présente en détail la méthode développée ainsi que le classifieur d'images utilisé. Finalement, dans la section 5.3, nous présentons les résultats expérimentaux obtenus

et nous les comparons à ceux de l'état de l'art, à la fois pour la tâche de localisation et pour la tâche de classification.

5.1 Motivation

La combinaison de la classification d'images et de la localisation d'objets a plusieurs motivations.

Avant de les exposer, rappelons ce que nous entendons par algorithme de classification d'images. Il s'agit d'un algorithme qui prend une image en entrée et donne en sortie la probabilité qu'un objet d'une classe d'intérêt soit présent dans l'image en question (ou un score représentant cette probabilité). Il répond par exemple à la question « il y a t-il une voiture dans cette image ?». Ces algorithmes, comme nous le verrons plus tard, utilisent une représentation globale de l'image (par exemple par sac-de-mots) et un classifieur entrainé de manière supervisée.

La motivation pour cette combinaison est qu'elle représente un moyen simple et peu coûteux d'incorporer des informations représentant le contexte général de l'image dans la tâche de localisation, et ce quelle que soient les méthodes de classification et de localisation utilisées. Les méthodes de localisation par fenêtre glissante prennent les décisions à partir d'une analyse très locale de l'image et ignorent le contenu global de l'image (contexte). Or, même si le contexte a été très peu exploité jusque là dans les approches de localisation, il peut jouer un rôle important et être une source d'indices précieux (voir figure 5.1 pour une illustration).

Notre intuition est qu'une telle combinaison doit permettre de minimiser le nombre de faux positifs apparaissant sur des images ne contenant pas les objets recherchés. Pour valider cette intuition et donner une meilleure idée sur le nombre et l'influence de ces faux positifs sur les performances du système, nous supposons disposer d'un classifieur d'images parfait. Nous mesurons alors les performances obtenues après élimination de ces faux positifs. Ces résultats sont présentés dans la table 5.1

Nous observons un gain important, avoisinant les 8% en termes de mAP. Ceci illustre l'intérêt que peut susciter une telle combinaison, sachant que les méthodes de classification d'images récentes donnent des résultats satisfaisants et peuvent désormais être considérées comme des technologies relativement matures.

Si la première motivation était liée à l'apport de la classification d'image à la localisation, l'inverse est également vrai et constitue une autre source de motivation. La combinaison permet en effet de prendre en compte la structure spatiale des objets dans la classification d'images, ce que la représentation par sac de mots ne permet pas de faire.

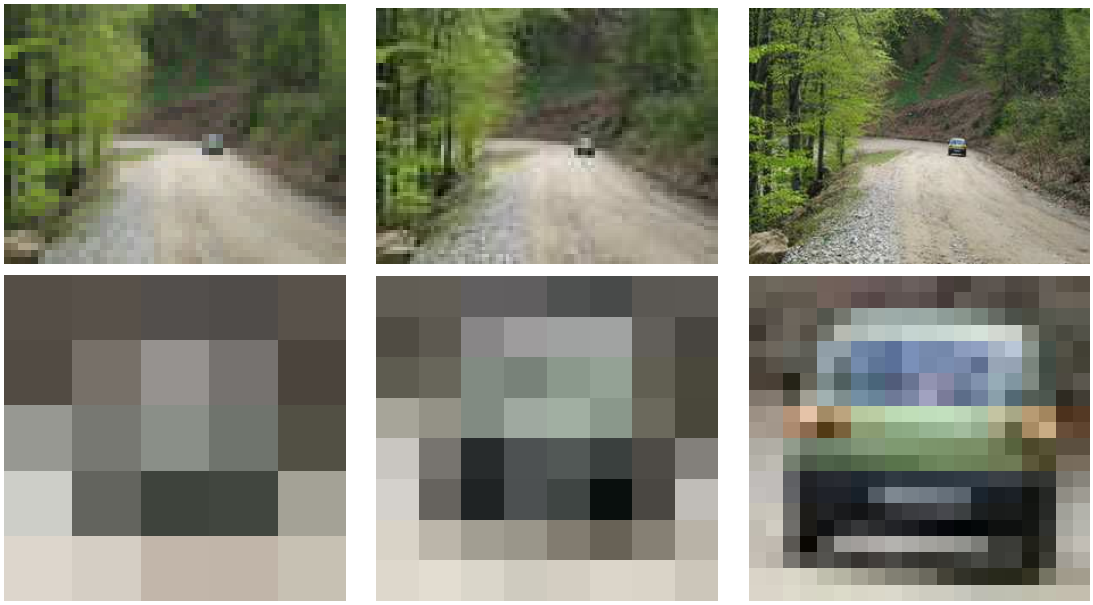


Fig. 5.1: Importance du contexte dans la détection : la même voiture est présente dans les trois images (à 3 résolutions différentes). Hors contexte (seconde ligne), seule la voiture de droite est facile à reconnaître pour l'œil humain. En revanche, en exploitant l'information du contexte, les 3 voitures deviennent beaucoup plus faciles à reconnaître.

Fig. 5.2: Importance de la structure géométrique dans la classification : l'image de droite est obtenue en brouillant l'image originale. L'image obtenue est très difficile à classer, et pourtant les deux images seraient décrites de la même façon par le modèle sac de mots.

Nous montrons dans la figure 5.2 deux images, celle de gauche est l'image d'un train clairement reconnaissable et celle de droite où la même image est brouillée en mélangeant l'ordre des blocs qui constituent l'image. Cette dernière est difficile

Classe	Détecteur seul	Détecteur + classifieur parfait
aeroplane	33.8	34.5
bicycle	43.0	50.5
bird	09.7	14.8
boat	09.6	09.6
bottle	18.7	33.1
bus	41.9	55.3
car	50.4	52.5
cat	15.0	25.7
chair	14.6	21.4
cow	23.9	47.3
diningtable	15.1	13.2
dog	15.4	22.1
horse	48.2	50.5
motorbike	41.7	51.9
person	20.2	18.4
pottedplant	16.1	16.2
sheep	21.2	40.0
sofa	20.3	33.6
train	29.1	45.0
tvmonitor	38.2	54.1
mAP	26.3	34.5

Tab. 5.1: Résultat de localisation obtenus en combinant la localisation par fenêtre glissante avec un classifieur d'image parfait.

à reconnaitre à l'œil, or la représentation par sac de mots des deux images serait pratiquement identique. Notons également que si un objet est petit ou s'il est entouré d'un contexte non usuel, il aura peu de poids dans la représentation globale de l'image et pour cette raison le classifieur aura du mal à faire une bonne prédiction. L'algorithme de localisation pourrait permettre dans ce cas d'améliorer la prédiction.

Finalement une troisième motivation est que les informations utilisées par le classifieur et le détecteur sont différentes. Cette hypothèse peut être vérifiée en observant qu'une image est relativement souvent classée différemment (avec une probabilité sensiblement différente) par le détecteur d'objets et le classifieur d'images. Nous avons cherché à quantifier ce phénomène. Pour ce faire, nous appliquons notre détecteur et transformons ses sorties de détection en scores de classification (comme présenté dans le paragraphe 5.2.4). Puis, nous utilisons le classifieur décrit dans le paragraphe 5.2.1. La figure 5.3 montre la densité de probabilité des vrais positifs (toutes classes de PASCAL confondus) en fonction de la probabilité donnée par le

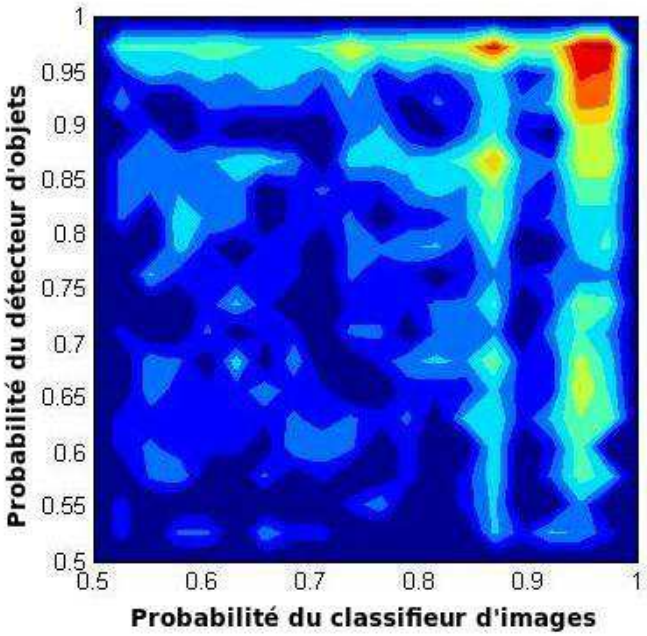


Fig. 5.3: Densité de probabilité des vrais positifs en fonction des probabilités données par le classifieur d'images et le détecteur d'objets. Les couleurs chaudes (resp. froides) correspondent à des probabilités fortes (resp. faibles). Ces résultats sont ceux obtenus sur la base PASCAL 2007.

détecteur et celle donnée par le classifieur. On observe un pic au point $(1, 1)$ où les deux modalités confirment la présence d'un objet, mais plus important encore, on observe qu'un nombre important de vrais positifs se situent sur les bords de la figure, là où une seule modalité en est certaine. Cette observation valide notre hypothèse. En effet, si un objet est masqué ou tronqué, le détecteur aura du mal à le trouver alors que, pour ce même objet, le classifieur aura plus de chance de le trouver en se basant sur d'autres informations (comme le contexte ou encore les parties visibles de l'objet). Inversement, si un objet est relativement petit et qu'il se situe dans un contexte non habituel, le détecteur n'aura pas du mal à le retrouver, contrairement au classifieur d'images. La figure 5.4 illustre ces deux cas.

5.2 Description de la méthode de combinaison

Nous présentons dans cette section la méthode proposée pour combiner la classification d'images et la localisation d'objets. Nous commençons d'abord par présenter le classifieur d'images utilisé, puis décrivons notre modèle de combinaison.

(a)

(b)

Fig. 5.4: Complémentarité de la classification et de la détection : (a) Les voitures sont détectées avec un score fort alors que la classification d'images donne un score faible. (b) Les voitures sont détectées avec un score faible, car partiellement visible, mais la classification d'image donne un excellent score.

5.2.1 Description du classifieur d'images

Dans le cas de combinaison de méthodes, nous pensons qu'une amélioration ne peut être considérée comme significative que si elle améliore les performances des meilleures méthodes de l'état de l'art. En effet, il est toujours plus facile d'obtenir une amélioration quand les résultats de départ sont moyens.

Pour cette raison, nous utilisons le classifieur *INRIA_flat* [58] qui a prouvé ses capacités durant les compétitions PASCAL VOC 2007 [19] et 2008 [20] où une version légèrement améliorée de celui-ci a gagné ces deux éditions de la compétition. La représentation d'images utilisée par le classifieur INRIA_flat est la représentation par « sac de mots ». Des points d'intérêt sont extraits par un détecteur de Harris-Laplace [60], le Laplacien [55], le Hessien [60] et Harris-Harris mais aussi via une extraction dense. Les patchs extraits sont décrits par les descripteurs SIFT [56] et Opponent-SIFT [76] qui est une version de SIFT prenant en considération la couleur. Chaque descripteur est ensuite quantifié avec l'algorithme *k-means*. Pour ce qui est de la constitution des sacs de mots, plusieurs grilles spatiales sont utilisées et un sac de mots est crée pour chaque élément de cette grille.

Les grilles spatiales utilisées sont la grille 1x1 qui correspond à l'image entière, la grille 2x2 qui correspond à couper l'image en quatre quarts et la grille 3x1 qui correspond à avoir 3 bande horizontales. Ce partitionnement de l'image permet de capturer une partie de la structure géométrique de l'image.

Pour ce qui est de la classification à proprement parler, INRIA_flat utilise un classifieur SVM. Le noyau utilisé est le noyau *RBF* avec une distance χ^2. Ce noyau assure la bonne combinaison des différentes sources d'information. En effet, avec toutes les

combinaisons des configurations de description, le classifieur INRIA_flat utilise en somme 30 canaux d'information ({5 Extraction} × {2 Descriptions} × {3 Grilles}). Le noyau s'écrit comme suit :

$$K(X_j, X_k) = \prod_{ch} \exp(-\gamma_{ch} D_{ch}(X_j, X_k)) \tag{5.1}$$

où D_{ch} est la distance du χ^2 pour le canal ch et γ_{ch} est une constante définit pour chaque canal selon l'heuristique de [88] comme suit :

$$\gamma_{ch} = \frac{Nb^2}{\sum_{i=1}^{Nb} \sum_{j=1}^{Nb} D_{ch}(X_i, X_j)}, \tag{5.2}$$

où Nb est le nombre d'exemples d'apprentissage et x_i le $i^{\text{ème}}$ exemple d'apprentissage.

5.2.2 Modèle de combinaison

Notre modèle de combinaison repose sur une idée très simple : toutes les instances d'objets ne sont pas nécessairement détectables par les deux modalités (à savoir le classifieur et le détecteur). A titre d'exemple, la probabilité donnée par le classifieur de voitures pour une image est la probabilité qu'une voiture existe dans cette image sachant qu'elle est détectable par ce même classifieur.

Dans le cas contraire, cette probabilité n'aura aucun sens.

Nous essayons ici de quantifier cette notion de détectabilité et proposons d'apprendre pour chaque classe d'objets et pour chaque modalité une probabilité de détectabilité. Plus formellement, notons $P(D_i)$ la probabilité qu'un objet soit détectable par le classifieur d'images et $P(D_w)$ la probabilité [1] qu'un objet détectable par l'algorithme de localisation par fenêtre glissante.

La probabilité donnée par le classifieur d'images est donc la probabilité $P(O|S_i, D_i)$, où S_i est le score obtenu par le classifieur. Ceci nous donne la probabilité d'apparition d'un objet dans une image sachant qu'il est détectable. De la même façon, la probabilité donnée par le détecteur d'objets est la probabilité $P(O|S_w, D_w)$.

La probabilité d'avoir un objet dans une image sachant le score de classification est donc :

$$P(O|S_i) = P(D_i)P(O|S_i, D_i) + P(\overline{D_i})P(O|S_i, \overline{D_i}) \tag{5.3}$$

[1] Dans notre notation, i est utilisée pour référer aux classifieurs d'images, et w aux détecteurs

et de la même manière, la probabilité d'avoir un objet dans une fenêtre sachant le score de détection est donnée par

$$P(O|S_w) = P(D_w)P(O|S_w, D_w) + P(\overline{D_w})P(O|S_w, \overline{D_w}) \tag{5.4}$$

où $P(O|S_i, \overline{D_i})$ (resp. $P(O|S_i, \overline{D_w})$) est la probabilité que l'objet soit présent quand il n'est pas détectable par le classifieur d'images (resp. détecteur d'objets).

La probabilité finale d'une fenêtre pour la tache de localisation est donné donc par

$$P(O|S_w, S_i) \propto P(O|S_i) \times P(O|S_w). \tag{5.5}$$

Pour la tâche de classification, la probabilité finale pour l'image est obtenue comme suit

$$P(O|S_{bw}, S_i) \propto P(O|S_i) \times P(O|S_{bw}), \tag{5.6}$$

où S_{bw} est le score de classification obtenu par détection (voir paragraphe 5.2.4).

Notons que lors de l'apprentissage, tous les objets sont considérés comme détectables et que les probabilités conditionnelles $P(O|S_i, \overline{D_i})$, $P(O|S_i, \overline{D_w})$ ainsi que les probabilités a priori $P(D_i)$ et $P(D_w)$ sont des constantes obtenus par validation croisée : nous sélectionnons les valeurs qui maximisent l'AP sur un ensemble de validation.

5.2.3 Calcul des probabilités à partir des scores de détection

Pour transformer les sorties d'un SVM en probabilités, la technique la plus couramment utilisée est celle de Platt [65]. Ces travaux ont montré que la distribution des probabilités en fonctions des sorties des SVM est sigmoïdale. Pour transformer les scores en probabilités, il suffit de trouver les paramètres A et B de la fonction sigmoïde :

$$Proba = \frac{1}{1 + \exp(-A * score + B)} \tag{5.7}$$

qui minimisent une certaine fonction de coût par rapport aux probabilités a posteriori obtenus sur un ensemble de validation.

Or, bien que nos scores de détection soient des sorties de SVM, la procédure de suppression de non maxima introduit de multiples modifications qui font que les valeurs des scores sont altérées et que leur distribution change (les détections chevauchantes sont fusionnées). En plus, la définition de ce qui est vrai et faux et donc les probabilités a posteriori changent aussi : deux fenêtres sur un même objet sont considérés toutes deux comme des positifs avant suppression de non maximas ce qui n'est plus

le cas après. Les sorties du détecteur ne sont donc pas comparables à celles des SVM et la distribution de probabilités en fonctions de ces scores n'a aucune garantie d'être sigmoïdale, et c'est en pratique ce qui arrive. Nous ne pouvons donc pas utiliser cette technique pour transformer nos scores en probabilités.

Pour cette raison, nous utilisons une méthode simple qui consiste à estimer les probabilités a posteriori sur un ensemble de validation. Cette estimation se fait en construisant un histogramme qui compte le nombre de détections positives et négatives pour des intervalles de score donnés. Nous utilisons alors une interpolation linéaire pour calculer les probabilités pour les images de test à partir des scores obtenus.

5.2.4 Classification par détection

Nous présentons ici la technique utilisée pour obtenir des probabilités de classification d'images à partir de celles de localisation d'objets. Cette probabilité est la probabilité qu'au moins une des fenêtres de détection soit positive. Elle est donnée par la formule suivante :

$$P^{cls} = P_1^{det} + (1 - P_1^{det}) * (P_2^{det} + (1 - P_2^{det}) * (... * (P_{Nb}^{det})...)) \tag{5.8}$$

où P_i^{det} est la probabilité que la $i^{\text{ème}}$ meilleure fenêtre de détection contient un objet et Nb le nombre de fenêtres considérées.

Nous avons testé plusieurs valeurs de Nb et avons observé que les meilleurs résultats sont obtenus en considérant seulement une fenêtre par image ($Nb = 1$). C'est cette valeur que nous utilisons pour les résultats présentés ci-dessous. Nous pensons que la baisse de performance enregistré en augmentant Nb est dû à la non fiabilité de l'estimation de probabilités.

5.3 Expérimentation

Nous présentons dans cette section les résultats de localisation obtenus par l'approche de combinaison décrite dans ce chapitre. Pour ce faire, nous nous basons sur le détecteur à 2 niveaux décrit dans le chapitre 4 et le classifieur INRIA_flat qui donnent des résultats similaires à ceux de l'état de l'art. Nous choisissons de travailler sur des bases de données réalistes où les objets sont vus sous différents angles et possèdent une grande variance intra-classe et où le contexte est particulièrement compliqué et difficile à appréhender. La base d'image utilisée à cet effet est la base

PASCAL 2007 [19]. Dans un deuxième temps, nous comparons nos résultats à ceux obtenus par les participants au challenge PASCAL VOC 2008 [20] [2].

5.3.1 Résultats de classification

Nous présentons dans la table 5.2 les résultats de classification obtenus sur la base PASCAL 2007, par localisation uniquement (1ère colonne), par l'approche INRIA_flat (2ème colonne), par produit des probabilités (3ème colonne) et par notre méthode de combinaison (4ème colonne). En termes de mAP, les résultats obtenus par localisation sont modestes, mais pour quelques classes telles que les bouteilles et les voitures, la classification par localisation donne de très bons résultats et arrive même à battre le classifieur INRIA_flat. Prenons l'exemple des bouteilles, celles-ci appa-

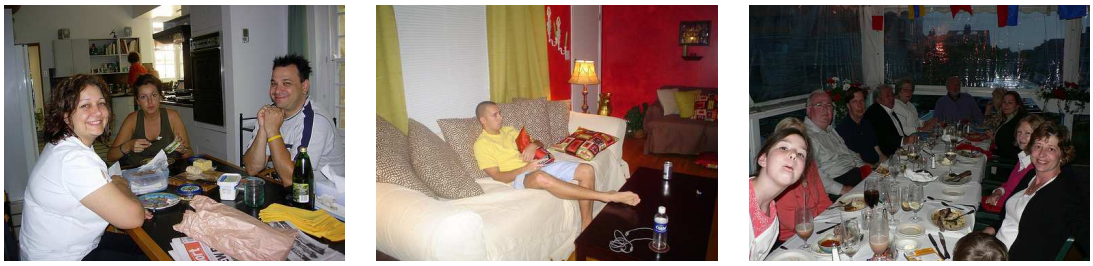


Fig. 5.5: Images de bouteilles : les deux images de gauche contiennent des bouteilles alors que celle de droite n'en contient pas. Ces images reflètent la difficulté de la tâche pour les classifieurs d'images.

raissent la plupart du temps dans des images d'intérieur et occupent généralement une petite portion de l'image (voir figure 5.5). Ceci porte à croire que le classifieur d'images de bouteilles va plus s'attacher au contexte qu'à l'objet lui même. Par contre le détecteur de bouteilles va apprendre l'apparence des bouteilles et saura donc mieux les retrouver.

Avec la combinaison proposée ici (colonne "Notre combinaison"), nous observons un gain significatif de 3.4% en termes de mAP par rapport au classifieur INRIA_flat. En plus, pour quelques classes (ex. bouteille, plante) le gain obtenu s'approche des 10% en termes d'AP.

Les améliorations les plus significatives sont obtenues pour les classes où le classifieur et le détecteur ne se basent pas sur les mêmes informations : 11% pour les bouteilles, 9.5% pour les plantes (le classifieur s'attarde sur le feuillage alors que le détecteur lui se concentre plus sur le pot) ou encore les vaches (6.8%), les moutons (6.1%) et

[2] les annotations de l'ensemble de test n'étant pas divulguées, les évaluations sont effectuées par les organisateurs de la compétition

	Localisation	INRIA_flat	Produit	Notre combinaison
plane	58.2	76.7	75.0	**77.2**
bike	59.5	64.4	69.0	**69.3**
bird	19.3	55.8	51.7	**56.2**
boat	26.9	**69.3**	67.7	66.6
bottle	37.5	34.1	**48.8**	45.5
bus	60.8	62.6	**68.7**	68.1
car	80.2	76.0	83.3	**83.4**
cat	36.7	**58.4**	54.6	53.6
chair	39.3	55.9	57.6	**58.3**
cow	40.7	44.3	**53.7**	51.1
table	28.3	60.3	56.6	**62.2**
dog	35.2	**48.3**	46.4	45.2
horse	67.9	77.4	**78.7**	78.4
moto	63.8	63.5	69.4	**69.7**
person	74.0	85.7	84.8	**86.1**
plant	38.4	42.9	51.8	**52.4**
sheep	40.1	48.3	**54.4**	**54.4**
sofa	35.7	49.0	**55.5**	54.3
train	56.7	**76.4**	73.0	75.8
tv	54.3	54.5	**62.8**	62.1
mAP	47.7	60.1	63.2	**63.5**

Tab. 5.2: Résultats de classification pour la base PASCAL 2007 obtenus par localisation (1ère colonne), par classification (2ème colonne) et par combinaison (deux dernières colonnes).

les sofas (5, 3%) où le classifieur va s'intéresser au contexte et le détecteur à la structure géométrique des objets. En revanche, pour quelques classes comme les chats et les chiens la combinaison détériore les résultats. La colonne "Produit" montre les résultats obtenus avec une méthode plus simple qui consiste en la multiplication de probabilités. Même s'ils sont légèrement inférieurs, ces résultats s'avèrent très proches de ceux obtenus par notre approche de combinaison. Nous avons aussi évalué d'autre méthodes de combinaison de probabilités comme les règles de MAX ou MIN [43] ou encore d'utiliser un classifieur SVM pour générer des prédictions en se basant sur les deux scores. Les résultats obtenus par ces différentes approches sont moins bons que ceux obtenus par la méthode proposée ou par le produit de probabilités.

La figure 5.6 présente des exemples d'images et montre l'effet qu'a eu la combinaison sur la classification de ces images.

Fig. 5.6: Effet de la combinaison sur la classification : (a) exemples de vrais positifs où le score a été renforcé. (b) exemples de faux positifs où le score a été réduit. (c) exemple de faux positifs où le score a été renforcé.

La table 5.3 compare la méthode de classification proposée aux cinq meilleures méthodes du PASCAL VOC challenge 2008 [20]. Notre méthode obtient une mAP de 57.7%, ce qui représente un gain de 2.8% par rapport à la meilleure approche du challenge. De plus, notre approche obtient les meilleures performances sur 13 des 20 catégories. Pour ce qui est du PASCAL VOC challenge 2007, nous obtenons un mAP de 63.5% (voir tableau 5.2). Comparés à ceux obtenus au challenge, ces résultats représentent une amélioration de 4.1% en termes de mAP par rapport à la meilleure méthode et obtiennent les meilleurs résultats sur 15 des 20 catégories.

	LEAR_flat	LEAR_shotgun	Surr.UvA_SRKDA	UvA_Soft5ClrSift	UvA_TreeSFS	Notre méthode
plane	80.1	**81.1**	79.5	79.7	80.8	80.1
bike	51.8	52.9	54.3	52.1	53.2	**57.7**
bird	60.5	**61.6**	61.4	61.5	**61.6**	60.7
boat	66.9	67.8	64.8	65.5	65.6	**69.4**
bottle	29.1	29.4	30.0	29.1	29.4	**43.7**
bus	52.0	52.1	52.1	46.5	49.9	**52.7**
car	57.4	58.7	59.5	58.3	58.5	**70.0**
cat	58.6	59.9	59.4	57.4	59.4	**60.7**
chair	48.7	48.5	48.9	48.2	48.0	**50.9**
cow	31.0	32.0	**33.6**	27.9	30.1	33.4
table	39.2	38.6	37.8	38.3	39.6	**42.9**
dog	47.6	47.9	46.0	46.6	45.0	**50.1**
horse	64.2	65.4	66.1	66.0	**67.3**	66.0
moto	64.6	65.2	64.0	60.6	60.4	**69.3**
person	87.0	87.0	86.8	87.0	87.1	**87.3**
plant	28.6	29.0	29.2	31.8	30.1	**36.1**
sheep	33.3	34.4	**42.3**	42.2	41.5	40.4
sofa	42.6	43.1	44.0	45.3	45.4	**45.5**
train	73.1	74.3	**77.8**	72.3	74.3	73.5
tv	59.8	61.5	61.2	**64.7**	59.8	64.4
mAP	53.8	54.5	54.9	54.0	54.3	**57.7**

Tab. 5.3: Comparaison des résultats de classification à l'état de l'art : nous obtenons la meilleure performance pour 13 des 20 classes. Notons que les meilleures méthodes participant au challenge obtiennent la même performance à 1% près que notre classifieur d'images et que notre méthode les surpasse de près de 3%. Notons que la méthode Lear_flat est équivalente à la méthode INRIA_flat.

5.3.2 Résultats de localisation

Nous présentons dans la table 5.4 les résultats de localisation obtenus par la méthode de localisation seule (1$^{\text{ère}}$ colonne), par produit des probabilités (2$^{\text{ème}}$ colonne), par la méthode de localisation combinée avec l'approche contextuelle de [23] (3$^{\text{ème}}$ colonne) et par notre méthode de combinaison avec le classifieur INRIA_flat (4$^{\text{ème}}$ colonne).

Nous remarquons d'abord que l'approche de combinaison proposée permet d'obtenir les meilleurs résultats avec un gain de 2.6% en termes de mAP par rapport au détecteur de base. Plus intéressant encore, les améliorations les plus importantes sont obtenues pour les classes d'animaux comme les vaches (7.6%) ou les moutons (6.1%) mais aussi pour les classes d'intérieur comme les bouteilles (4.5%) et les chaises (3.4%) ce qui confirme notre hypothèse que la combinaison avec le classifieur

	Notre détecteur	Contexte de [23]	Produit	Notre combinaison
plane	33.8	**35.4**	33.9	35.1
bike	43.0	44.7	40.7	**45.6**
bird	09.7	10.6	10.8	**10.9**
boat	09.6	05.5	11.9	**12.0**
bottle	18.7	20.3	22.7	**23.2**
bus	41.9	**42.1**	36.8	**42.1**
car	50.4	50.7	47.1	**50.9**
cat	15.0	18.7	18.7	**19.0**
chair	14.6	16.2	17.4	**18.0**
cow	23.9	26.2	**31.9**	31.5
table	15.1	14.2	17.0	**17.2**
dog	15.4	16.3	16.5	**17.6**
horse	48.2	49.0	48.0	**49.6**
moto	41.7	42.9	38.1	**43.1**
person	20.2	20.5	20.9	**21.0**
plant	16.1	17.0	18.4	**18.9**
sheep	21.2	23.4	26.6	**27.3**
sofa	20.3	21.0	23.6	**24.7**
train	29.1	**30.7**	28.6	29.9
tv	38.2	39.3	37.2	**39.7**
mAP	26.3	27.2	27.3	**28.9**

Tab. 5.4: Résultats de localisation sur la base PASCAL 2007 obtenus par notre détecteur à deux niveaux seul (première colonne), combiné avec des informations du contexte (3 dernières colonnes).

s'apparente à l'utilisation de l'information du contexte, très discriminante pour ce type de classes.

Remarquons que pour la classe "personne", la combinaison améliore très peu les performances. En effet, les personnes peuvent être présentes sur un contexte quelconque, et de ce fait, l'information contextuelle importe peu.

Nous comparons aussi notre méthode d'exploitation du contexte à celle de [23]. Cette dernière, apprend un classifieur SVM avec comme primitives, le score initial de la fenêtre de détection, sa position et le score de la meilleure détection dans l'image pour toutes les autres catégories. Le score final est alors la sortie du classifieur SVM appris. Nous appliquons cette approche sur nos détections et reportons le résultat dans la deuxième colonne de la table 5.4. L'amélioration obtenue est moindre que celle obtenue par notre méthode de combinaison. Ces résultats montrent que l'information apportée par le classifieur d'images est plus utile que celle que l'infor-

mation contextuelle émanant des détections des autres catégories mais suggère que l'incorporation de ces deux informations peut améliorer les résultats.

La figure 5.7 présente des résultats visuels de cas où la combinaison renforce des bonnes détections, contredit des faux positifs et se trompe en renforçant des faux positifs et ce pour trois classes d'objets différentes.

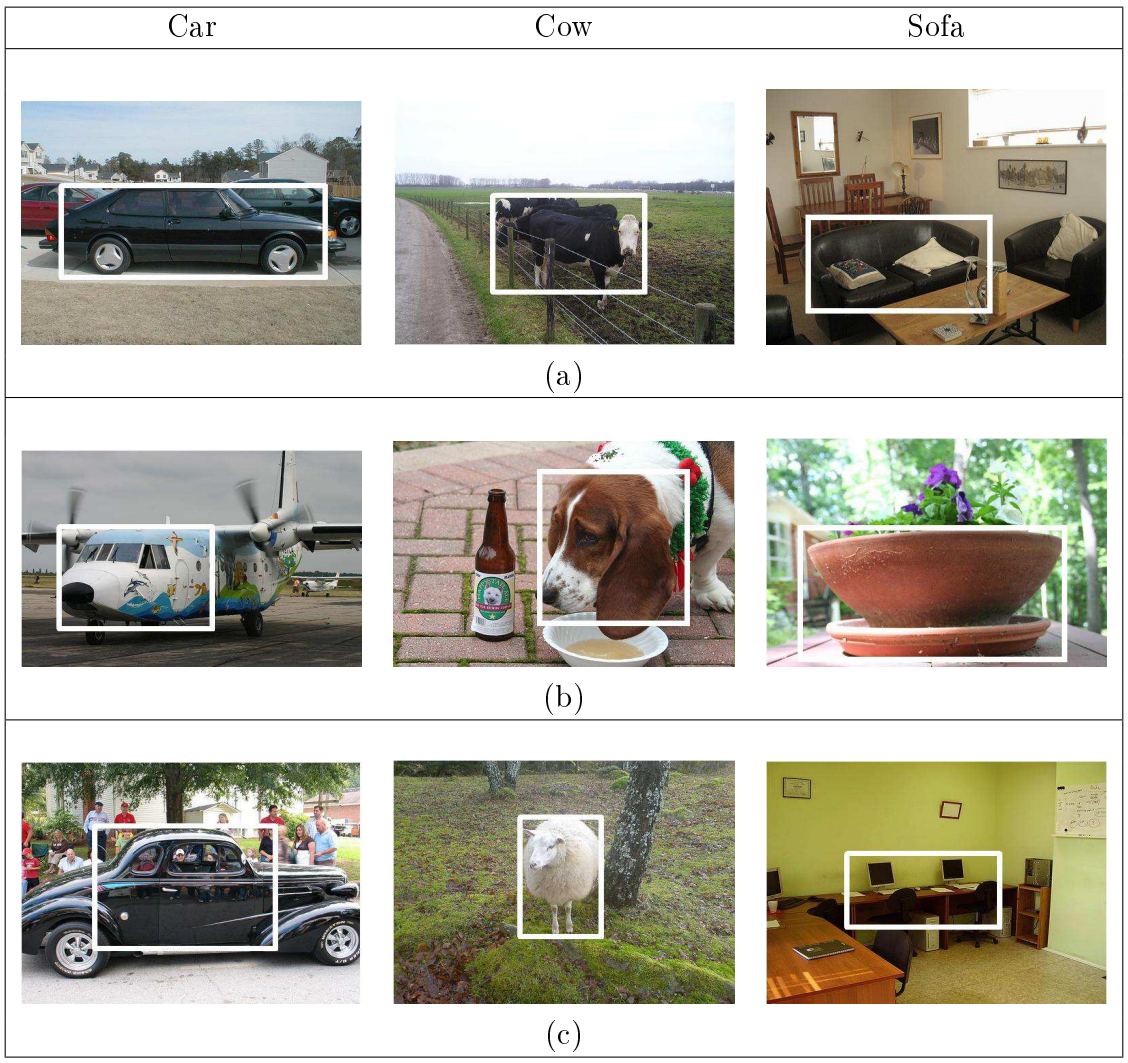


Fig. 5.7: Effet de la combinaison sur la localisation : (a) exemples de bonnes détections où le score a été renforcé. (b) exemples de faux positifs où le score a été réduit. (c) exemple de faux positifs où le score a été renforcé.

À quelques détails près, la méthode présentée ici est celle qui a participé au PASCAL VOC challenge 2008 [20] pour la tâche de localisation. Cette méthode a remporté la compétition en obtenant les meilleurs résultats sur 11 des 20 catégories et un mAP comparable à celui obtenu par la méthode "UoCTTIUCI" [23] (co-vainqueur

du challenge). La table 5.5 montre les résultats obtenus par les meilleures méthodes du challenge.

	CASIA_Det	Jena	MPI_struct	Oxford	UoCTTIUCI	XRCE_Det	Notre méthode
plane	25.2	4.8	25.9	33.3	32.6	26.4	**36.6**
bike	14.6	1.4	8.0	24.6	**42.0**	10.5	33.8
bird	9.8	0.3	10.1	-	**11.3**	1.4	10.7
boat	10.5	0.2	5.6	-	11.0	4.5	**11.4**
bottle	6.3	0.1	0.1	-	**28.2**	0.0	23.3
bus	23.2	1.0	11.3	-	23.2	10.8	**23.7**
car	17.6	1.3	10.6	29.1	32.0	4.0	**36.6**
cat	9.0	-	**21.3**	-	17.9	7.6	15.8
chair	9.6	0.1	0.3	-	**14.6**	2.0	12.9
cow	10.0	4.7	4.5	12.5	11.1	1.8	**17.9**
table	13.0	0.4	10.1	-	6.6	4.5	**15.1**
dog	5.5	1.9	**14.9**	-	10.2	10.5	9.7
horse	14.0	0.3	16.6	32.5	32.7	11.8	**36.5**
moto	24.1	3.1	20.0	34.9	38.6	13.6	**39.4**
person	11.2	2.0	2.5	-	**42.0**	9.0	19.6
plant	3.0	0.3	0.2	-	**12.6**	1.5	11.6
sheep	2.8	0.4	9.3	-	16.1	6.1	**19.4**
sofa	3.0	2.2	12.3	-	13.6	1.8	**16.3**
train	28.2	6.4	23.6	-	24.4	7.3	**29.5**
tv	14.6	13.7	1.5	-	**37.1**	6.8	35.6
mAP	12.7	-	10.4	-	**22.8**	07.1	22.7

Tab. 5.5: Comparaison des résultats de localisation à l'état de l'art : notre méthode obtient la meilleure performance pour 11 des 20 classes.

Pour ce qui est de la base PASCAL 2007, nous obtenons un mAP de 28.9%(voir le tableau 5.4). Comparé aux résultats du challenge, nous obtenons la meilleure performance sur 18 des 20 catégories et un gain de 11.8% en termes de mAP par rapport à la meilleure méthode. Ceci confirme l'amélioration et l'intérêt qu'ont suscité les méthodes de localisation durant ces dernières années.

5.4 Conclusion

Dans ce chapitre nous avons proposé une méthode combinant classification d'images et localisation d'objets en vue de l'amélioration des performances sur chacune des deux tâches. Comme nous l'avons montré, cette méthode permet effectivement une amélioration significative des résultats aussi bien pour la tâche de classification que pour la tâche de localisation.

Ces résultats montrent d'abord l'importance et le potentiel de l'information contextuelle dans l'amélioration des algorithmes de localisation d'objets, et ce même pour des bases d'image difficile.

Ils montrent aussi qu'il est possible d'aborder la tâche de classification d'images à partir de celle de détection d'objets, que pour quelques classes ceci donne de meilleurs résultats que les meilleures classifieurs d'images existants et que la combinaison permet une amélioration significative de la qualité des résultats.

Nous pensons que ce type de combinaison de méthodes, qui pourrait être étendue à d'autres tâches comme celle de la segmentation, sera incontournable dans la mise en place de classifieurs robustes. Une autre possibilité d'extension de ces travaux est celle de la combinaison des classifieurs et des détecteurs des différentes classes d'objets. Les travaux récents de [16] exploitent cette piste et reporte des résultats très encourageants.

Notons que cette approche, publiée à la conférence ICCV 2009 [33], a été reprise par un des gagnants de la tâche de classification [42] lors du challenge PASCAL VOC 2009.

6

Approche hiérarchique pour la détection et l'identification de véhicules

Sommaire

Les travaux proposés dans les chapitres précédents sont tous directement liés à la notion de classe d'objets. Or cette séparation en classes repose sur une définition purement sémantique des objets, séparation qui n'est pas forcément corrélée à la similarité visuelle : deux objets de classes différentes peuvent se ressembler alors que deux objets de la même classe peuvent être visuellement différents.

C'est ce qui nous a poussé dans les chapitres précédents à utiliser plusieurs classifieurs distincts pour chaque classe et pour chaque apparence, afin qu'ils voient leur travail facilité en ne devant reconnaître que des objets d'apparence relativement similaire (par exemple Voiture vue de profil).

Cependant, lorsque des similarités importantes existent entre des objets de différentes classes il peut être opportun, pour mieux différencier les objets du fond, d'entrainer des classifieurs sur des exemples provenant de ces classes visuellement proches, la séparation entre les classes ne se faisant que dans un second temps. Par exemple, plutôt que d'entrainer un détecteur de 'vélos vus de profils' et un détecteur de 'motos vues de profils' il peut être préférable d'entrainer un détecteur de 'vélos

et motos vus de profils', qui bénéficiera de deux fois plus d'exemple d'entrainement et sera donc probablement plus performant.

C'est ce que nous proposons ici à travers une méthode à deux niveaux. Le premier correspond à la détection des objets et le second à l'identification des objets détectés. Les termes *détection* et *identification* sont définis dans ce chapitre de la manière suivante : la détection correspond à la localisation des objets présents dans une image sans reconnaitre la classe à la quelle ils appartiennent. L'identification correspond à l'attribution d'une classe à un objet préalablement détecté.

Nous proposons d'étudier dans ce chapitre dans quelle mesure cette stratégie est opportune, et de quelle manière les détecteurs d'objets peuvent être rendus plus performants en regroupant éventuellement des exemples d'apprentissage de classes visuellement proches.

Ce chapitre est organisé comme suit. La première section présente la méthode proposée ici que nous appelons *classification hiérarchique*. Puis dans la deuxième section, nous présentons les résultats expérimentaux obtenus avec cette méthode.

6.1 Présentation de la méthode

Nous présentons ici notre méthode de classification hiérarchique. Cette méthode décompose la tâche de localisation de classes d'objets en deux sous tâches : la localisation des objets puis leur identification (voir figure 6.1 pour une illustration). Cette décomposition permet de se concentrer dans un premier temps sur la séparation entre les objets et le fond. Le classifieur appris à cet effet, que nous appelons *détecteur générique*, se focalise sur les différences entre objets et fonds. Les similarités entre objets qui constituent un handicap pour les approches classiques se trouvent être bénéfiques à ce niveau et permettent une meilleure reconnaissance. Nous supposons qu'à la sortie de cette étape toutes les régions sélectionnées ne contiennent que des objets. Ceci permet aux classifieurs effectuant l'identification de se concentrer uniquement sur les différences entre classes.

Notons aussi que cette approche permet d'accélérer les calculs puisque le nombre de passages de fenêtres glissantes sur l'image est réduit à chaque fois que des détecteurs spécifiques sont fusionnés.

Nous présentons dans ce qui suit plus en détails ces deux niveaux.

6.1.1 Première étape : détection

La construction du détecteur générique suit l'approche proposée dans le chapitre 4. L'unique différence est que les exemples d'apprentissage utilisés ici correspondent à la fusion d'exemples d'apprentissage provenant de classes visuellement proches.

Le détecteur obtenu est donc un détecteur à deux étages de classifieurs, le premier linéaire, le second non-linéaire. Le descripteur utilisé est le descripteur HOG.

La question de savoir quelles classes doivent être regroupées pour obtenir des performances optimales se pose donc. Nous en sommes restés au stade de la validation du concept et n'avons pas proposé de méthodes automatique de regroupement. Nous avons manuellement formé des groupes, qui peuvent aller de la fusion de toutes les classes à des regroupements plus limités par exemple de deux classes différentes. Notons que dans tous les cas, les différentes vues (profil, avant, etc.) sont traités séparément. En effet, nous pensons que l'information de point de vue joue un rôle important et qu'une bonne modélisation de la classe (même générique) nécessite l'apprentissage de classifieurs distincts par point de vue. Leur apparence est trop différente pour être apprise par un unique classifieur. Cela est également vrai pour l'étape d'identification, où les classifieurs appris ne raisonnent que par vue. Nous évaluons dans le paragraphe 6.2.1 l'influence de cette procédure sur les performances d'identification.

6.1.2 Deuxième étape : identification

Une fois les objets détectés, il faut encore déterminer leur catégorie. Pour ce faire, nous avons implémenté un classifieur multi-classes. Deux stratégies de classification multi-classes ont été considérés : la stratégie "1 contre 1" (*"1 vs 1"*) et "1 contre tous" (*"1 vs all"*).

La stratégie "1 contre 1", consiste à apprendre un classifieur par paire de catégories. Ce classifieur sera spécialisé dans la séparation des deux classes de cette paire. Pour appliquer ce classifieur multi-classes, nous appliquons tous les classifieurs de paires. Chacun de ces classifieurs vote pour une classe et la fenêtre en question est affectée à la classe obtenant le plus grand nombre de votes.

La stratégie "1 contre tous" quant à elle, consiste à apprendre un classifieur par catégorie pour séparer cette catégorie de toutes les autres. En procédant ainsi, ce classifieur concentre toute son énergie sur les spécificités de cette catégorie et les différences par rapport aux autres catégories. Lorsque cette stratégie est appliquée, tous ces classifieurs sont appliqués et la fenêtre en question est affectée à la catégorie obtenant la plus haute probabilité.

Le choix de la stratégie ("1 contre 1" ou "1 contre tous") est effectué par validation croisée. En d'autres termes, la stratégie qui obtient les meilleures performances sur un ensemble de validation est sélectionnée.

Le classifieur élémentaire dans les deux cas est un SVM non-linéaire avec noyau RBF χ^2. Le paramètre γ de ce noyau est obtenu comme précédemment selon l'heuristique de [88].

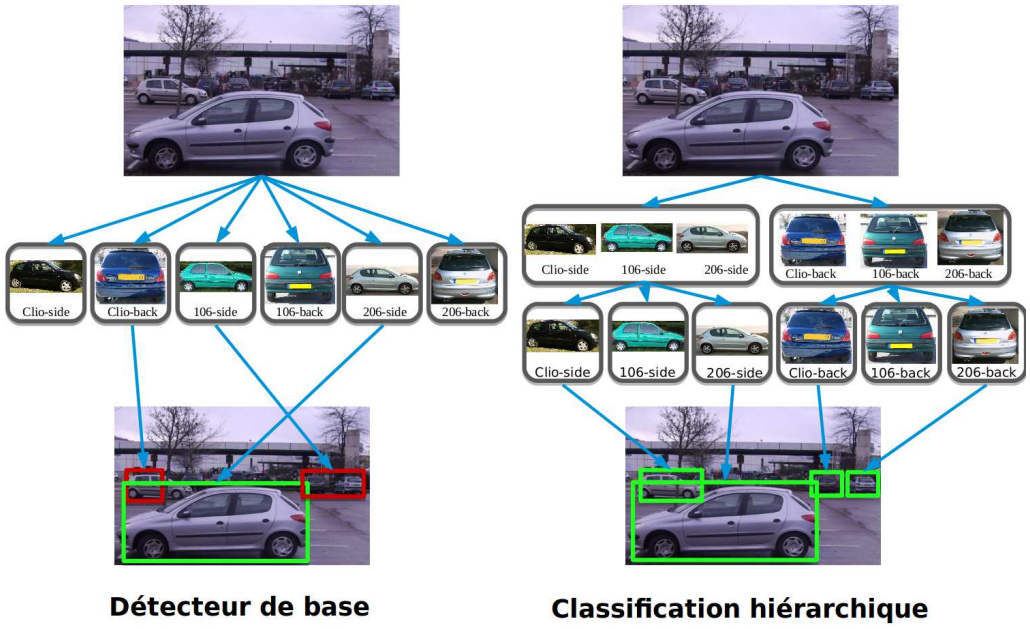


Fig. 6.1: Principe de la détection hiérarchique : Le détecteur de base utilise un détecteur par classe et par vue. Notre approche hiérarchique détecte d'abord la présence des objets pour les identifier ensuite.

Correction des positions des fenêtres Notre méthode de détection utilise un seul ratio d'aspect par vue. Or, les objets des différentes classes peuvent avoir des ratios d'aspects très différents. Nous utilisons donc pour la localisation le ratio d'aspect moyen de tous les objets (un par point de vue) et changeons les annotations d'apprentissage en agrandissant les fenêtres pour qu'elles aient toutes le même ratio d'aspect. Cette procédure peut donner des détections moins précises (des fenêtres de détection nettement plus grandes que l'objet) et plusieurs bonnes détections vont ainsi être considérées comme des faux positifs par le critère d'appariement de PASCAL [20]. Nous surmontons ce problème en appliquant, après identification, l'inverse de la transformation moyenne (appliquée pour obtenir les annotations d'apprentissage).

6.2 Résultats expérimentaux

Nous présentons dans cette section les résultats expérimentaux obtenus par l'approche de classification hiérarchique proposée ici. Nous comparons ces résultats à ceux obtenus par l'approche décrite dans le chapitre 4 que nous appelons *baseline*. Dans le premier paragraphe, nous commençons par évaluer la performance de détec-

tion (capacité à trouver l'objet sans l'identifier). Puis, nous présentons les résultats après identification. Les bases de données sur lesquelles nous travaillons ici sont la base MBDA, Base de voitures et la base des Voitures de PASCAL (voir section 2.2 pour plus de détails sur les bases de test).

6.2.1 Détection d'objets

Nous présentons ici les résultats de détection obtenus par la méthode de classification hiérarchique que nous comparons à la baseline. Pour obtenir des résultats de détection à partir de la baseline, nous fusionnons les sorties des détecteurs de toutes les catégories. Ainsi, les confusions entre classes sont ignorées et seule la capacité de détection (non d'identification) est évaluée. Concernant le classifieur hiérarchique, nous comparons deux approches. La première apprend un classifieur par vue (appelé *Détecteur générique par vue*) et une deuxième qui apprend un seul classifieur toutes vues confondues (appelé *Détecteur générique global*).

Méthode	Base MBDA	Base de voitures	Voitures de PASCAL
Détecteur de base	67.4	85.4	45.1
Détecteur générique global	67.7	85.3	48.8
Détecteur générique par vue	**72.8**	**89.2**	**50.4**

Tab. 6.1: Résultats de détection (AP) pour les différents détecteurs et les différentes bases considérées.

Les mesures de performance sont présentées dans la table 6.1. Nous observons que pour les trois bases de données, le détecteur générique par vue donne les meilleurs résultats et que le gain obtenu par rapport à la baseline est considérable (5% pour la base de MBDA, 4% pour la base de voitures et 5% pour les voitures de PASCAL). Ce gain est occasionné par deux facteurs. Le premier est que la tâche devient plus facile lorsqu'on se concentre uniquement sur la séparation des objets du fond, économisant ainsi l'effort de construction de frontières entre les différentes classes. Le second est que le nombre d'exemples d'apprentissage est plus important pour les détecteurs génériques. Il est évidemment souhaitable que le détecteur générique obtienne des performances meilleures que la baseline, sur la tâche de détection seule, pour pouvoir espérer une amélioration des résultats après identification des objets. En ce qui concerne la différence entre le détecteur générique global et le détecteur générique par vue, nous pensons que le gain obtenu par ce dernier est dû à la différence d'apparence entre les différentes vues. Ceci pousse le détecteur à généraliser à toutes les vues ce qui rend la frontière avec le fond plus difficile à trouver. Ceci confirme notre intuition selon laquelle que le groupement des classes et le partitionnement par vue constitue un compromis raisonnable.

6.2.2 Identification d'objets

Nous évaluons ici l'algorithme complet — à savoir la détection suivie de l'identification — et nous le comparons à la baseline. La procédure d'évaluation est celle des challenges PASCAL VOC [20].

La table 6.2 présente les résultats obtenus sur la base MBDA, la table 6.3 présente ceux de la base de voitures et la table 6.4 présente les résultats sur la base des voitures de PASCAL. Comparé à la baseline, l'approche de classification hiérarchique donne des résultats nettement meilleurs. Ceci est plus perceptible pour la base MBDA où le gain en termes de mAP est de 7% et dépasse même les 20% pour quelques catégories telle que le "Camion 3". Le gain est plus modéré pour la base de voitures (3% en termes de mAP) mais il atteint tout de même 7% pour quelques catégories telles que les "Punto". Nous montrons aussi pour ces deux approches les résultats qu'on aurait obtenus si on disposait d'un détecteur parfait ou d'un classifieur multi-classes parfait. Nous remarquons que pour la base MBDA, une amélioration de l'une ou de l'autre composante impliquerait une nette amélioration de la qualité des résultats. Par contre, pour la base de voitures, nous remarquons que la détection ne constitue pas un problème : utiliser un détecteur parfait donnerait des résultats proches que ceux obtenus. C'est l'étape d'identification qui s'avère problématique. La difficulté de cette étape peut s'expliquer par (a) le nombre réduit des exemples d'apprentissage par classe et (b) par le fait que l'identification sur cette base est plus difficile que sur la base MBDA. En effet, même si une classe n'est constituée que par des voitures du même modèle, des variances inter-modèles peuvent subsister (break, décapotable, 3 ou 5 portes, ...) ce qui n'est pas le cas de véhicules de la base MBDA. Nous avons aussi appliqué notre approche sur la base des voitures de PASCAL. Les résultats obtenus sont sensiblement les mêmes que ceux obtenus par la baseline. Les résultats sont reportés dans la table 6.4.

Notons que l'étape de correction des fenêtres joue un rôle important dans la qualité des résultats de la base MBDA où cette étape offre un gain de 3.6% en mAP. Pour les autres bases d'images, aucune d'amélioration notable n'a été observé (les ratios d'aspect des différentes catégories sont proches).

Matrices de confusion Un point fort de notre approche est qu'elle permet de réduire la confusion entre objets. En effet, pendant la deuxième étape, nous nous concentrons sur définition des frontières entre les classes et les classifieurs obtenus se trompent donc moins que ceux de la baseline. D'autre part, avec cette approche, une fenêtre ne peut être affectée qu'à une et une seule classe (alors que la même fenêtre peut être retrouvée plusieurs fois pour des classes différentes par la baseline). Ceci est particulièrement visible pour les 3 classes de camions de la base MBDA (voir figure 2.8 pour une illustration) qui sont extrêmement similaires.

Catégorie	Baseline	Classif. hiérarchique	Identif. parfaite	Détection parfaite
Véhicule léger 1	35.3	**42.2**	72.7	84.6
Véhicule léger 2	39.0	**55.7**	81.8	87.9
Véhicule léger 3	**41.4**	40.8	72.7	82.6
Véhicule léger 4	53.7	**54.5**	72.7	90.1
Véhicule léger 5	26.0	**44.9**	63.6	89.4
Camion 1	28.0	**55.2**	81.8	71.5
Camion 2	16.4	**32.2**	72.7	75.7
Camion 3	16.4	**35.5**	72.7	61.8
Poids lourd 1	62.1	**72.9**	90.9	89.9
Poids lourd 2	49.6	**71.4**	81.8	86.8
Poids lourd 3	51.0	**65.3**	81.8	87.3
Poids lourd 4	75.2	**75.5**	90.9	89.5
Poids lourd 5	**82.4**	79.1	81.8	90.6
Poids lourd 6	**69.5**	67.7	81.8	90.1
Poids lourd 7	**75.4**	73.7	81.8	90.6
Poids lourd 8	67.7	**73.0**	90.9	90.4
Poids lourd 9	**81.4**	72.6	90.9	90.4
Poids lourd 10	**80.4**	78.7	90.9	89.7
Poids lourd 11	82.2	**85.3**	90.9	90.8
Poids lourd 12	70.8	**76.2**	90.9	90.1
mAP	55.2	**62.6**	81.8	86.0

Tab. 6.2: Résultats obtenus par classification hiérarchique pour la base MBDA (AP).

Catégorie	Baseline	Classif. hiérarchique	Identif. parfaite	Détection parfaite
106	47.6	**53.8**	90.9	0.385
206	51.1	**55.3**	100	0.547
407	75.0	**77.8**	90.9	0.701
Ax	50.9	**53.4**	90.9	0.666
Clio	**45.5**	45.1	81.8	0.471
Logan	**30.9**	29.7	90.9	0.432
Punto	21.2	**27.2**	81.8	0.285
Qashqai	67.4	**71.6**	90.9	0.704
mAP	48.7	**51.7**	89.7	0.524

Tab. 6.3: Résultats obtenus par classification hiérarchique pour la base de voitures (AP).

Nous présentons dans la table 6.5 la matrice de confusion à un taux de rappel de 10% obtenu avec la baseline et celle proposée dans ce chapitre pour les 3 classes de camions. Avec la classification hiérarchique, la confusion entre les camions est nettement réduite. Ceci est particulièrement évident pour la classe “camion 3” pour qui la précision à 10% de rappel est passée de 43% à 81% principalement à cause de la réduction de la confusion avec les autres classes de camions. Finalement, nous mon-

Catégorie	Baseline	Classif. hiérarchique
Sport	0.423	0.424
4x4	0.145	0.144
Autre	0.213	0.241

Tab. 6.4: Résultats obtenus par classification hiérarchique pour les voitures de PASCAL (AP).

trons dans la figure 6.2 des exemples de détections qui, contrairement à la baseline, on été correctement identifiés par le classifieur hiérarchique.

Punto(grande) / Clio(petite)

106

Punto

Fig. 6.2: Exemples d'objet correctement détectés par la classification hiérarchique sur la base de voitures.

		Classe détectée				
Détecteur de		Camion 1	Camion 2	Camion 3	Autre	Fond
	Camion 1	85.3	3.1	6.7	4.5	0.4
	Camion 2	15.0	41.7	20.9	19.4	3.0
	Camion 3	22.9	14.5	43.4	19.2	0.9

(a)

		Classe détectée				
Détecteur de		Camion 1	Camion 2	Camion 3	Autre	Fond
	Camion 1	97.5	1.0	1.0	0.5	0.0
	Camion 2	4.0	51.8	34.2	8.4	1.6
	Camion 3	8.0	0.8	81.0	9.4	0.8

(b)

Tab. 6.5: Matrices de confusion à un taux de rappel de 10% (a) avec la baseline et (b) avec la classification hiérarchique pour les 3 classes de camions de la base MBDA.

6.3 Conclusion et discussion

Nous avons introduit dans ce chapitre une nouvelle approche hiérarchique pour la détection et l'identification des objets, basée sur la méthode de localisation présentée dans les chapitres précédents. Nous avons montré que cette approche permet d'améliorer significativement la qualité des détecteurs pour les classes souffrantes d'un taux de confusion élevé, c'est-à-dire les classes visuellement proches.

Lors de nos travaux nous avons également expérimenté cette méthode sur d'autres classes de la base d'images PASCAL. Nous avons par exemple regroupé quelques classes d'animaux et essayé de construire un détecteur générique pour ces classes. Les résultats obtenus étaient très en deçà de ceux obtenus par la baseline. Le même phénomène a été observé pour les groupes "voitures-bus" et "vélo-moto". Ceci montre que la définition des classes peut influer grandement sur la qualité des résultats et que même si les travaux récents se sont concentrés sur l'amélioration des méthodes de représentation et de modélisation pour améliorer la qualité des détections, il existe d'autres pistes à explorer. En effet, une question parait très légitime au vu de ces résultats : dans le cas ou la fusion des catégories détériore les résultats, le partitionnement de ces catégories en sous-catégories n'améliorerait-il pas les performances ? La définition actuelle des classes est optimale ? Sinon, comment l'améliorer ?

Notre intuition après l'analyse des résultats expérimentaux est qu'il y a un compromis à trouver pour avoir un partitionnement optimal. Nous pensons que ce partitionnement optimal dépend de deux paramètres principaux qui sont la similarité inter-classes et la richesse du fond (c'est-à-dire quelle point il contient des distracteurs). Si le fond est relativement simple et que deux classes sont très similaires, leur fusion semble être la meilleure solution car elle permet au détecteur d'exploiter les indices en commun. Par contre, si une classe possède une forte variance intra-classe et que le fond est difficile, il vaut mieux partitionner car, en généralisant aux deux classes, le détecteur va probablement se tromper sur des fenêtres du fond et générer plus de faux positifs (voir la figure 6.3 pour une illustration). La suite de ces travaux pourrait donc être de proposer une méthode de partitionnement et de regroupement automatique des classes d'objets pour construire un arbre de classifieurs.

Une autre direction serait de s'inspirer des travaux de [1] pour améliorer l'étape d'identification en proposant une représentation d'images plus adaptée, offrant ainsi plus d'indices au classifieur multi-classes.

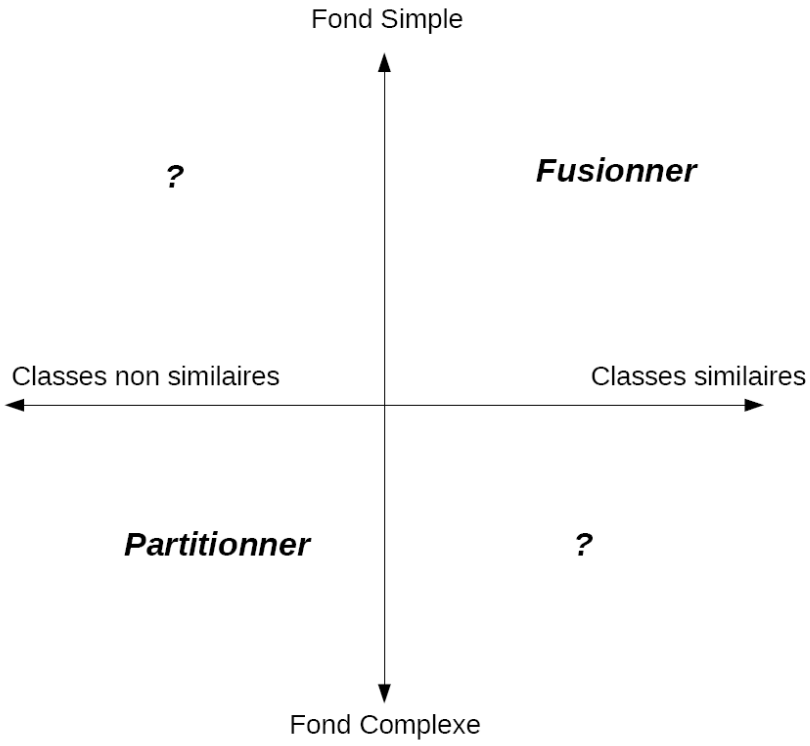


Fig. 6.3: Méthodologie de groupement et partitionnement

7

Conclusion et perspectives

Dans ce travail, nous nous sommes intéressés au problème de la reconnaissance d'objets dans les images à travers les tâches de localisation d'objets et de catégorisation d'images. Nous avons introduit une méthode de localisation basée sur une modélisation rigide des objets que nous avons combiné avec des approches de catégorisation d'images. Tout au long de notre travail nous avons présenté et évalué plusieurs contributions que nous résumons dans la section 7.1. Puis, dans la section 7.2, nous présentons des directions possibles pour les travaux de recherche à venir.

7.1 Contributions

- Notre première contribution a été de proposer une méthode efficace pour la localisation d'objets dans les images qui a constitué le point de départ de nos travaux. Cette méthode est basée sur un classifieur linéaire et des descripteurs HOG et Bag-of-Words, appliqués sur une fenêtre glissante. L'étude paramétrique de cette méthode a permis d'évaluer l'influence des différents composants et de souligner l'importance de certains paramètres notamment ceux de représentation d'image comme le type de découpage spatial, le chevauchement entre les carreaux et la normalisation des descripteurs, ou encore ceux d'apprentissage et notamment le choix des exemples d'apprentissage négatifs.

- Notre deuxième contribution s'est intéressée à l'amélioration de la méthode d'apprentissage automatique utilisée. Pour ce faire, nous avons proposé l'utilisation d'une cascade de deux classifieurs. Le premier étage de la cascade est le classifieur linéaire décrit préalablement, qui malgré sa rapidité et sa capacité à rejeter la plupart des faux positifs, ne permet pas de s'adapter à la complexité des objets. Le deuxième étage de la cascade est un classifieur non linéaire, très coûteux (de l'ordre de 2000 fois plus que le linéaire), mais qui donne d'excellents résultats de classification. Cette cascade permet d'allier rapidité et précision, améliorant

considérablement la qualité des détections. Nous avons ensuite présenté de multiples résultats expérimentaux permettant une meilleure compréhension des forces et des faiblesses de cet algorithme. Les résultats obtenues par ce détecteur sur une base d'images fournie par MBDA France ont été publiés dans la conférence MCM-ITP [34] en Juin 2009.

- Nous nous sommes intéressés ensuite à la combinaison de la localisation d'objets avec la catégorisation d'images. Cette combinaison permet, d'une part, de prendre en compte l'information contextuelle dans la localisation des objets, et d'autre part d'exploiter la structure géométrique des objets dans la catégorisation d'images. Nous avons constaté dans nos expérimentations une amélioration significative des résultats aussi bien pour la tâche de classification que pour la tâche de localisation. Ces travaux ont été publiés dans la conférence ICCV 2009 [33] et ont été présentés en session orale. Les résultats préliminaires de localisation obtenus par cette méthode ont été soumis au PASCAL VOC challenge 2008 et ont obtenu la meilleure performance sur 11 classes parmi 20 et ont été présentés lors du PASCAL Visual Object Classes Challenge Workshop [35].

- Un des problèmes sur lesquels nous nous sommes penchés est celui de la localisation de classes d'objets très similaires. Nous avons proposé une méthode en deux étapes. La première étape est celle de la détection où l'algorithme se charge de trouver la position de tous les objets sans en déterminer leurs classes. La deuxième étape, celle de l'identification, se charge alors de déterminer la classe de chacun des objets. En procédant ainsi, nous sommes capables de détecter la présence de la plupart des objets même si ces derniers sont mal identifiés par la suite. Cette méthode a été expérimentée sur une base d'images de véhicules fournie par MBDA France ainsi que sur une autre base d'images que nous avons crée. Les résultats obtenus montrent que la méthode proposée permet d'obtenir une nette amélioration des performances pour les bases d'images où existe une grande similarité entre les classes.

- Notre dernière contribution a été l'évaluation détaillée des méthodes proposées dans ce manuscrit sur la base d'images fournie par MBDA France. Les résultats de l'évaluation sont présentés dans l'Annexe A (classé "confidentiel" et donc absent de la version publique du manuscrit).

7.2 Perspectives

Les résultats obtenus pendant ces dernières années ouvrent un certain nombre de perspectives pour des travaux futurs.

- Motivés par le contexte applicatif, nous avons choisi d'utiliser tout au long de notre travail des modèles rigides pour représenter les objets. En effet, notre algorithme devait être capable de détecter des objets de petite taille. Une autre possibilité aurait été d'utiliser un modèle par parties. Ce type de modèles, a connu un regain d'intérêt suite aux améliorations introduites par de récents travaux et notamment ceux de [24]. Une direction à explorer serait d'adapter l'approche présentée ici à une modélisation par parties des objets. Il serait effectivement intéressant de voir, entre autre, l'influence de la cascade linéaire/non linéaire sur la qualité des détections.

- Dans ce travail, nous avons proposé l'utilisation d'une cascade à deux étages pour la classification des fenêtres. Une extension possible serait l'augmentation du nombre de niveaux dans la cascade pour accélérer les calculs. Ceci permetterait de consacrer plus de puissance de calculs aux fenêtres susceptibles de contenir les objets cibles. Nous pouvons envisager que le premier niveau de la cascade soit un classifieur très rapide compatible avec la technique des images intégrales de scores (calcul du score effectué en quatre opérations élémentaires) comme suggéré dans les travaux de [45, 80]. Les primitives utilisées à ce niveau peuvent être des primitives simples et faciles à calculer, comme par exemple les ondelettes de Haar [64]. Nous pouvons aussi penser à des étages de classification intermédiaires constituant un compromis acceptable entre pouvoir discriminatif et puissance de calcul comme par exemple les noyaux additifs [57].

- La technique d'exploration des images utilisée dans nos travaux est celle des fenêtres glissantes. Une des limitations de cette technique est le nombre important de fenêtres à explorer. Une piste à explorer serait donc de s'inspirer de récents travaux pour s'affranchir des fenêtres glissantes. Nous pouvons penser à utiliser le modèle implicite de forme pour générer les fenêtres susceptibles de contenir les objets comme suggéré dans [80]. De la même façon, nous pouvons envisager l'apprentissage de ces modèles implicites de forme non seulement sur les primitives se trouvant sur les objets, mais aussi sur celles appartenant au fond, exploitant ainsi l'information contextuelle. Une autre possibilité serait de construire des cartes de saillance sur lesquelles s'appuyer afin de limiter le nombre de fenêtres à explorer.

- Une des limitations de notre méthode est qu'il n'existe aucune coordination entre les détecteurs des différentes classes et même entre des vues différentes d'une même classe. Il semble donc opportun d'essayer d'améliorer le modèle proposé pour qu'il intègre les informations de similarité et de cooccurence entre les objets. Ceci peut se faire au niveau de l'apprentissage des modèles en intégrant les informations émanant d'autres classes ou des autres vues d'une même classe. On pourrait, par exemple, imaginer des modèles 3D des objets [54] permettant de résoudre les problèmes rencontrés avec les objets vus sous des angles peu communs.

Une autre possibilité serait de modéliser la cooccurence des objets des différentes classes. Ce modèle pourra alors être utilisé plus tard pour repondérer les fenêtres de localisation obtenus en fonction de leur concordance en s'inspirant des travaux de [16]. Ce type d'approches permettrait d'éliminer un bon nombre de faux positifs.

- La méthode de classification hiérarchique proposée dans le chapitre 6 découpe le problème de localisation d'objets en deux tâches, une première tâche de détection puis celle d'identification. Des résultats préliminaires ont montré que d'autres découpages donnent des résultats encourageants. En effet nous pouvons envisager une hiérarchie à plusieurs niveaux permettant une identification de plus en plus précise de l'objet. Il serait donc intéressant de pouvoir construire un algorithme qui permettrait de trouver automatiquement un découpage des classes plus proche de l'optimal. Un tel algorithme pourrait générer une nouvelle définition hiérarchique des classes basée sur la similarité visuelle et non sur la définition sémantique des objets.
  D'autre part, la méthode de représentation d'images utilisée est la même pour les deux étapes de détection et d'identification. Or les informations utilisées pour différencier un objet du fond ne sont pas les mêmes que celles utilisées pour différencier deux objets l'un de l'autre. Il serait donc opportun de proposer une représentation adaptée pour chacune de ces deux tâches. Nous pouvons imaginer qu'une représentation plus fine et qui se concentre sur les parties discriminatives des objets serait utilisée pour l'étape d'identification.

Publications

Les travaux menés au cours de cette thèse ont été publiés dans des conférences et workshops internationaux et ont reçu des distinctions lors de compétitions internationales. Nous les énumérons dans ce qui suit.

Conférences internationales

- H. Harzallah, C. Schmid et F. Jurie. Combining efficient object localization and image classification. IEEE International Conference on Computer Vision, October 2009 (Oral).

- H. Harzallah, F. Jurie et C. Schmid. Vehicle Identification. MCM-ITP conference, June 2009.

Workshops

- H. Harzallah and C. Schmid and F. Jurie and A. Gaidon. Classification aided two stage localization. PASCAL Visual Object Classes Challenge Workshop, in conjunction with ECCV, October 2008.

- M. Marszałek, C. Schmid, H. Harzallah et J. Van de Weijer. Learning Object Representations for Visual Object Class Recognition. PASCAL Visual Object Classes Challenge Workshop, in conjunction with ICCV, October 2007.

Compétitions

- H. Harzallah, C. Schmid, F. Jurie et A. Gaidon. Gagnant de la tâche de localisation d'objets dans les images pour 11 classes parmi 20. PASCAL Visual Object Classes Challenge 2008.

- H. Harzallah, C. Schmid, M. Marszałek et F. Jurie. Mention honorable pour la tâche de localisation d'objets dans les images. PASCAL Visual Object Classes Challenge 2007.

Bibliographie

[1] J. Aghajanian, J. Warrell, S. J. D. Prince, P. Li, J. L. Rohn, and B. Baum. Patch-based within-object classification. In *ICCV*, 2009.

[2] H. Bay, A. Ess, and T. Tuytelaarsand L. Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110 :346–359, June 2008.

[3] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondence. In *CVPR*, pages 26–33, 2005.

[4] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, USA, 1 edition, January 1996.

[5] G. Bouchard and B. Triggs. Hierarchical part-based visual object categorization. In *CVPR*, pages I 710–715, June 2005.

[6] L. Bourdev and J. Brandt. Robust object detection via soft cascade. In *CVPR*, 2005.

[7] S.C. Brubaker, J.X. Wu, J. Sun, M.D. Mullin, and J.M. Rehg. On the design of cascades of boosted ensembles for face detection. *IJCV*, 77(1-3) :65–86, 2008.

[8] P. Carbonetto, N. de Freitas, and K. Barnard. A statistical model for general contextual object recognition. In *ECCV*, 2004.

[9] G. Carneiro and D. Lowe. Sparse flexible models of local features. In *ECCV*, pages 29–43, 2006.

[10] O. Chum and A. Zisserman. An exemplar model for learning object classes. In *CVPR*, 2007.

[11] D. Crandall, P. Felzenszwalb, and D. Huttenlocher. Spatial priors for part-based recognition using statistical models. In *CVPR*, pages 10–17, 2005.

[12] G. Csurka, C. Dance, L. Fan, J. Williamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV workshop on Statistical Learning in ComputerVision*, pages 59–74, 2004.

[13] N. Dalal. *Finding people in images and videos*. PhD thesis, Institut National Polytechnique de Grenoble, july 2006.

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

[15] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, 2006.

[16] C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for multi-class object layout. In *ICCV*, 2009.

[17] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert. An empirical study of context in object detection. In *CVPR*, 2009.

[18] M.M. Dundar and J.B. Bi. Joint optimization of cascaded classifiers for computer aided detection. In *CVPR*, 2007.

[19] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[20] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html.

[21] 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/index.html.

[22] Z. Fan and B. Lu. Fast recognition of multi-view faces with feature selection. In *ICCV*, 2005.

[23] P. Felzenszwalb, D. Mcallester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[24] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32 :1627–1645, 2010.

[25] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient matching of pictorial structures. In *CVPR*, pages 66–73, 2000.

[26] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003.

[27] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *CVPR*, volume 1, pages 380–397, 2005.

[28] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. *PAMI*, 30(1) :36–51, 2008.

[29] M. Fischler and R Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, C22(1) :67–92, Jan 1973.

[30] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1) :119–139, 1997.

[31] M. Fritz and B. Schiele. Decomposition, discovery and detection of visual categories using topic models. In *CVPR*, 2008.

[32] D. Gavrila. Pedestrian detection from a moving vehicle. In *ECCV*, 2000.

[33] H. Harzallah, F. Jurie, and C. Schmid. Combining efficient object localization and image classification. In *ICCV*, 2009.

[34] H. Harzallah, C. Schmid, and F. Jurie. Vehicle identification. In *MCM-ITP conference*, jun 2009.

[35] H. Harzallah, C. Schmid, F. Jurie, and A. Gaidon. Classification aided two stage localization, oct 2008. PASCAL Visual Object Classes Challenge Workshop, in conjunction with ECCV.

[36] G. Heitz and D. Koller. Learning spatial context : Using stuff to find things. In *ECCV*, 2008.

[37] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. In *CVPR*, 2006.

[38] C. Huang, H. Ai, Y. Li, and S. Lao. Vector boosting for rotation invariant multi-view face detection. In *ICCV*, 2005.

[39] A. K. Jain, Yu Zhong, and M. Dubuisson-Jolly. Deformable template models : a review. *Signal Process.*, 71(2) :109–129, 1998.

[40] T. Joachims. Text categorization with support vector machines : Learning with many relevant features. In *European Conference on Machine Learning*, pages 137–142, 1998.

[41] M. J. Jones and P. A. Viola. Fast multi-view face detection. Technical Report TR2003-96, MERL, 2003.

[42] F. S. Khan, J. van de Weijer, A. Bagdanov, N. Elfiky, D. Rojas, M. Pedersoli, X. Boix, P. Gonfaus, H. salahEldeen, R. Benavente, J. Gonzalez, and M. Vanrell. Cvc classification approach, oct 2009. PASCAL Visual Object Classes Challenge Workshop, in conjunction with ICCV.

[43] J. Kittler, M. Hatef, R. P.W. Duin, and J. Matas. On combining classifiers. *PAMI*, 20(3) :226–239, 1998.

[44] S. Kumar and M. Hebert. A hierarchical field framework for unified context-based classification. In *ICCV*, 2005.

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

[46] I. Laptev. Improvements of object detection using boosted histograms. In *BMVC*, 2006.

[47] D. Larlus. *Création et utilisation de vocabulaires visuels pour la catégorisation d'images et la segmentation de classes d objets*. PhD thesis, INPG, nov 2008.

[48] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features : Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[49] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Workshop on Statistical Learning in Computer Vision, ECCV*, 2004.

[50] B Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *IJCV*, 77(1-3), 2008.

[51] B. Leibe and B. Schiele. Scale-invariant object categorization using a scale-adaptive mean-shift search. In *DAGM04*, 2004.

[52] L. Li and L. Fei-Fei. What, where and who ? classifying events by scene and object recognition. In *ICCV*, 2007.

[53] S. Z. Li and Z. Zhang. Floatboost learning and statistical face detection. *PAMI*, 26(9) :1112–1123, 2004.

[54] J. Liebelt and C. Schmid. Multi-view object class detection with a 3d geometric model. In *CVPR*, pages 1688–1695, jun 2010.

[55] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30 :79–116, 1998.

[56] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2) :91–110, 2004.

[57] S. Maji and A. C. Berg. Max-margin additive classifiers for detection. In *ICCV*, 2009.

[58] M. Marszałek, C. Schmid, H. Harzallah, and J. van de Weijer. Learning object representations for visual object class recognition. In *Visual Recognition Challange workshop, in conjunction with ICCV*, 2007.

[59] K. Mikolajczyk. Multiple object class detection with a generative model. In *CVPR*, pages 26–36, 2006.

[60] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1) :63–86, 2004.

[61] M. Minsky. Steps toward artificial intelligence. In *Computers and Thought*, pages 406–450. McGraw-Hill, 1961.

[62] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29(1) :51–59, January 1996.

[63] A. Opelt, A. Pinz, and A. Zisserman. Learning an alphabet of shape and appearance for multi-class object detection. *IJCV*, 80(1) :16–44, 2008.

[64] C. Papageorgiou and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38(1) :15–33, 2000.

[65] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 1999.

[66] F. Porikli. Integral histogram : a fast way to extract histograms in cartesian spaces. In *CVPR*, 2005.

[67] C. Schmid and Roger Mohr. Local greyvalue invariants for image retrieval. *PAMI*, 19 :530–535, 1997.

[68] B. Scholkopf and A. J. Smola. *Learning with Kernels : Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[69] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *CVPR*, 2008.

[70] Techcrunch.com. 2 billion photos on flickr. http://techcrunch.com/2007/11/13/2-billion-photos-on-flickr/, 2007.

[71] Techcrunch.com. Flickr hits its 5 billionth photo. http://techcrunch.com/2010/09/18/flickr-5-billionth-photo/, 2010.

[72] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiel, and L. Van Gool. Towards multi-view object class detection. In *CVPR*, 2006.

[73] A. Torralba. Contextual priming for object detection. *IJCV*, 53(2) :169–191, 2003.

[74] A. Torralba, K. Murphy, and W. Freeman. Sharing features : efficient boosting procedures for multiclass object detection. In *CVPR*, 2004.

[75] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *CVPR*, pages 586–591. IEEE Comput. Sco. Press, 1991.

[76] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluation of color descriptors for object and scene recognition. In *CVPR*, 2008.

[77] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *PAMI*, 2010.

[78] V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[79] V. N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, September 1998.

[80] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009.

[81] P. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 57(2) :137–154, 2004.

[82] P. A. Viola and M. J. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.

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

[84] M. Weber, M. Welling, and P. Perona. Towards automatic discovery of object categories. In *CVPR*, 2000.

[85] L. Wolf and S.M. Bileschi. A critical view of context. *IJCV*, 69(2) :251–261, 2006.

[86] B. Wu and R. Nevatia. Cluster boosted tree classifier for multi-view, multi-pose object detection. In *ICCV*, 2007.

[87] A.L. Yuille. Deformable templates for face recognition. *Journal of Cognitive Neuroscience*, 3(1), 1991.

[88] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories : a comprehensive study. *IJCV*, 73(2) :213–238, 2007.

[89] Q. Zhu, S. Avidan, M. C. Yeh, and K. T. Cheng. Fast human detection using a cascade of histograms of oriented gradients. In *CVPR*, 2006.