Création et utilisation de vocabulaires visuels pour la catégorisation d'images et la segmentation de classes d'objets

Diane Larlus

INSTITUT POLYTECHNIQUE DE GRENOBLE

THÈSE

pour obtenir le grade de

DOCTEUR DE L'INSTITUT POLYTECHNIQUE DE GRENOBLE

Spécialité : Mathématiques et Informatique

préparée au Laboratoire Jean Kuntzmann

dans le cadre de l'École Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique

présentée et soutenue publiquement

par

Diane Larlus

le 28 novembre 2008

Création et utilisation de vocabulaires visuels pour la catégorisation d'images et la segmentation de classes d'objets

---

Directeur de thèse : Pr. Frédéric Jurie

JURY

Pr. Roger Mohr	Président
Pr. Frédéric Jurie	Directeur de thèse
Pr. Patrick Perez	Rapporteur
Pr. Jean-Pierre Cocquerez	Rapporteur
Dr. Stephane Herbin	Examinateur
Pr. Matthieu Cord	Examinateur

Résumé

Ce manuscrit s'intéresse à l'interprétation d'images fixes et en particulier à la reconnaissance de classes d'objets. Les différentes approches considérées sont toutes des variations du modèle par sac-de-mots, utilisant des représentations locales.

Nous débutons ce manuscrit par l'étude de différentes méthodes de création du vocabulaire visuel et par l'évaluation de ces vocabulaires dans le contexte de la catégorisation d'images.

La première méthode proposée s'intéresse aux représentations denses d'images. Les descripteurs ainsi extraits sont quantifiés en mots visuels à l'aide d'un algorithme de clustering à deux phases qui tient compte des données. Une étude paramétrique complète de cette approche est réalisée.

La deuxième méthode de création de vocabulaires visuels que nous présentons intègre les vocabulaires au modèle de représentation des images. Ce modèle génératif utilise les labels de classes des images à travers des variables latentes d'aspect. L'estimation du modèle entraîne la création de mots visuels compacts et discriminants.

Enfin, nous montrons qu'il est possible de remplacer les vocabulaires visuels classiques (comme ceux introduits précédemment) par des arbres de décision aléatoires. Chaque arbre propose une quantification de l'espace de représentation des descripteurs en mots visuels. Construits à partir des labels de classes, ils sont discriminants. Des classifieurs simples sont utilisés pour chaque nœud, ce qui permet un traitement rapide des images de test. Les arbres aléatoires sont également appliqués à la création en ligne de cartes de saillance qui guident le procédé d'échantillonnage des descripteurs.

Dans la deuxième partie du manuscrit, nous étudions deux méthodes de segmentation de classes d'objets.

La première méthode est basée sur un modèle à variables latentes d'aspect étendu. Au lieu de considérer les aspects à l'échelle de l'image, cette méthode les modélise à l'échelle de sous-régions. Ces régions semi-globales sont d'intersection non vide et partagent ainsi de l'information, ce qui permet une régularisation des labels locaux. Les classifications locales sont fournies par des statistiques sur des mots visuels.

La deuxième méthode de segmentation combine les propriétés de régularisation très locales permises par un champ de Markov avec un modèle d'apparence qui fournit des contraintes à plus grande échelle. Le modèle d'apparence est basé sur des régions qui représentent chacune un objet et qui sont des collections de mots visuels. Nous étudions également l'utilisation d'arbres de décision à la place des mots.

Pour finir, cette dernière méthode est appliquée à un problème concret de recherche visuelle, pour une plateforme robotique, en générant des hypothèses sur la position de l'objet dans le champ de vision du robot.

Abstract

THIS thesis deals with the interpretation of static images, with a focus on recognising object categories. We consider several different approaches, which are all variations on the bag-of-words model, and all use local image descriptors.

The first part of the thesis examines different methods for creating visual vocabularies. We aim to create vocabularies which perform well for image categorisation. The first method proposed uses dense image representations. Feature descriptors are extracted and then quantised to visual words, using a two-stage clustering algorithm. We provide a full quantitative evaluation of the method.

The second method we propose for creating visual vocabularies integrates the vocabulary into an image representation model. This generative model uses the image class labels via latent variables describing object aspect. Training the model leads to the creation of a compact and discriminative set of visual words.

Next we show that traditional visual vocabularies (like the ones used above) can be replaced by random decision trees. Each tree provides a quantisation of the space of descriptor representations into visual words. Since the trees are constructed using the image class labels, they have good classification performance. Each node uses a simple classifier, so processing of test images is fast. The random trees are also used for online learning of saliency maps which guide the process of descriptor sampling.

The second part of the thesis deals with object category segmentation. We first present a method which uses an extended latent aspect based model. Instead of considering aspects at the image level, the method models them at the sub-region level. These semi-global regions can overlap and share information, allowing the local predictions to be improved. The local classifications are based on visual word statistics.

The second segmentation method combines the low-level consistency properties of a Markov Random Field with an appearance model which provides higher-level constraints. The appearance model is based on regions which each represent a single object as a set of visual words. We also evaluate using decision trees instead of visual words.

Finally, the method is applied to a real-world visual search problem for a humanoid robot. The method is used to generate hypotheses about the position of an object in the robot's field of view.

Remerciements

Je tiens à remercier en tout premier lieu Frédéric Jurie qui a dirigé cette thèse dans la continuité de mon stage de master. Tout au long de ces trois années, il m'a apporté son aide et ses précieux conseils, même après son départ pour l'université de Caen. Je le remercie vivement.

Je remercie les rapporteurs de cette thèse, Patrick Perez et Jean Pierre Cocquerez, pour l'intérêt qu'ils ont porté à mon travail. Merci également aux autres membres du jury qui ont accepté de juger ce travail : Stephane Herbin, Matthieu Cord et Roger Mohr.

L'équipe LEAR est un cadre privilégié pour effectuer une thèse. Parmi ceux qui ont contribué à mes réflexions, je remercie tout spécialement Jakob Verbeek, mais aussi Cordelia Schmid, Bill Triggs et Hervé Jegou.

J'ai eu également le plaisir de collaborer avec le laboratoire JRL de Tsukuba. Je pense en premier lieu à Olivier Stasse qui a initié et soutenu notre collaboration ainsi qu'à toutes les personnes formidables que j'ai rencontrées là-bas.

Je n'oublierai pas les aides permanentes reçues du personnel administratif et plus particulièrement d'Anne Pasteur.

Adrien, Benoît, Carole, Charlotte, Marc, Marie Thé, Matthieu, Moray, Rémi et Sonia ont relu attentivement des parties de ce manuscrit. Quelques lignes pour certains, la totalité pour Marc, merci du temps qu'ils m'ont consacré.

Une pensée émue pour tous les gens que j'ai eu le plaisir de rencontrer à l'INRIA Grenoble et autour. Ils sont trop nombreux pour être cités ici, mais ils ont contribué à rendre ces trois années très agréables.

Enfin, pour son soutien sans faille et permanent, je tiens à remercier Marc.

Grenoble, le 21 octobre 2008.

Table des matières

Liste des figures

Liste des tables

Introduction et positionnement de la thèse

1

Sommaire

Ce chapitre décrit les motivations scientifiques et le positionnement de cette thèse. Au cours de ce chapitre, nous verrons que les grandes quantités d'images qui sont créées et utilisées dans notre vie quotidienne posent de véritables défis à la communauté des chercheurs en vision par ordinateur. Nous verrons également que les pixels de ces images ne sont pas utilisés directement mais que des représentations combinant des groupes de pixels sont plus efficaces. Nous introduirons la tâche de reconnaissance de catégories, ainsi que la notion d'apprentissage dont nous aurons besoin pour la résoudre. Enfin nous définirons les tâches de catégorisation d'images et de segmentation de catégories d'objets, ainsi que les travaux qui ont inspirés nos contributions.

Récemment, les systèmes d'acquisition de supports numériques, en particulier pour les images et les vidéos, se sont fortement démocratisés, grâce à une miniaturisation et à une baisse des coûts significative. Le stockage de toute cette information est également facilité. Enfin, l'hyper-connexion et le développement de réseaux sociaux en ligne puissants (comme les sites communautaires) rendent possible l'accès à une quantité de données numériques dont le volume a explosé ces dernières années. La question est maintenant de se retrouver dans cet immense amas de données. Pour organiser et référencer ce contenu, il faut être en mesure de l'interpréter, de lui donner un sens. Cette tâche qui semble pourtant si aisée pour un être humain pose des problèmes à ce jour insurmontables par une machine. Pourtant, seule une automatisation de l'interprétation rendrait possible l'exploitation de ces images numériques disponibles en si grand nombre.

1.1 Positionnement du problème

1.1.1 Reconnaissance d'objets

Le champ de l'interprétation des images se verra réduit ici à celui de la reconnaissance, en particulier la reconnaissance des objets. La reconnaissance est « *l'action par laquelle on retrouve dans sa mémoire l'idée, l'image d'une chose ou d'une personne quand on vient à la revoir* » [1]. Cela nous place dans une configuration où une connaissance est supposée acquise, connaissance qui est ensuite utilisée pour effectuer cette action de reconnaissance. L'apprentissage automatique (ou *machine learning*) semble particulièrement adapté à cette problématique. La connaissance est acquise à partir d'un ensemble d'exemples dans une étape d'apprentissage, qui construit automatiquement un modèle de l'objet à reconnaître. Nous reviendrons plus en détail sur l'apprentissage dans la section 1.2.

Apprendre à reconnaître une unique instance d'objet n'est pas suffisant pour bon nombre de problèmes, nous nous intéresserons ici à la reconnaissance des objets à l'échelle de la catégorie. Cependant, envisager la reconnaissance de catégories d'objets soulève de nombreuses questions. La première est celle de la définition et de l'organisation de ces catégories. Par exemple, la chèvre de M. Seguin est une chèvre, mais aussi un animal domestique, un animal à cornes, un quadrupède, etc. La question du bon découpage des catégories est difficile à aborder. Nous supposerons par la suite que les catégories sont définies par l'exemple, au moyen de listes d'images (chaque image est étiquetée comme positive ou négative pour chaque catégorie d'intérêt, en fonction de son appartenance à la catégorie).

Même une fois la catégorie précisément définie, les objets qu'elle contient peuvent avoir des apparences très variées (voir l'illustration de la figure 1.1). Ces variations peuvent être dues à des changements d'échelle ou de points de vue, des occultations partielles ou encore des changements d'illumination. Mais plus encore, les apparences de deux objets de la même catégorie peuvent être très différentes. Souvent, les catégories sont définies de façon fonctionnelle, plutôt que visuelle. Par exemple, un objet sera classé comme une chaise à partir du moment où il a pour fonction de permettre de s'asseoir, disposera en général d'un dossier, mais ne possédera pas forcément quatre

[1] http ://fr.wiktionary.org/wiki/reconnaissance

FIG. 1.1 – *Gauche : exemples de variation que peuvent subir les images d'une catégorie donnée. Droite : cet objet a une apparence étrange, mais il appartient bien à la catégorie chaise de par sa fonction.*

pieds (voir figure 1.1). Les modèles des objets que nous allons créer devront être capables de faire face à toutes ces variations d'apparence. De plus, le fond sur lequel se présente l'objet peut être encombré et varier fortement, compliquant encore la reconnaissance de l'objet.

La reconnaissance d'objets peut prendre différentes formes. En général, trois tâches sont distinguées. La *catégorisation d'images* (ou classification d'images) consiste à donner un label à une image de test en fonction de la présence ou non d'un objet appartenant à une catégorie donnée. Cette tâche peut être généralisée à la prédiction d'un label indiquant la nature de l'objet parmi une liste exhaustive. La *détection d'objets* désigne la tâche de localisation approximative (rectangle englobant l'objet) des objets d'une catégorie donnée. Enfin la *segmentation de classes d'objets* consiste à déterminer quels sont les pixels de l'image qui appartiennent à un objet d'une des classes d'intéret. Cela revient à classifier les régions unitaires de l'image (en général les pixels) comme appartenant à un objet d'une des catégories considérées ou non. La tâche de segmentation au sens large a parfois une définition différente sur laquelle nous reviendrons plus tard.

Ces trois tâches sont étroitement liées, et les mêmes outils peuvent être mis en œuvre pour les résoudre : extraction de primitives visuelles, représentation locales des images, construction de modèles, appariement entre modèles et images, classification, etc.

1.1.2 Représentation des images

La question de la représentation des objets est cruciale. Il est évidemment peu judicieux d'utiliser directement les valeurs de pixels de l'image, qui peuvent être très différentes pour des images presque identiques. Prenons pour exemple la figure 1.2. Si on compare les images pixel à pixel, lorsque l'objet est légèrement translaté, la représentation devient complètement différente. Et même si la mise en correspondance est parfaite, il suffit de remplacer l'objet par un objet presque identique pour que la valeur des pixels soit entièrement modifiée.

La représentation d'images a fait l'objet de beaucoup de contributions ces dernières années. En effet, plus la représentation est adaptée, plus les règles de décision dans cet espace sont faciles à définir. Nous ne nous intéresserons pas ici aux méthodes qui utilisent un modèle 3D de l'objet qui doit ensuite être associé à la projection 2D dans l'image, mais uniquement aux méthodes

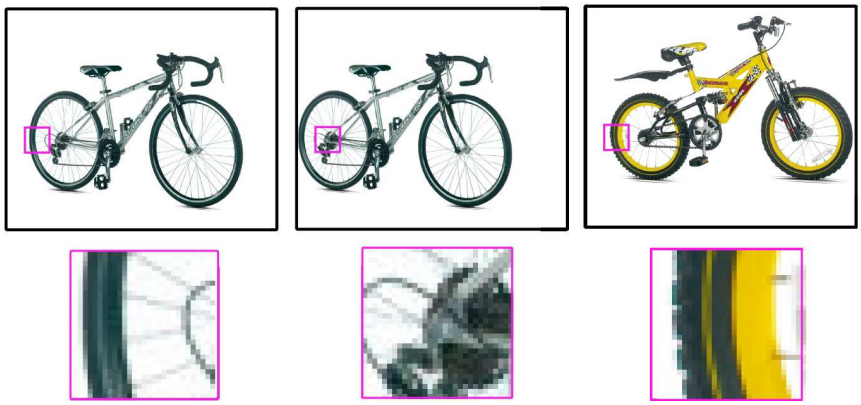


FIG. 1.2 – *Les pixels sont une très mauvaise représentation pour comparer la similarité entre deux images.*

qui apprennent des modèles statistiques à partir d'images d'apprentissage 2D.

Nous distinguons deux types d'approche. Historiquement les méthodes globales ont été proposées en premier. Elles calculent une signature de l'image dans sa globalité, à l'aide de différents descripteurs basés par exemple sur la couleur ou bien encore la texture. Ces méthodes sont relativement simples à utiliser. En fonction du descripteur choisi, elles peuvent être robustes à certaines variations comme l'illumination, ou le contraste. Pour s'avérer robustes aux changements de points de vue et d'échelle, elles nécessitent une quantité gigantesque d'images d'apprentissage. Enfin, elles ne sont pas du tout adaptées aux occultations et à la présence de fonds encombrés.

C'est pour palier à ces limitations très contraignantes que les méthodes locales ont été proposées [76]. Elles considèrent les images comme des collections de régions d'intérêt locales, généralement de taille assez faible par rapport à la taille totale de l'image. Elles sont appelées *patchs*. Seules les régions considérées comme importantes ou saillantes sont utilisées, ce qui permet de limiter la quantité d'information à manipuler. Par leur nature locale et leur capacité à permettre aisément les appariements entre images, ces méthodes conduisent à des algorithmes robustes aux translations et aux changements d'échelle, aux occultations, et à la présence d'un fond difficile, qui sont des caractéristiques classiques des images en condition réelle. C'est pourquoi elles sont majoritaires parmi les méthodes de reconnaissance d'instance d'objet (par mise en correspondance de primitives locales) et de catégories. Pour toutes ces raisons, les méthodes que nous allons développer reposent sur l'utilisation de descripteurs locaux.

Les représentations locales reposent sur deux étapes importantes :

- ▷ le choix des positions et échelles des patchs extraits,
- ▷ la conversion des pixels de ces patchs en une signature locale de l'image, appelée descripteur.

La sélection des patchs peut se faire selon différentes stratégies. Parmi les plus populaires, nous pouvons citer l'extraction dense (grille en position et en échelle), l'extraction aléatoire, et l'utilisation de différents critères de saillance comme les points d'intérêt. Concernant le descripteur, de très nombreux types d'indices visuels ont été proposés. Nous pouvons citer, par exemple, les contours, les textures, la couleur, etc.

Une fois que les descripteurs locaux sont extraits de l'image à classer, il

s'agit de les utiliser pour trouver la catégorie de l'image. C'est le mécanisme de prédiction de la classe. Il n'est pas très efficace de devoir comparer la nouvelle image à toutes celles existantes pour prendre une décision, ce qui fait que la construction d'un modèle est souvent préférée. Le modèle par *sac-de-mots* est devenu particulièrement populaire durant ces dernières années, en raison de la qualité des résultats qu'il permet d'obtenir. Il s'agit d'une approche initialement développée pour la catégorisation de textes, domaine dans lequel elle s'est avérée très performante [33]. Chaque document est représenté par un histogramme basé sur la fréquence d'apparition de chaque mot du vocabulaire. Les histogrammes subissent diverses normalisations, visant à les affranchir de la taille du document ainsi qu'à amplifier les mots discriminants et/ou fréquents. Différentes stratégies peuvent être ensuite utilisées pour la classification des documents. Nous notons en particulier les deux familles d'approche les plus typiques que sont la classification par *maximum a posteriori* (MAP) ou *maximum de vraisemblance* (ML) [30] impliquant des modèles génératifs des documents ou la classification par recherche de fonctions discriminantes notamment au moyen de classifieurs de machines à vecteurs de support (SVM) ou des k plus proches voisins (KNN) [30]. Les modèles génératifs comme les modèles génératifs sont construits à partir d'exemples.

Ce type d'approche par *sac-de-mots* peut être transposé au cas de la catégorisation d'images [13]. Dans ce cas, les images sont caractérisées par un histogramme qui compte le nombre d'occurrences de chaque classe de représentations locales, que nous appellerons *mots visuels*, par analogie. Reste à définir ce qu'est une classe de représentation locale, et cela occupera une place importante dans cette thèse.

Bien entendu, contrairement au cas de l'analyse de documents, le vocabulaire visuel n'est pas une donnée intrinsèque aux images. Il n'existe pas de vocabulaire unique pour décrire les descripteurs de patchs. Ce vocabulaire doit être construit pour répondre à des attentes particulières. Ces mots sont créés par quantification vectorielle, c'est-à-dire en transformant l'espace continu des descripteurs de patchs locaux en un ensemble discret de *clusters* qui représentent les *mots visuels*. Les images sont donc vues comme des collections de mots visuels au même titre que les textes sont vus comme des collections de mots. Ensuite, les méthodes de classification de texte basées sur les mots deviennent applicables.

Cette représentation d'images par sac-de-mots s'est avérée très efficace pour la classification d'images. Sa principale force réside dans la représentation compacte associée à chaque image : un histogramme de comptage dont la dimension est égale à la taille du vocabulaire visuel. Cette représentation est souple, et intrinsèquement assez robuste aux variations d'apparence des objets mentionnées plus haut.

Cependant, en pratique, les mots visuels ne sont pas aussi précis que les mots textuels. Il est impossible de créer des mots qui soient toujours observés sur la même partie d'un objet et jamais ailleurs. Mais l'accumulation de statistiques sur l'ensemble de l'image rend l'approche robuste à ces imprécisions.

Par contre, comme ces méthodes n'utilisent pas de modèle géométrique des objets ; elles ne considèrent aucune contrainte sur la position ou l'ordre d'apparition des mots.

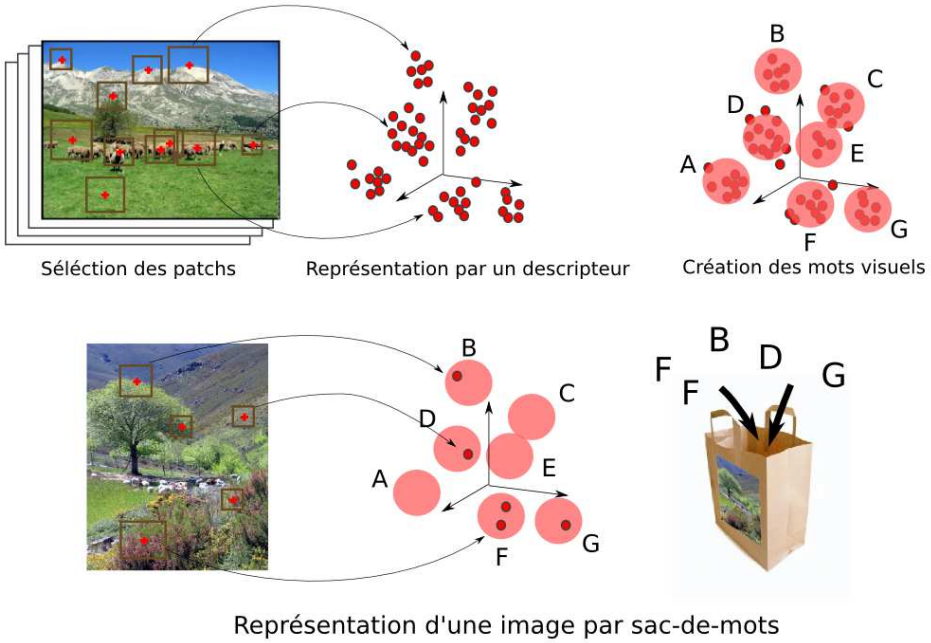


FIG. 1.3 – *Représentation sac-de-mots. Pendant l'apprentissage, un vocabulaire visuel est créé par quantification vectorielle de l'ensemble des descripteurs de patchs des images d'apprentissage. Ensuite, une image est représentée par ses occurrences de mots visuels.*

Certaines approches par patch, incorporent de la géométrie entre les mots, comme celle de Leibe et Schiele [51] ou bien encore celle de Agarwal *et al* [1]. La première utilise un système de votes probabilistes sur le centre de l'objet, par tous les patchs correspondant à un mot visuel d'objet. La deuxième comporte un système de relations entre paires de patchs encodant la distance et l'orientation observée pour la paire. Lorsqu'un modèle est reconnu, il gagne en fiabilité. Ces méthodes permettent ainsi d'éviter certains faux-positifs. Cependant, elles sont dépendantes du point de vue, puisque la géométrie est rigide.

C'est pourquoi nous avons choisi de ne pas utiliser de tels modèles géométriques très contraints. Lorsque des informations de géométrie seront nécessaires, pour les tâches de détection et de segmentation, nous utiliserons des contraintes géométriques beaucoup plus flexibles (travaux présentés dans les chapitres 5, 6 et 7).

Un autre caractéristique des méthodes par sac-de-mots est leur forte dépendance au choix du vocabulaire. Nous avons vu qu'il est créé de façon artificielle par une quantification de l'espace des descripteurs. À l'origine, cette quantification était faite de façon indépendante de la tâche de reconnaissance. Mais un vocabulaire mieux adapté permet une meilleure reconnaissance. Parmi les travaux qui ont inspiré notre étude, il y a tout d'abord ceux de Jurie et Triggs [36], qui montrent que les descripteurs locaux d'images, extraits de façon très dense dans les images, comportent des propriétés statistiques de peuplement de l'espace très particulières, qui sont à prendre en compte lors de la création du vocabulaire. Nous avons donc proposé une méthode de clustering pour créer des vocabulaires visuels qui soient adaptés aux spécificités des données denses. C'est l'objet des travaux du chapitre 2.

Ensuite, d'autres travaux se sont intéressés à la création de vocabulaires visuels discriminants. Parmi les plus marquants, citons les travaux de Winn *et al* [98] et ceux de Perronnin *et al* [70]. Dans [98], les auteurs suggèrent de construire un vocabulaire visuel compact et plus discriminant en regroupant des paires de mots visuels à partir d'un vocabulaire visuel initial très large. Cependant, si deux descripteurs sont regroupés initialement dans le même cluster, ils ne peuvent être séparés ultérieurement. Dans [70], un vocabulaire universel est combiné avec des vocabulaires spécifiques aux classes. Seulement, le but n'est pas de mettre en valeur les différences entre les classes, mais les différences entre l'apparence moyenne encodée par le vocabulaire universel et l'apparence d'une classe donnée. Nous avons également étudié les possibilités de création d'un vocabulaire discriminant qui ne souffre pas des limitations relevées pour ces deux méthodes. C'est l'objet du chapitre 3.

Enfin les travaux de Marée *et al* [55] proposent d'utiliser des forêts aléatoires pour quantifier l'espace des descripteurs en un ensemble discret de feuilles, et ainsi classifier chaque patch d'image à l'aide du label associé à la feuille. Chaque patch vote ensuite individuellement, et la classe majoritaire l'emporte. Une telle quantification des données est très efficace, puisque l'association d'un patch à une feuille est très rapide. De plus elle a l'avantage d'utiliser les informations de classe pour guider la position des frontières de décision. Les feuilles peuvent donc être vues comme des mots visuels, intrinsèquement discriminants. Cependant, il n'y pas de représentation globale de l'image comme c'est le cas avec l'histogramme de la méthode par sac-de-mots. Une telle représentation permettrait d'apprendre l'importance des différentes décisions prises par les feuilles (au lieu d'avoir des votes équi-importants), comme cela est fait lors de l'entraînement d'un classifieur sur les histogrammes d'occurrence de mots visuels. Cette possibilité est étudiée chapitre 4. Les forêts aléatoires sont également utilisées en remplacement des mots visuels pour la méthode proposée dans le chapitre 6.

1.1.3 Lien avec la segmentation

Quel que soit le modèle utilisé, il est bien sûr plus facile de reconnaître un objet quand les informations, dans notre cas les patchs locaux, proviennent uniquement de l'objet à reconnaître et non du fond. Lorsqu'un fond encombré est présent dans l'image, cela augmente le risque d'incorporer des patchs de fond dans le modèle d'objet. C'est pour cette raison que sur les premières bases de reconnaissance d'objets, puis de reconnaissance de catégories d'objets, les objets étaient représentés sur un fond uniforme. Puis des premières bases se sont intéressées aux images en condition réelle, mais avec toujours l'objet constituant la plus grosse partie de l'image (Caltech 4 [20]) ce qui limite l'influence du fond. De nos jours, les objets à reconnaître ne constituent plus l'élément principal de l'image. Mais pour les premières bases de ce type, les objets étaient localisés dans les images, par exemple par un masque de segmentation. Cette information est utilisée lors de l'apprentissage, pour limiter l'influence du fond. C'est le cas par exemple pour les premiers résultats sur la base TU-Graz02 [68].

Puis des méthodes ont été développées pour apprendre à reconnaître et à localiser les objets dans les images, sans utiliser de segmentation pour

FIG. 1.4 – *Le dalmatien photographié par R. C. James : un système difficile à reconnaître, même pour l'œil humain*

l'apprentissage. Et pendant longtemps, la reconnaissance a été traitée sans la segmentation.

Parallèlement à ce travail sur la reconnaissance d'objets, et prenant racine dans des travaux bien plus anciens, tout un pan de la vision par ordinateur s'est intéressé à la tâche de segmentation d'images. Cette tâche consiste à regrouper les pixels en régions qui ont des propriétés communes, parce qu'elles représentent un objet, ou une partie d'objet, ou encore l'entité sémantique d'une scène. Des méthodes utilisant les contours, l'apparence des régions, des graphes, etc. ont obtenu des résultats de plus en plus pertinents, bien qu'il soit difficile de définir avec précision la justesse d'une segmentation.

Il faut aussi s'interroger sur les limites de méthodes de segmentation qui n'utilisent aucune interprétation du contenu de la scène. En effet les tâches de reconnaissance et de segmentation sont étroitement liées. Cela n'est pas sans rappeler le paradoxe de Kenneth M. Sayre tel qu'il l'avait noté pour la reconnaissance de caractères. Il s'énonce comme suit : la segmentation ne peut être faite qu'une fois la reconnaissance effectuée, et la reconnaissance n'est possible qu'après la segmentation de l'objet[2] [75]. Cette réflexion peut évidemment être appliquée au problème de la reconnaissance et de la segmentation de catégories d'objets. Prenons par exemple la célèbre photographie de R. C. James, présentée figure 1.4. Une segmentation convenable ne peut être obtenue qu'une fois l'animal identifié, et réciproquement, l'animal ne peut être reconnu que lorsque nous avons été capables de deviner les contours qui le délimitent.

Cette réflexion sur le fait qu'une bonne segmentation dépend du contenu de l'image, est à l'origine par exemple des travaux de Martin *et al* [57], dans lesquels une phase d'apprentissage est utilisée pour apprendre ce qu'un être humain considère comme une bonne segmentation. Mais plus pertinentes par rapport à notre travail, des méthodes traitant simultanément de la segmen-

[2]Le passage exact : « *Identification of letters within a cursive line requires locating the beginning and the end points of individual letter-inscriptions. It is common to think of this as a task to be accomplished before the individual inscriptions can be recognized. But this is paradoxical, since the individual letters must be recognized as such before it can be determined where one inscription leaves off and another begins.* »

tation et de la reconnaissance sont apparues tout récemment. Le premier travail notable est celui de Leibe et Schiele [51], où des segmentations sont utilisées pour apprendre un modèle de l'objet. Au moment de la reconnaissance, les parties apprises comme appartenant à l'objet sont utilisées pour produire une segmentation de l'objet dans les images de test. La méthode n'est applicable que pour une seule classe d'objet à la fois, une seule vue de l'objet et pour des objets supposés rigides. Mais c'est une des premières fois que ces deux problèmes sont traités simultanément. La résolution conjointe de ces deux problèmes nous intéressera tout particulièrement dans ce manuscrit.

Ces travaux ont influencé beaucoup de méthodes. Citons parmi celles qui nous ont le plus inspirés, les travaux de Shotton *et al* [80]. Leur méthode, Textonboost, produit des segmentations de scènes, telles que les pixels sont labellisés comme appartenant à une des catégories d'objets considérées, ou à l'une des classes de fond. Les résultats obtenus sont très prometteurs. Cependant, cette méthode n'est pas capable de modéliser différentes instances de la même catégorie dans une image donnée. Cela induit une perte de précision dans la modélisation.

Les chapitres 5 et 6 proposent des méthodes de segmentation adaptées aux cas où les objets présentent une très forte variation d'apparence. La reconnaissance des différents objets présents dans l'image permet de guider la tâche de segmentation, et réciproquement. D'autre part, la tâche de segmentation telle que nous l'avons définie peut être évaluée de façon objective dès lors que la liste des pixels appartenant aux objets est supposée connue.

1.2 Apprentissage

Comme nous venons de le voir, il apparaît séduisant que les méthodes de reconnaissance soient capables d'apprendre automatiquement les modèles des objets à reconnaître. Cela fait l'objet d'une phase préliminaire à la reconnaissance, dite phase d'apprentissage.

1.2.1 Apprentissage et supervision

La reconnaissance de forme est l'action de prédire le label associé à une donnée d'entrée. Par exemple pour la catégorisation d'images, le label correspond à une classe d'objet, et l'entrée est l'image. Pour réaliser cette action, il est possible d'utiliser des règles ou des heuristiques choisies à la main. Mais cela présente de grandes limitations en pratique : la prolifération de règles et d'exceptions, ainsi qu'une mauvaise généralisation, ce qui conduit à de mauvais résultats. L'approche par apprentissage automatique (*machine learning*) au contraire, suppose l'utilisation d'un ensemble de données d'apprentissage, constitué d'exemples, permettant d'estimer les paramètres optimaux d'un modèle adaptatif, capable de prédire les labels. Cette approche est bien meilleure que celle qui utilise des heuristiques, puisqu'elle permet une meilleure généralisation et l'adaptation à chaque base d'apprentissage proposée.

Nous avons longuement parlé de la représentation des images dans la section précédente. Cette partie est souvent appelée extraction de primi-

tives (*feature extraction*) dans la littérature de l'apprentissage automatique et consiste à choisir une représentation adaptée au problème à résoudre.

Différents types d'apprentissage sont en général distingués, en fonction de la nature des données d'apprentissage.

- ▷ Dans l'*apprentissage supervisé*, un algorithme dispose d'une base d'apprentissage pour laquelle chaque exemple est associé à un label. C'est la configuration la plus classique pour la tâche de classification.
- ▷ Pour l'*apprentissage non-supervisé*, un ensemble de données est modélisé, sans qu'aucun de ces exemples ne soient labellisés.
- ▷ L'*apprentissage semi-supervisé* combine des exemples labellisés et des exemples non labellisés.
- ▷ Pour l'*apprentissage par renforcement*, un algorithme apprend à l'aide d'un retour sur chaque action effectuée par celui-ci.
- ▷ Enfin, lors de l'*apprentissage transductif*, les données de test sont utilisées en même temps que les données d'apprentissage pour entraîner le modèle.

Dans les différents chapitres de ce manuscrit, nous rencontrerons différents types d'apprentissage, selon la tâche considérée. Lorsque nous nous intéresserons à la classification des images, nous adopterons une approche utilisant un apprentissage supervisé : les images d'apprentissage sont supposées être labellisées comme contenant ou non l'objet qui nous intéresse. Au contraire, la construction de vocabulaires visuels est généralement faite de façon non-supervisée, les mots visuels sont extraits à partir de l'ensemble des patchs d'apprentissage sans qu'aucun label n'indique quel patch appartient à quel mot. C'est cette approche classique que nous adopterons pour la construction des vocabulaires proposés dans le chapitre 2. Puis nous remettrons en question cette méthodologie, dans le chapitre 3, en utilisant de l'information, venant au niveau global de l'image, pour regrouper les patchs en mots. Lors de cette étape, les images de tests sont considérées simultanément, ce qui fait que la construction du modèle se fait de façon transductive. Enfin pour la construction des arbres de décision qui remplacent les vocabulaires visuels (chapitres 4 et 6), nous utiliserons, lorsqu'elle est disponible, l'information sur les labels au niveau du patch, ce qui donne une classification supervisée.

Compromis biais-variance. Une difficulté classique rencontrée lors de l'apprentissage d'un modèle, est celui du *sur-apprentissage*. Lorsque le modèle est trop complexe ou trop flexible, il s'attarde sur les spécificités des données d'apprentissage, ce qui entraîne une forte variance. En effet, le même modèle appris sur différentes données d'apprentissage donnera des paramètres très différents. Au contraire, si le modèle est trop contraint, il n'aura quasiment aucune variance mais un fort biais. La modélisation n'est pas très précise. Dans les deux cas (trop fort biais, ou trop forte variance) les performances de classification chutent. Il faut trouver un compromis.

1.2.2 Génératif *vs* Discriminatif

Les modèles d'apprentissage sont en général classifiés en *modèles génératifs* et *modèles discriminatifs*. Ce sont deux approches différentes, correspondant à deux visions différentes de l'apprentissage. Un modèle génératif suppose que les variables observées sont générées à partir de paramètres non obser-

vés et spécifie la probabilité jointe entre les observations et les labels. Les modèles génératifs sont utilisés pour modéliser directement les données. La probabilité conditionnelle d'une prédiction de label peut être ensuite obtenue en utilisant la règle de Bayes. Les modèles discriminatifs quant à eux modélisent la dépendance d'une variable de label sur les variables observées. Cela est fait en modélisant directement la distribution de probabilité du label sachant l'observation. Il existe une autre catégorie de modèles discriminatifs, que l'on appelle plus souvent fonctions discriminatives. Ces fonctions associent directement un label à chaque entrée. Les probabilités ne jouent dans ce cas aucun rôle.

Les modèles génératifs modélisent la probabilité jointe de toutes les variables, alors que les modèles discriminatifs s'intéressent uniquement à la variable cible (ici le label à prédire) conditionnée par les variables observées. Ainsi, ces types d'approches possèdent chacun leurs avantages et inconvénients. Les modèles génératifs permettent une vraie modélisation de chaque classe et le modèle obtenu peut être appris indépendamment, alors que pour les modèles discriminatifs, il faut effectuer un nouvel apprentissage chaque fois qu'une classe est ajoutée. D'un autre coté les modèles discriminatifs se concentrent sur les frontières entre les classes, là où la décision est la plus difficile à prendre, et ainsi, ils sont souvent plus efficaces et plus performants en terme de classification.

1.2.3 Différentes méthodes

Beaucoup d'algorithmes d'apprentissage ont été développés. La façon dont ils sont présentés dans la littérature dépend principalement du formalisme employé pour leur représentation. Dans cette section nous verrons quelques modèles qui seront fréquemment utilisés dans ce manuscrit.

Les **arbres de décision** sont des modèles prédictifs, qui permettent de mettre en correspondance une observation et une valeur cible. Deux types de valeurs sont possibles, discrètes ou continues, selon qu'il s'agisse d'un arbre de classification ou de régression. Nous nous intéresserons aux arbres de classification. Un arbre de décision possède des nœuds, reliés entre eux par des arrêtes ou branches. Les arbres sont parcourus en profondeur depuis la racine jusqu'à une feuille. Chaque nœud contient un classifieur unitaire qui détermine la branche par laquelle se continue le parcours. Les feuilles de l'arbre contiennent les labels de décision. Nous nous intéresserons plus particulièrement aux arbres de décision aléatoires (définis section 4.3.2) qui permettent de réduire la complexité de l'apprentissage. Nous les avons utilisés pour construire des cartes de saillance et remplacer les vocabulaires visuels dans le chapitre 4. Ils sont construits selon une approche discriminative.

les **machines à vecteurs de support**, ou *Support Vector Machine* (**SVM**) sont des fonctions discriminatives. Intuitivement, les SVM sont basées sur deux principes. Le premier est de maximiser la marge du classifieur, c'est à dire la distance entre la frontière de décision et les échantillons les plus proches. Le deuxième est l'utilisation d'une fonction noyau (*kernel trick*) pour travailler dans un espace de représentation où les données sont linéairement séparables. Les détails mathématiques peuvent être trouvés par exemple dans [3]. Les SVM présentent de bonnes propriétés de généralisation,

en effet, ils sont capables de construire des modèles sans sur-apprentissage, même dans le contexte de peu d'exemples d'apprentissage représentés par des vecteurs de grande dimension. Nous avons utilisé dans ce manuscrit des SVM à noyau linéaire pour classer les histogrammes d'occurrences de mots visuels. Un noyau du χ^2 pourrait peut être améliorer les résultats de classification, mais nous sommes convaincus que les conclusions tirées dans ce manuscrit sur les vocabulaire utilisés avec un noyau linéaire pourraient se transposer à un classifieur utilisant un espace de plus grande dimension.

Les **modèles graphiques** utilisent avantageusement des représentations des distributions de probabilité par des diagrammes. Cela permet de visualiser de façon simple et agréable les structures du modèle probabiliste. Cette modélisation permet également d'avoir un aperçu des propriétés du modèle à partir du graphe, notamment les propriétés d'indépendance conditionnelle (voir [3]). Un graphe est constitué de nœuds, qui représentent les variables aléatoires, et d'arcs qui connectent ces nœuds entre eux. Ce sont des relations probabilistes entre les variables. Le graphe capture la façon dont les facteurs joints dépendent seulement d'un sous ensemble de variables. (p360 Bishop). Plusieurs types de modèles graphiques sont généralement distingués :

- ▷ Les *réseaux bayésiens* ou *modèles graphiques directs* sont tels que le lien entre les nœuds a une direction particulière, indiquée par des flèches, et permet de modéliser une relation causale.
- ▷ Une deuxième classe très populaire est celle des champs de Markov (MRF) ou *modèle graphique indirect*. Les liens n'ont pas de direction, et modélisent des contraintes entre les variables aléatoires.

Les **modèles à variables latentes** sont des modèles graphiques directs. Ils présentent un ensemble de variables, dont certaines sont observées directement, les autres sont appelées variables latentes. Ces modèles supposent en général que les variables observées sont obtenues à partir de l'une des variables latentes et qu'elles sont indépendantes entre elles sachant les variables latentes.

Les **modèles à variables latentes d'aspect** sont des modèles à variables latentes d'un intérêt tout particulier. Les deux plus utilisés en vision sont les modèles pLSA et LDA. Introduits initialement pour la classification de textes, ces modèles sont basées sur une représentation en mots visuels. Le modèle pLSA [31] ou *Probabilistic Latent Semantic Analysis*, introduit par Hofmann suppose que les images sont décrites par des distributions sur des variables d'aspect, les *topics*, et que chaque topic possède une probabilité de générer chacun des mots. Les deux étant modélisés par des distributions multinomiales. Le modèle LDA ou *Latent Dirichlet Allocation*, introduit par Blei, Ng et Jordan [4], suppose quant à lui que ces probabilités multinomiales sont obtenues à l'aide d'un a priori de Dirichlet. Les modèles des chapitres 3 et 5 sont des extensions du modèle LDA.

Enfin, les **champs de Markov** nous intéresserons tout particulièrement. Ce sont des modèles graphiques indirects. Les modèles MRF supposent des contraintes spatiales entre les labels d'éléments voisins dans le graphe. Ils s'intéressent à la probabilité conjointe des observations et des labels. Ils servent en général à régulariser un champ de labels à l'échelle d'une image.

1.3 Bases d'images considérées

Dans les différents chapitres de ce manuscrit, les expériences sont conduites sur des bases de références, communément utilisées dans le domaine de la reconnaissance d'objets, et disponibles librement.

1.3.1 Base ETH-80

La base ETH-80 a été introduite pour la première fois dans [51]. Cette base contient 8 catégories d'objets, et chaque catégorie possède entre 10 et 14 instances. Nous avons considéré dans notre expérience un sous-ensemble de cette base, constitués des 4 classes *pomme*, *voiture*, *vache*, *tasse*. Chaque instance d'objet a été photographiée selon 41 vues différentes, prises selon des angles uniformément distribués. Notre ensemble présente au final 820 images. Deux exemples de chaque catégorie d'objets sont présentés figure 1.5.

Ces images sont utilisées pour la classification de catégories d'objets. Bien qu'elles n'aient pas été prises dans des conditions réelles (fond bleu), elles sont intéressantes pour deux raisons. Tout d'abord, l'absence de fond garantit que toute l'information provient des objets eux-mêmes. En effet, des tests peuvent être fait en supprimant toute influence du fond. Deuxièmement, cette base est intéressante pour la variété de ses points de vue. Construire un algorithme capable de regrouper une vue de face et une vue de profil du même objet dans la même catégorie, est une question ouverte et intéressante.

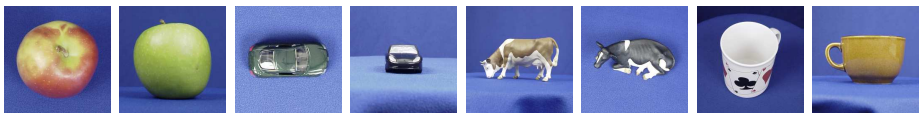


FIG. 1.5 – *Sur cette image sont montrés deux exemples pour chaque catégorie considérée sur la base ETH (pommes, voitures, vaches et tasses).*

1.3.2 Base TU-Graz02

La base TU-Graz02 contient des images d'une des 3 catégories d'objet suivantes : les *vélos*, les *voitures* et les *piétons* ainsi que des images ne contenant aucune de ces catégories (classe *fond*). L'ensemble de la base contient 404 images de vélos, 420 images de voitures, 311 images de personnes et 380 images de fond. Parmi celles-ci, 300 images de chaque catégorie possèdent un masque de segmentation fait à la main, à l'aide d'un outil « brosse » qui permet de dessiner le support spatial des différentes instances de la classe. Des exemples d'instances des différentes classes d'images, ainsi qu'un exemple de masque pour chaque catégorie, sont présentés figure 1.6

Ces masques de segmentation permettent d'apprendre des modèles d'objets précis ne contenant pas de fond mais aussi d'évaluer des résultats de segmentation. Ainsi, cette base d'images sera utilisée pour la classification d'images et la segmentation d'objets.

1.3.3 Base d'oiseaux

Cette base a été proposée par [49]. Elle contient 6 catégories d'oiseaux, et 100 images par catégorie. Les classes d'oiseaux présentées sont l'aigrette (egret),

FIG. 1.6 – *Exemples d'images de la base Graz. Sur la première ligne, les vélos, la deuxième ligne les personnes, et sur la troisième ligne, des exemples de fond et de voitures. La dernière colonne montre les annotations proposées comme vérité terrain pour une image de chaque catégorie d'objets.*

le canard mandarin (mandarin duck), le harfangs des neiges (snowy owl), le puffin, le toucan, et le canard carolin (wood duck).

Pour les différentes tâches considérées dans ce manuscrit, elle sera découpée en 300 images d'apprentissage et 300 images de test, selon le découpage proposé dans [49]. Des images de cette base sont proposées figure 1.7. Elle est utilisée en général pour la tâche de classification d'images, même si des résultats de segmentation sont aussi proposés ici. Pour ces derniers, des annotations supplémentaires ont été réalisées pour les images d'apprentissage : des boîtes englobantes donnant une idée approximative de la position du ou des animaux présents dans les images.

FIG. 1.7 – *Pour chaque classe de la base des oiseaux, deux exemples sont proposés.*

1.3.4 Base de papillons

Proposée par [48], cette base d'images contient 7 catégories de papillons, ce qui constitue en tout 619 images. Elles sont divisées en 182 images d'apprentissage et 437 images de test, selon le protocole de [48]. Les classes de papillons présentes sont le vulcain (Admiral, 111 images), le *Papilio polyxene* (Black Swallowtail, 42 images), le Machaon (83 images), le Monarche - divisé en deux classes : avec ailes fermées (74 images) avec ailes ouvertes (84 images) -, le paon de jour (Peacock, 134 images), le *Heliconius charithonia* (Zebra, 91 images). Des exemples de ces différentes catégories sont montrés figure 1.8. Comme pour la base d'oiseaux, cette base est utilisée pour

la classification d'images, même si la tâche de segmentation est également considérée, à partir des boîtes englobantes des images d'apprentissage.

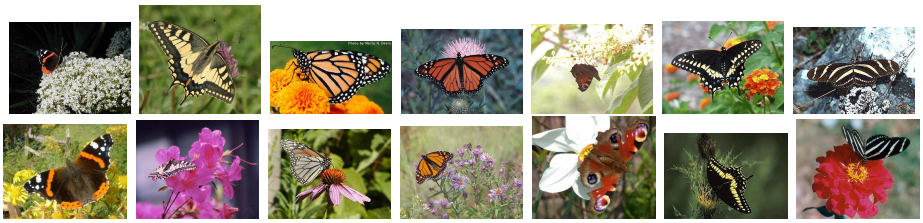


FIG. 1.8 – *Exemples de représentants des différentes classes de la base des papillons.*

1.3.5 Base Pascal VOC 2005

Cette base a été proposée à l'occasion de la compétition « Pascal Visual Object Classes (VOC) Challenge 2005 » [16], pour la classification d'images et la détection d'objets. Nous l'utiliserons dans le cadre de la classification. Elle contient 684 images d'apprentissage et 437 images de test. Les images appartiennent chacune à une des catégories suivantes : les motos, les vélos, les voitures et les personnes. Chaque image contient au moins un objet d'une de ces classes. Des exemples de chaque catégorie sont présentés figure 1.9. Les images sont considérées dans le cadre de tâches de classification binaire : pour chaque classe d'objet mentionnée plus haut, il faut prédire si un objet de la classe est présent dans l'image ou non.

FIG. 1.9 – *Deux exemples pour chaque catégorie de la base Pascal VOC 2005*

1.3.6 Base Pascal VOC 2006

En 2006, une nouvelle base a été proposée pour la compétition Pascal VOC [17]. Comme la précédente, elle est utilisée pour la classification et la détection. Cette base est constituée d'un ensemble d'images contenant une ou plusieurs des 10 catégories suivantes : les vélos, les bus, les chats, les voitures, les vaches, les chiens, les chevaux, les motos, les personnes et les moutons. Pour la plupart des images, un grand nombre d'objets apparaissent simultanément. L'ensemble complet contient 5304 images qui sont divisées en 1277 images d'apprentissage, 1341 images de validation et 2686 images de test. Les images d'apprentissage sont annotées avec des boîtes englobantes précisant la nature et la position des objets. Quelques images exemples, et les annotations associées sont présentées dans la figure 1.10.

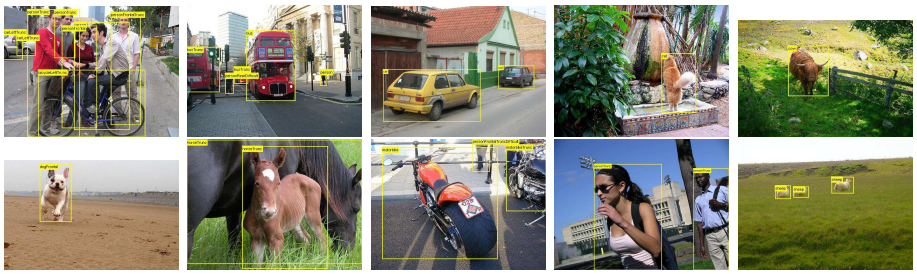


FIG. 1.10 – *Exemples d'images contenant les 10 classes de la base Pascal VOC 2006 ainsi que les annotations fournies.*

1.3.7 Base Pascal VOC 2007

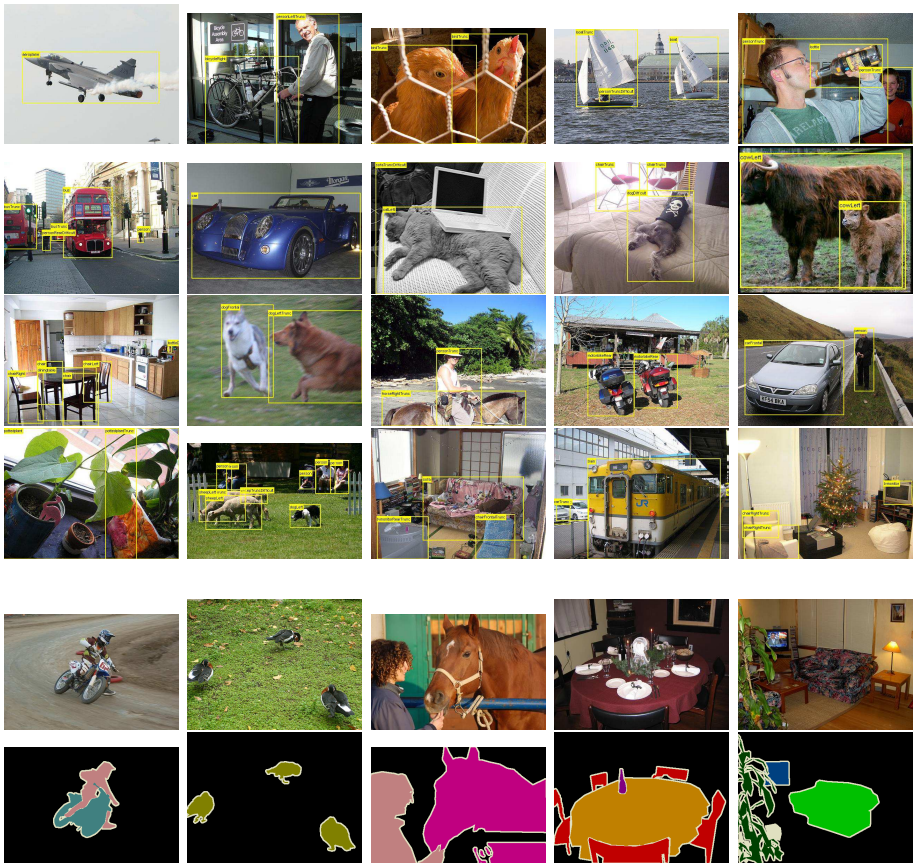


FIG. 1.11 – *Exemples d'images de la base Pascal VOC 2007, ainsi que les annotations fournies sous forme de boîtes englobantes. Deux dernières lignes, exemples d'annotations fournies pour la segmentation.*

Enfin, en 2007, la compétition Pascal VOC [15] a été étendue à 20 classes. En plus des 10 précédemment citées, les catégories suivantes sont ajoutées : les oiseaux, les bateaux, les bouteilles, les chaises, les avions, les plantes en pot, les sofas, les tables, les trains et les écrans (télévision/ordinateur). Les images sont divisées en 2501 images d'apprentissage, 2510 images de validation et 4952 images de test. Un aperçu de la base est possible dans la figure 1.11. Toutes les images d'apprentissage disposent d'annotations précisant le nombre, la classe et la position des objets à l'aide de boîtes

englobantes. Ces images et les annotations associées sont fournies pour les tâches de classification et de détection. Parmi les images d'apprentissage, 422 images sont précisément segmentées, donnant la position des objets au pixel près. Cet ensemble est utilisé pour l'apprentissage dans le cadre de la tâche de segmentation.

1.3.8 Base Microsoft MSRC

FIG. 1.12 – *Exemples d'images de la base Microsoft, et les annotations associées.*

La base Microsoft Research Cambridge 2 (ou MSRC) contient 591 images, manuellement segmentées en 21 catégories. Chaque image contient typiquement des objets provenant de 2 à 5 catégories, les régions d'image étant marquées de façon relativement précises, avec un outil de type « brosse ». Des exemples d'images et d'annotations sont montrées figure 1.12. Les catégories considérées sont les bâtiments, l'herbe, les arbres, les vaches, les moutons, le ciel, les avions, l'eau, les visages, les vélos, les fleurs, les panneaux de signalisation, les oiseaux, les livres, les chaises, les voitures, les chats, les chiens, les corps et les bateaux. Certaines zones de l'image ne sont pas annotées, cela signifie qu'elles n'appartiennent à aucune des classes citées précédemment. Nous utiliserons cette base dans le cadre de la segmentation de catégories d'objet.

1.4 Organisation du reste du document

Chacun des chapitres suivants est dédié à une contribution, et inclut directement un positionnement de l'approche proposée vis à vis de l'état de l'art, les résultats associés et une conclusion, qui lui sont propres.

Dans le chapitre 2, nous présentons une méthode de création de vocabulaires visuels, utilisée dans le cadre de l'approche par sac-de-mots, et particulièrement adaptée aux images représentées à l'aide d'un très grand nombre de descriptions locales. Ces travaux ont été présentés lors de la conférence RFIA en 2006, et font l'objet de la publication associée [42].

Le chapitre 3 s'intéresse également au problème de la création de vocabulaires visuels. Dans ce chapitre, nous étudierons comment créer un vocabulaire visuel discriminant, adapté à la tâche de reconnaissance. Des résultats ont fait l'objet d'une présentation orale à BMVC en 2006, et d'une publication associée [42]. Plus récemment, ces résultats ont été étendus dans le journal *Image and Vision Computing* [45].

Dans le chapitre 4, la classification d'images est réalisée par la combinaison d'arbres de décision aléatoires et de classifieurs plus classiques en reconnaissance d'objets. Nous montrons comment la classification locale par les arbres de décision peut être utilisée pour construire des cartes de saillance, guidant le processus d'échantillonnage des descriptions locales vers les zones contenant potentiellement les objets d'intérêt. Ces résultats ont donné lieu à une publication pour un workshop en coordination avec ECCV en 2006 [61].

Le chapitre 5 présente une extension du modèle LDA, qui contrairement au modèle classique, possède plusieurs documents, qui se chevauchent, par image. Ce modèle est utilisé pour la segmentation de catégories d'objets. Nos résultats ont été présentés à VISAPP en 2007, [43] est la publication correspondante. Une extension de ces travaux est également disponible dans le journal *Image and Vision Computing* [45] précédemment cité.

Le chapitre 6 propose un algorithme qui combine un modèle génératif, qui permet la reconnaissance et une estimation de la position des objets dans les images, avec un champ de Markov qui produit un découpage précis de ces objets. Une version préliminaire de ces travaux a été présentée à la conférence RFIA en 2008 [46], mais l'essentiel du travail a été publié à CVPR la même année [44]. Puis nous avons également proposé une alternative à ce travail, utilisant les arbres de décision présentée dans le chapitre 4. La comparaison du modèle original avec son alternative est proposée dans des travaux soumis pour la revue I3 [41] et au journal IJCV [47].

Enfin, le chapitre 7 présente une application possible du modèle présenté dans le chapitre 6 à un cadre robotique, où le modèle est utilisé pour proposer des hypothèses sur la position et l'échelle d'objets. Ces hypothèses sont les entrées d'un algorithme de recherche visuelle active. Des améliorations du modèle, spécifiques à l'application considérée, sont proposées. Des expériences menées sur la plateforme robotique humanoïde HRP-2 ont fait l'objet d'une publication à la conférence Humanoids [84].

Pour finir, durant la thèse, j'ai également pris part à des travaux qui ont fait l'objet des publications [10] et [11] mais qui ne seront pas détaillés dans ce manuscrit.

Une méthode efficace pour la quantification vectorielle de larges volumes de descripteurs visuels

2

Sommaire

Le *vocabulaire visuel* est une notion imaginée, par analogie avec les mots textuels, dans le but de pouvoir construire des modèles statistiques des images (modèle par sac-de-mots par exemple). Les *mots visuels* sont obtenus par quantification vectorielle de descripteurs locaux extraits des images. Différents types de quantification, dont vont dépendre les performances des algorithmes, sont possibles. Dans ce chapitre, nous nous plaçons dans le cas où les images sont décrites par des descripteurs locaux échantillonnés de façon dense. La construction du vocabulaire visuel doit donc tenir compte de ce grand nombre de données et de leurs caractéristiques. Nous proposons ici une nouvelle méthode de quantification adaptée à ce contexte et capable de construire rapidement des vocabulaires de taille importante.

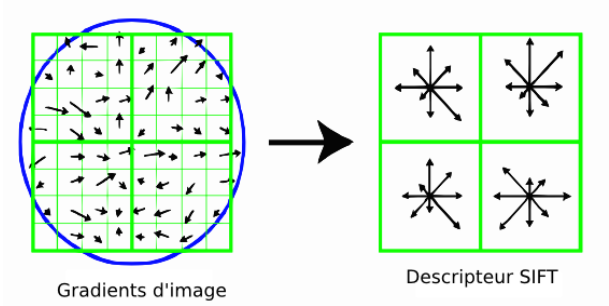


FIG. 2.1 – *Illustration du principe de calcul du descripteur SIFT, pour une grille* 2×2 *(figure de [54])*

2.1 Introduction

Ce chapitre s'intéresse à la classification d'images, c'est-à-dire à l'assignation d'une (ou plusieurs) étiquettes aux images, en fonction de la présence ou de l'absence d'objets particuliers. Pour les raisons évoquées dans l'introduction, nous nous proposons d'utiliser ici un modèle par sac-de-mots (présentée en détail dans la section 1.1.2) des images. Cela soulève un certain nombre de questions telles que la sélection des informations locales, leur description ainsi que la construction du vocabulaire visuel.

2.1.1 Représentation locale des images

Nous avons vu dans la section 1.1.2 que les pixels ne pouvaient être utilisés individuellement, et qu'il était préférable de les regrouper en indices visuels locaux (appelés ici patchs). Différents descripteurs visuels peuvent être utilisés pour représenter les patchs ; dans ce chapitre nous utiliserons le descripteur SIFT (Scale Invariant Feature Transform [54]). Ce descripteur a été utilisé avec succès dans de nombreuses applications de reconnaissance d'objets et permet d'obtenir les meilleures performances dans un contexte de mise en correspondance de points d'intérêt [60].

Centré en un point particulier de l'image, le descripteur SIFT est constitué d'un histogramme grossier des orientations des gradients contenus dans un voisinage de ce point. De manière plus précise, les gradients sont groupés en 8 orientations différentes, dans chaque case d'une grille 4x4 centrée sur ce point particulier, donnant ainsi des vecteurs de dimension 128[1]. La taille de la fenêtre utilisée pour définir le voisinage est directement liée à l'échelle à laquelle l'image est analysée. Une illustration du descripteur est donnée figure 2.1.

2.1.2 Représentation des images et stratégie de catégorisation

Nous venons de voir comment représenter localement l'image au moyen de descripteurs locaux. Nous allons désormais décrire comment combiner ces représentations locales pour former une représentation de l'ensemble de l'images.

[1] Ces valeurs numériques sont celles utilisées dans ce mémoire. Elles peuvent bien entendu être modifiées pour s'adapter à différents contextes. Nous avons repris les valeurs préconisées par D. Lowe, valeurs déterminées de manière à donner des performances optimales de mise en correspondance de points d'intérêt [54].

Pour représenter globalement une image ou un objet, nous avons adopté une approche de type *sac-de-mots*, introduite dans la section 1.1.2. Nous avons signalé que, contrairement au cas de l'analyse de documents, le vocabulaire visuel n'est pas une donnée intrinsèque aux images. Il n'existe pas de vocabulaire unique pour décrire les images ; le vocabulaire doit être construit pour répondre à des propriétés particulières.

C'est dans ce point que réside le cœur de ce chapitre et des deux suivants : comment produire le *meilleur* vocabulaire, c'est-à-dire celui qui permettra d'obtenir les meilleurs performances de catégorisation d'images.

Une fois le vocabulaire construit, nous représentons l'image en prenant tous les descripteurs se trouvant sur les nœuds d'une grille régulière (à la fois en position et échelle) et en remplaçant le descripteur par le mot du vocabulaire qui le représente. Les techniques utilisées pour le texte deviennent alors applicables.

Nous utiliserons en particulier ici la représentation d'image par histogrammes d'occurrences (sac-de-mots), qui consiste simplement à compter le nombre de fois que chaque mot visuel apparaît dans l'image, suivi d'une classification au moyen de classifieurs binaires SVM linéaires [89], choisis en raison de leurs bonnes performances même dans le cas de grande dimension. Nous entraînons un classifieur, pour chaque catégorie d'objet dont la présence doit être détectée dans l'image.

2.1.3 Construction de vocabulaires visuels

Comme nous venons de le dire, la construction du vocabulaire est au cœur de notre étude. Aucun vocabulaire visuel n'existant de manière explicite, il s'agit de s'interroger sur les propriétés que doivent posséder les vocabulaires visuels et sur les méthodes à utiliser pour les construire.

La construction du vocabulaire suppose une quantification de l'espace de représentation des descripteurs locaux. Il s'agit en effet de construire une fonction de l'espace de représentation (R^{128} dans notre cas) vers un espace discret de labels.

L'espace de représentation des patchs n'est pas peuplé de manière dense et uniforme. Certains motifs visuels (théoriquement possibles) peuvent ne jamais apparaître dans les images tandis que d'autres peuvent être très fréquents. La première conséquence de cette remarque est que le vocabulaire doit être adapté aux images rencontrées, c'est-à-dire il doit être le reflet des descriptions locales présentes dans les images.

La méthode la plus utilisée pour construire les vocabulaires visuels consiste à partir des descripteurs rencontrés dans les images d'apprentissage (statistiquement représentatives des données) et à les regrouper en un nombre fini de *clusters* au moyen d'un algorithme de *clustering* (quantification vectorielle). Le nombre de clusters représente la taille du vocabulaire.

Les vocabulaires visuels reposent sur trois composants différents :

1. les descripteurs locaux d'images,
2. le nombre et la position des primitives locales décrites par ces descripteurs,
3. l'algorithme utilisé pour la quantification.

Dans ce chapitre, ainsi que dans les deux suivants, nous nous intéresserons au troisième point.

Le reste du chapitre est organisé comme suit. Tout d'abord nous présentons quelques travaux sur la classification d'images et la création de vocabulaires visuels. Puis nous présentons notre contribution. La section dédiée aux expériences présente une étude paramétrique de la méthode ainsi que des comparaisons avec l'état de l'art. Finalement nous concluons le chapitre.

2.2 État de l'art

Le terme de *texton*, proposé initialement par Julesz [34] il y a 20 ans de cela, représente l'élément de base des textures des images. Dans [52], Leung et Malik proposent une méthode de construction de texton : leur idée est de représenter localement l'image au moyen de convolutions avec des banques de filtres gaussiens orientés, puis de quantifier ces représentations locales avec un algorithme des k-moyennes (*k-means*). Chaque point de l'image est ainsi traité. Il s'agit d'une représentation dense des images. Bien que le terme de vocabulaire visuel n'ait pas été utilisé dans ces travaux, l'idée est la même et les deux termes textons et mots visuels peuvent être considérés comme similaires.

Plutôt que de traiter l'image de manière dense, il est possible de sélectionner un sous ensemble des patchs qui sont plus informatifs que les autres, à l'aide d'un critère de saillance, réduisant ainsi la quantité d'information à traiter (méthodes éparses). Une des premières approches éparses utilisant un vocabulaire est celle de Weber *et al* [96]. Le vocabulaire est appris à partir d'un ensemble d'images dont des points d'intérêts sont extraits avec un détecteur de Förstner, à une seule échelle. La quantification de ce petit nombre de patchs est réalisée par les k-moyennes.

Csurka *et al* [13] ont utilisé une approche par sac-de-mots, où des descripteurs SIFT [54] sont utilisés pour représenter localement les images. Les images ne sont considérées qu'en un petit nombre de points, choisis par l'algorithme Harris-affine [59]. La quantification produisant le vocabulaire est obtenue par l'algorithme des k-moyennes, appliqué aux descripteurs locaux. Cette combinaison d'un détecteur de points d'intérêt et d'un clustering par l'algorithme des k-moyennes est devenue très populaire pour les représentations par sac de mots [13, 82, 96, 56].

Bien que tous les travaux précédents utilisent l'algorithme des k-moyennes pour quantifier les descripteurs, d'autres méthodes ont été utilisées. Leibe *et al* [51] ont proposé une méthode de détection d'objets qui utilise un vocabulaire visuel appris en détectant des points de Harris dans des images d'apprentissage, puis qui regroupe ces points au moyen d'un algorithme de clustering hiérarchique. Les descripteurs locaux sont des vecteurs de niveaux de gris. Le détecteur proposé par Agarwal *et al* [1] repose sur une approche similaire, mais en incorporant des relations géométriques entre les mots du vocabulaire. Le vocabulaire est là encore obtenu avec un algorithme de clustering hiérarchique.

2.2.1 Clustering de descripteurs denses

Bien que les points d'intérêt soient des outils très puissants dans le contexte de la mise en correspondance entre deux vues différentes du même objet,

il ne sont pas suffisamment répétables dans le cas des catégories d'objets, comme montré par [35].

De même, des travaux récents [67] montrent que pour un petit nombre d'échantillons, les détecteurs de points d'intérêt sont pertinents, mais que la stratégie la plus efficace reste l'échantillonnage intensif et aléatoire des images. Même si individuellement ils sont plus discriminants, leur petit nombre constitue une limite majeure. En effet, ces détecteurs suppriment des régions potentiellement utiles, comme les régions sans textures.

Pour ces raisons, il nous semble préférable d'échantillonner les images aussi densément que possible, afin de supprimer tout risque de manquer des informations utiles à la classification. D'autres auteurs utilisent ce type d'échantillonnage, nous avons évoqué tout à l'heure les textons, mentionnons également les travaux de Winn *et al* [98], entre autres, qui utilisent tous les pixels de l'image, sans aucune sélection. L'inconvénient de cet échantillonnage dense est qu'il augmente fortement le nombre de descripteurs extraits des images. Alors que les points d'intérêt ne sélectionnent que quelques centaines de points par image, notre stratégie d'échantillonnage génère plus de 10 000 descripteurs. Ces descripteurs doivent être quantifiés pour produire le vocabulaire.

Lorsqu'on traite un grand nombre de données, des méthodes de quantification spécifiques sont nécessaires. Deux types d'approches sont possibles. La première considère toujours des distances dans l'espace des descripteurs, simulant ainsi une estimation de la densité. C'est l'objet de ce chapitre et des méthodes présentées dans l'état de l'art, dans le paragraphe suivant. Une autre approche consiste à s'affranchir de la représentation des densités, et ne considérer que les frontières entre les régions dont le support représente un mot. C'est la stratégie qui est étudiée dans le chapitre 4, et ce chapitre contient un état de la littérature associée.

Une combinaison arborescente de quantifications obtenues par l'algorithme des k-moyennes et de clustering agglomératif [50] a été proposée pour la mise en correspondance d'objets et la reconnaissance. Une méthode basée sur une hiérarchie des k-moyennes a été appliquée par [65]. Des vocabulaires plus larges sont ainsi créés et utilisés pour une mise en correspondance spatiale rapide dans le contexte de recherche des objets. Il est montré que cette méthode surpasse la méthode d'approximation des k-moyennes [71] qui fait aussi bien que l'algorithme des k-moyennes standard, toujours dans un contexte de mise en correspondance spatiale.

Ces méthodes permettent de construire des vocabulaires larges et efficaces, mais ils sont tous basés sur l'algorithme des k-moyennes. Par conséquent, ils présentent la même faiblesse majeure que celui-ci : ils ne sont pas adaptés aux données déséquilibrées, comme souligné par [24, 92, 36]. Certains clusters sont particulièrement denses comparés aux autres, et l'algorithme des k-moyennes leur donne une influence supérieure. Certains clusters de faible densité ne sont pas trouvés et sont agglomérés à des clusters plus larges, même s'ils sont loin l'un de l'autre. Cependant il a été montré que la densité des descripteurs locaux est loin d'être uniforme [12, 36] et les données à quantifier présentent fortement ce déséquilibre. Cela est particulièrement vrai lorsque les descripteurs sont échantillonnés densément.

De plus il a été montré pour les images que les clusters les plus larges ne sont pas les plus informatifs [92, 36]. Les méthodes produisant le voca-

bulaire devraient prendre en compte ces particularités. Ce type de problème est également connu en analyse de données sous le nom d'*analyse des cas rares* [97].

La question des données très déséquilibrées obtenues lorsque les images sont échantillonnées densément, a été résolue par [35] en utilisant un algorithme de « translation de la moyenne » (*Mean Shift*), mais cette méthode a une très forte complexité en temps, donc elle ne peut être utilisée lorsque de grandes quantités d'images d'apprentissage sont considérées.

Les méthodes agglomératives peuvent traiter les données déséquilibrées, mais elles sont mal adaptées aux larges nombres d'échantillons. En effet, une recherche rapide des plus proches voisins est infaisable pour autant de vecteurs. La stratégie consistant à sous-échantillonner les vecteurs pour réduire leur nombre, et ensuite utiliser les méthodes sur le sous-échantillon n'est pas satisfaisante à cause de la raison précédemment citée : les clusters sont fortement déséquilibrés et l'échantillonnage sélectionnerait principalement les clusters les plus larges.

La contribution principale de ce chapitre est un algorithme de création de vocabulaires visuels efficaces capables de traiter un grand nombre de données et les caractéristiques des données associées.

2.2.2 Approche choisie

La méthode de clustering devrait (a) permettre de trouver les clusters de taille plus modeste et (b) être capable de faire face au très grand nombre de vecteurs traités. C'est pourquoi nous avons retenu les principes suivants :

- ▷ L'algorithme de clustering est appliqué sur un sous-échantillonnage des données, tous les vecteurs ne sont pas considérés simultanément
- ▷ L'échantillonnage est biaisé : un échantillonnage uniforme donnerait trop de poids aux clusters denses. Nous proposons de compenser en introduisant un échantillonnage biaisé des données
- ▷ L'algorithme de clustering fonctionne de façon séquentielle : les clusters sont choisis un par un, et chaque nouvelle itération prend en compte les clusters déjà trouvés.

La solution que nous proposons combine l'algorithme *online median* [58] avec un échantillonnage biaisé des données. Les régions contenant les centres des clusters sont considérées comme denses et sur-représentées. L'échantillonnage évite ces régions. Nous utilisons le fait que les centres sont choisis itérativement pour alterner entre une phase d'échantillonnage biaisé et une phase de clustering.

2.3 Méthode proposée

2.3.1 Online Median

L'algorithme *online median* [58], proposé par R. Mettu et C. Plaxton est une solution *online*[2] du problème des *k-median*. Au lieu d'optimiser globalement le placement de k centres (k fixé), ceux-ci sont placés un par un, selon le principe de *Facility Location*.

[2] Un algorithme de clustering *online* place les centres un par un mais peut revenir autant de fois que nécessaire sur les données. Nous distinguons ce terme de *streaming* qui signifie pour nous que les données ne peuvent être vues qu'une fois, dans un ordre déterminé.

Facility Location

Une chaîne de magasin veut s'implanter dans une ville, et cherche la façon la plus judicieuse de placer un premier magasin, en fonction de la situation géographique de ses clients. Les affaires marchent, la chaîne est amenée à placer un deuxième, puis un troisième, etc. magasins. Il est évident que les anciens magasins ne sont pas déplacés, mais les nouveaux sont placés de façon optimale afin de couvrir au mieux la clientèle. C'est ainsi qu'on pourrait définir le problème de « facility location » (voir illustration figure 2.2).

Ce problème revient à placer des centres (*facilities*) un par un de manière à minimiser la somme des distances entre chaque point et son centre le plus proche. Un centre placé ne sera plus jamais déplacé.

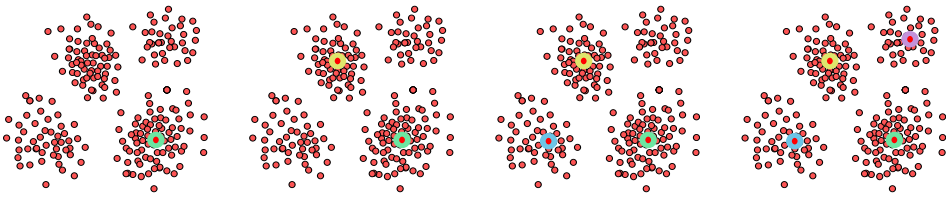


FIG. 2.2 – *Le problème de « facility location », illustré pour 4 placements.*

Algorithme

L'algorithme online median place successivement, et de façon définitive, les centres des clusters. Le processus s'arrête lorsqu'un critère d'arrêt est vérifié, ou lorsque toutes les données ont été choisies comme centre.

L'algorithme dispose de paramètres fixés $\alpha, \beta, \gamma, \delta$. Définissons tout d'abord la valeur d'une boule A de centre x et de rayon r.

$$val(A) = \sum_{y \in A} (r - d(x, y))$$

Le fils d'une boule A de centre x et de rayon r est un point y qui vérifie $d(x, y) < r\beta$. Rappelons que $d(y, X)$ représente $min_{x \in X} d(y, x)$.

Chaque étape d'ajout d'un centre se fait ainsi :

- ▷ (1) Calcul de la valeur de toutes les boules centrées en un point x des données D et de rayon $\frac{d(x,Z)}{\gamma}$ où Z est l'ensemble des centres déjà placés. Si $Z = \emptyset$ (cas du premier centre), le rayon choisi est $\max_{x,y \in D} d(x, y)$.
- ▷ (2) Sélection de la boule A_0 de valeur maximale.
- ▷ (3) Tant que A_i contient plusieurs fils
  - ▷ Considérons les boules centrées en y vérifiant $d(x_i, y) \leq \beta r_i$, de rayon $r_{i+1} = \frac{r_i}{\alpha}$. Soient A_{i+1} la boule de valeur maximale, et x_{i+1} le centre correspondant. Son rayon est le r_{i+1} précédent.
- ▷ (4) Lorsque la boule A_i n'a qu'un seul fils, on le choisit comme nouveau centre du cluster.

Cela revient à mettre à jour une boule dont le rayon diminue à chaque itération (par $r_{i+1} = \frac{r_i}{\alpha}$) et qui se déplace à chaque étape vers la région de « plus forte densité », estimée à travers la valeur.

Qualité de l'approximation

Le problème des *k-median* est combinatoire, et seule une approximation du résultat est calculable en un temps polynomial. La méthode proposée ici garantit une approximation de coût bornée par une constante par rapport au coût optimal.
Le coût d'une configuration X (= ensemble de centres choisis) est

$$cost(X) = \sum_{y} d(y, X)$$

On peut montrer que pour toute configuration choisie, on a

$$cost(Z_{|X|}) \leq 2\lambda(\gamma + 1)cost(X)$$

où $Z_{|X|}$ est la configuration choisie par l'algorithme pour un même nombre de centres.

Cet algorithme sépare l'espace en clusters de forme sphérique, avec une garantie sur la valeur de la fonction de coût. La complexité de la méthode est élevée ($n^2 \log(n)$). Elle ne convient donc pas directement aux données à traiter, en raison de leur grande quantité et de leur déséquilibre.

2.3.2 Algorithme proposé

L'algorithme online median présenté dans la section précédente est utilisé sur un échantillonnage biaisé. Cet échantillonnage n'utilise pas de calcul d'estimation des densités. Par contre, les régions contenant les centres sont considérées comme denses et sur-représentées. L'échantillonnage est fait de façon à éviter ces zones.

Nous tirons parti du fait que les centres soient placés les uns après les autres pour alterner phases d'échantillonnage biaisé, et placements de nouveaux centres.

Un clustering à deux phases

Comme nous venons de le signaler, nous utilisons le fait que les centres soient placés itérativement pour alterner une phase de réchantillonnage (choix de nouveaux descripteurs) et une phase de placement des centres. A chaque étape de placement des centres, ceux-ci sont ajoutés à la liste des centres déjà choisis. A chaque étape de réchantillonnage, tous les centres trouvés sont utilisés pour guider l'échantillonnage. Ainsi les descripteurs sont changés entre chaque nouveau groupe de clusters ajoutés. La figure 2.3 illustre ce procédé.

Échantillonnage biaisé

L'idée retenue est de favoriser la découverte de nouvelles régions, au détriment des clusters très denses. Il faut donc échantillonner loin des centres déjà placés. Pour cela un rayon d'influence est défini. Tous les vecteurs contenus dans une boule centrée sur un centre de cluster et de rayon ce rayon d'influence sont considérés comme affectés au cluster correspondant et ne seront pas échantillonnés.

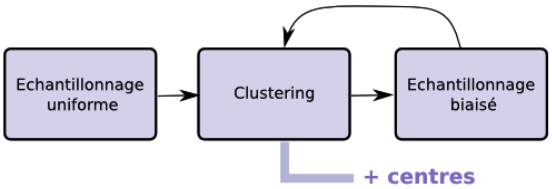


FIG. 2.3 – *Algorithme à deux phases : échantillonnage puis clustering. A chaque étape de clustering, de nouveaux centres sont ajoutés*

Lors d'une étape d'échantillonnage, on choisit uniquement des points hors des boules d'influence de tous les centres déjà placés (voir figure 2.4). La boule d'influence est un paramètre de l'algorithme dont nous étudierons l'impact dans la section 2.4. Plutôt que de considérer des régions d'influence sphériques, un modèle probabiliste pourrait être utilisé (modèle gaussien par exemple).

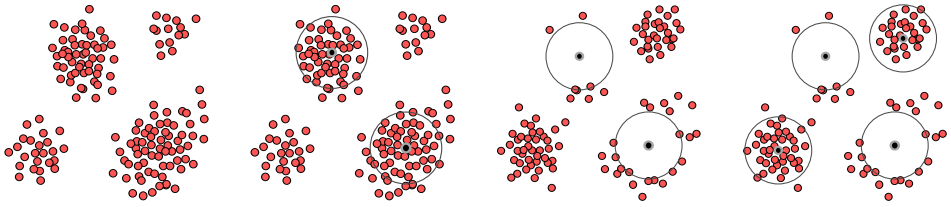


FIG. 2.4 – *Les points appartenant au rayon d'influence des centres placés pendant la première étape ne sont pas utilisés pour l'échantillonnage suivant. D'autres zones sont découvertes.*

Le rôle de cette boule d'influence est de limiter l'effet des zones de très forte densité afin qu'un centre ne soit pas placé au même endroit qu'un précédent. En effet, on évite ainsi des redondances dans le vocabulaire, et on trouve des classes moins peuplées qui peuvent s'avérer discriminantes. Nous revenons sur le sens à donner à cette région d'influence dans la section 2.3.3.

Les propriétés de convergence de l'algorithme initial ne sont plus préservées.

2.3.3 Construction des histogrammes

Le vocabulaire est constitué de l'ensemble des centres produits par l'algorithme que nous venons de décrire, à partir d'un ensemble d'apprentissage.

Comme nous l'avons expliqué section 2.1.2, les descripteurs locaux d'une image doivent être traduits en mots de vocabulaire.

Nous avons imaginé deux règles de traduction correspondant à deux visions différentes du problème :

- ▷ affectation du descripteur au cluster dont le centre est le plus proche du descripteur, quelle que soit la distance qui les sépare,
- ▷ affectation du descripteur à l'ensemble des clusters pour lesquels le point tombe dans la zone d'influence.

Dans le premier cas, on cherche à quantifier tout l'espace des descripteurs à l'aide des éléments du vocabulaire. L'espace est ainsi entièrement partitionné. Dans le deuxième cas, on utilise le même système que pour la

phase de clustering. Les points sont considérés comme appartenant à un cluster s'ils sont dans la boule d'influence centrée sur son représentant. Le reste est considéré comme du bruit. Ces deux possibilités sont comparées dans les expériences.

2.3.4 Utilisation des histogrammes pour la classification

Nous avons vu dans les sections précédentes qu'un vocabulaire visuel est construit en utilisant des descripteurs extraits des images d'apprentissage, et que ce vocabulaire est ensuite utilisé pour transformer chaque image en histogramme d'occurrence de mots visuels. Les histogrammes normalisés de toutes les images d'apprentissage sont utilisés pour entraîner un classifieur SVM [89] linéaire. Ce dernier peut ensuite prédire une classe pour tout histogramme normalisé construit sur une nouvelle image, et également donner un score de confiance associé.

2.4 Expériences

La partie expérimentale est divisée en 4 sous-parties. En premier lieu, nous passons en revue les paramètres de l'algorithme et leur influence sur les résultats finaux de classification. Ensuite, nous comparons le vocabulaire construit par notre algorithme avec d'autres méthodes de clustering. Nous comparons des techniques de réduction de la dimension afin de diminuer le nombre de mots visuels. Enfin nos résultats sont comparés à des méthodes à la pointe de la recherche sur les bases d'images présentées dans la section 2.4.1.

Commençons par nommer les bases d'images et décrire l'évaluation des résultats.

2.4.1 Bases d'images et évaluation des résultats

Les expériences sont principalement menées sur la base Pascal VOC 2005, présentée section 1.3.5. Nous considérons également la base TU-Graz02, introduite dans la section 1.3.2. Rappelons que toute deux contiennent des images difficiles, appartenant à différentes classes d'objets. Elles sont utilisées pour la classification dans un contexte binaire.

La classification d'images est évaluée grâce à des courbes ROC (*Receiver Operating Characteristic*). La courbe ROC représente le taux de vrais positifs TP en fonction du taux de faux positifs FP. Le taux de vrais positifs à l'EER (Equal Error Rate) est également calculé. Il correspond au point de fonctionnement de la courbe où $TP = 1 - FP$.

2.4.2 Étude paramétrique

Les résultats produits par notre méthode dépendent des paramètres suivants :

1. Valeur du rayon d'influence dans le clustering : ce rayon contrôle le nombre d'éléments affectés à chaque centre.
2. Utilisation d'un rayon d'influence pour la construction des histogrammes : l'affectation à un élément du vocabulaire ne se fait qu'à l'intérieur de ce rayon.

3. Nombre de mots du vocabulaire : le vocabulaire utilisé peut être plus ou moins grand.
4. Nombre de centres ajoutés à chaque itération de l'algorithme : sur un même échantillonnage des données, un certain nombre d'éléments du vocabulaire est produit, lors de chaque étape de clustering.
5. Nombre d'échantillons sélectionnés à chaque itération : l'algorithme choisit un échantillonnage plus ou moins important des données, lors de chaque étape de réechantillonnage.
6. Normalisation de l'histogramme : le classifieur utilise une version normalisée de la représentation des images, différentes normalisations sont proposées.
7. Données utilisées pour la création du vocabulaire : utilisation de toute l'image ou uniquement de la partie des images contenant l'objet.

L'étude jointe des influences des paramètres est difficilement réalisable, compte tenu du nombres de combinaisons possibles ; nous avons donc considérer un seul paramètre à la fois. Toutes les expériences qui suivent sont réalisées sur la base Pascal VOC 2005.

Normalisation

Les histogrammes obtenus doivent être normalisés de manière à les rendre invariants à différents paramètres, comme en particulier la taille des images.

Différents types de normalisation sont possibles. Nous allons comparer deux types de normalisation. La normalisation standard consiste à normaliser les vecteurs représentant les histogrammes. C'est ce que nous appellerons la normalisation par image. La deuxième normalisation binarise les vecteurs en seuillant à 1 toutes les valeurs différentes de 0.

Ces deux types de normalisation donnent des résultats similaires sur le nombre standard de mots que nous avons utilisé. Pour de plus petits vocabulaires, la normalisation par image fonctionne mieux alors que pour des vocabulaires visuels plus grands la normalisation binaire tends à surpasser la normalisation standard. Sauf mention contraire, nous utilisons la normalisation standard. Des résultats de comparaison entre ces deux normalisations sont présentées figure 2.5.

Influence du rayon

Nous avons montré comment les performances varient en fonction de la valeur du rayon d'influence (défini section 2.3.2). Pour cette sous-section, le rayon est utilisé à la fois pour le clustering, mais aussi pour le calcul de l'histogramme. Des résultats pour 4000 mots de vocabulaire sont présentés dans figure 2.5. Avec un rayon trop faible, même 4000 centres ne suffisent pas pour avoir une bonne représentation des images. Il faut donc un rayon d'au moins 0.6 avec une représentation SIFT, valeur utilisée par la suite, pour tester les autres paramètres. Avec un rayon trop élevé, on impose une trop grande distance entre les centres, et les points sont très vite tous affectés. Les derniers centres trouvés ne correspondent qu'à du bruit.

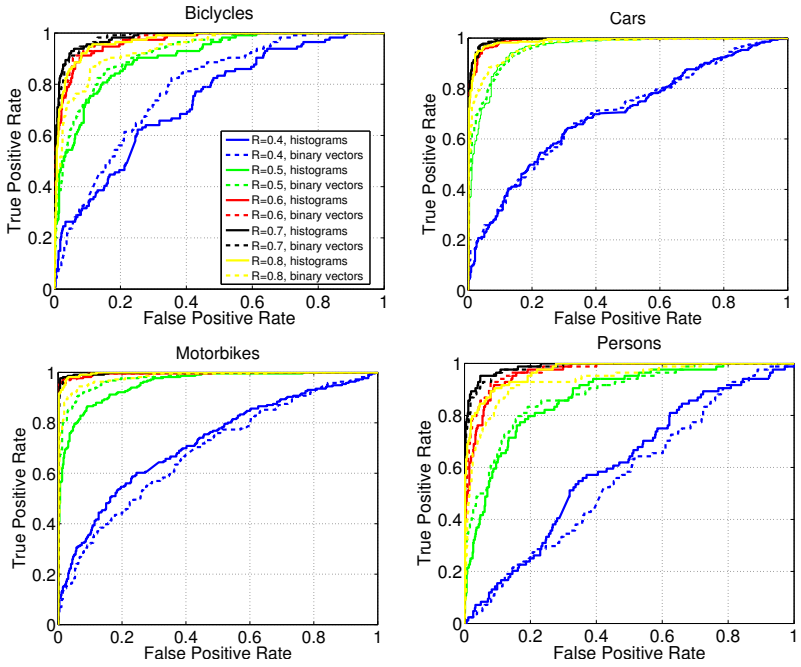


FIG. 2.5 – *Effet du rayon d'influence sur les performances de l'algorithme.* `histograms` *dénote une normalisation par image et* `binary vectors` *dénote une normalisation binaire*

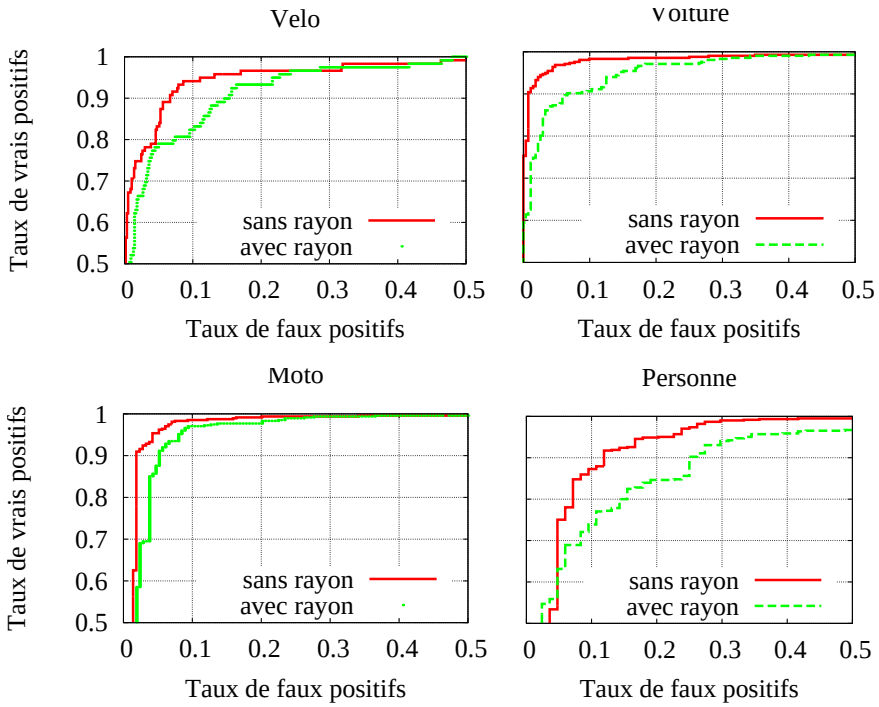


FIG. 2.6 – *Comparaison pour le même vocabulaire visuel, des résultats obtenus avec une affectation au plus proche et l'utilisation du rayon d'influence utilisé dans le clustering, lors de la construction des histogrammes.*

Utilisation du rayon d'influence pour le calcul des histogrammes

Le rayon que nous avons utilisé pour biaiser l'échantillonnage peut aussi être utilisé pour construire les histogrammes. Dans ce cas, nous supprimons les

descripteurs qui sont trop loin d'un centre déjà trouvé en utilisant ce rayon. Ceux-ci ne seront pas représentés par un mot visuel, et n'interviennent pas dans la représentation. Cette stratégie dégrade les résultats. Pour réaliser l'histogramme, l'utilisation d'une affectation totale, c'est-à-dire la création d'une partition des descripteurs, par affectation de tous les points à l'élément du vocabulaire le plus proche, s'avère bien meilleure. Garder le rayon d'influence du clustering (classification avec rejet) donne des résultats moins bons, comme le montre la figure 2.6.

Ce constat implique des conséquences pratiques. Lorsque plus tard nous souhaiterons utiliser des algorithmes de réduction de la dimension, la suppression de certains mots du vocabulaire imposera de recalculer les histogrammes.

Nombre de centres ajoutés par itération

Dans notre algorithme à deux phases, un certain nombre de centres est choisi lors d'une étape de clustering (voir section 2.3.2). L'idéal serait de ne prendre qu'un seul centre à chaque itération, afin de toujours avoir un échantillonnage optimal, mais ceci est bien trop coûteux en calculs, puisqu'il faudrait autant de phases d'échantillonnage, et donc de parcours des données, que d'éléments dans le vocabulaire. Les centres sont donc choisis par groupe dont la taille est un paramètre de l'algorithme.

Nous comparons l'évolution des résultats pour des groupes de 10 à 50 vecteurs.

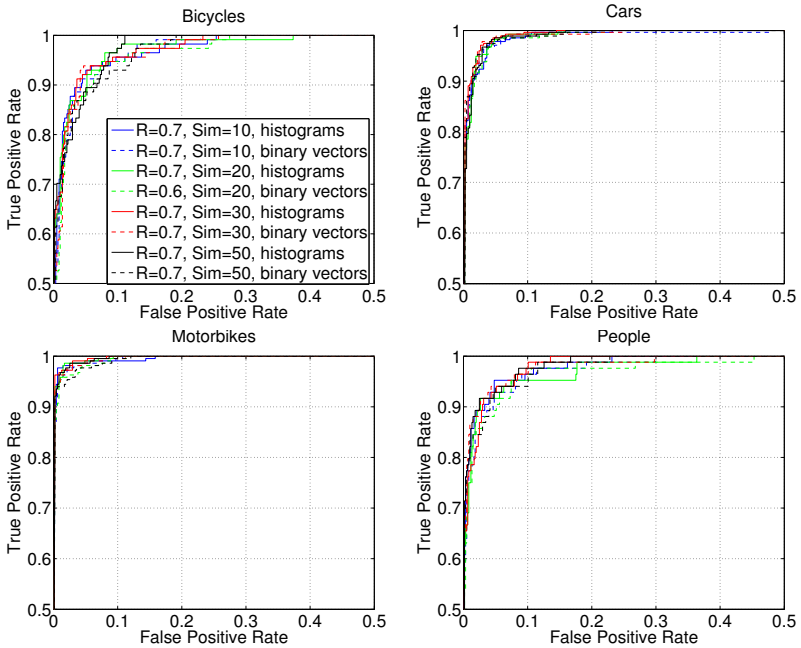


FIG. 2.7 – *Influence du nombre de centres choisis à chaque itération.*

Les courbes (2.7) nous montrent que la découverte d'un nombre assez large de centres au sein d'un même échantillonnage n'affecte pas la qualité des résultats obtenus, pour la plage de valeurs retenues.

Nombre de vecteurs échantillonnés par itération

Plus le nombre d'échantillons à disposition est grand, plus il est représentatif des données, mais plus la complexité augmente.

On compare ici différentes exécutions de la méthode, pour différentes taille d'échantillonnage, dans lesquels 20 centres sont extraits. Nous faisons varier le nombre de vecteurs échantillonnés de 500 à 5000.

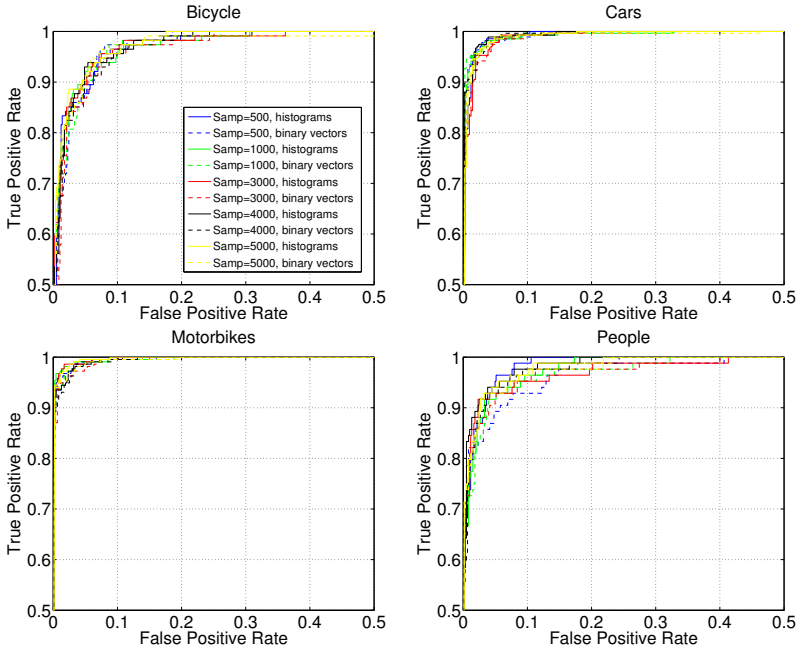


FIG. 2.8 – *Influence du nombre d'échantillons sélectionnés par itération.*

D'après la figure 2.8, ce paramètre n'a que peu d'influence sur les résultats de la classification, pour les plages de valeur choisies. Le nombre d'échantillons retenu pour le reste des expériences est de 3000 par itération.

Images entières ou région contenant l'objet

Pour faciliter l'apprentissage, et accélérer le processus de création de vocabulaire, nous avons utilisé les annotations fournies avec les images pour extraire les objets de chaque image selon leur boîte englobante[3]. C'est notre nouvelle base d'apprentissage « découpée ». Nous extrayons une centaine de descripteurs par image, pour un total de 300 000 points.

Les résultats obtenus sont comparés entre la base « découpée » et la base « non-découpée ». Dans les deux cas, l'apprentissage du classifieur se fait sur la base d'apprentissage initiale, non sur la base « découpée ».

Pour ces deux configurations, les résultats sont comparables, comme le montre la figure 2.9. L'algorithme est donc capable de déterminer quelles informations représentent convenablement les objets, même si leur position dans l'image n'est pas connu.

[3]des patchs provenant du fond (sans objet) et de même taille sont également extraits de manière à représenter le fond

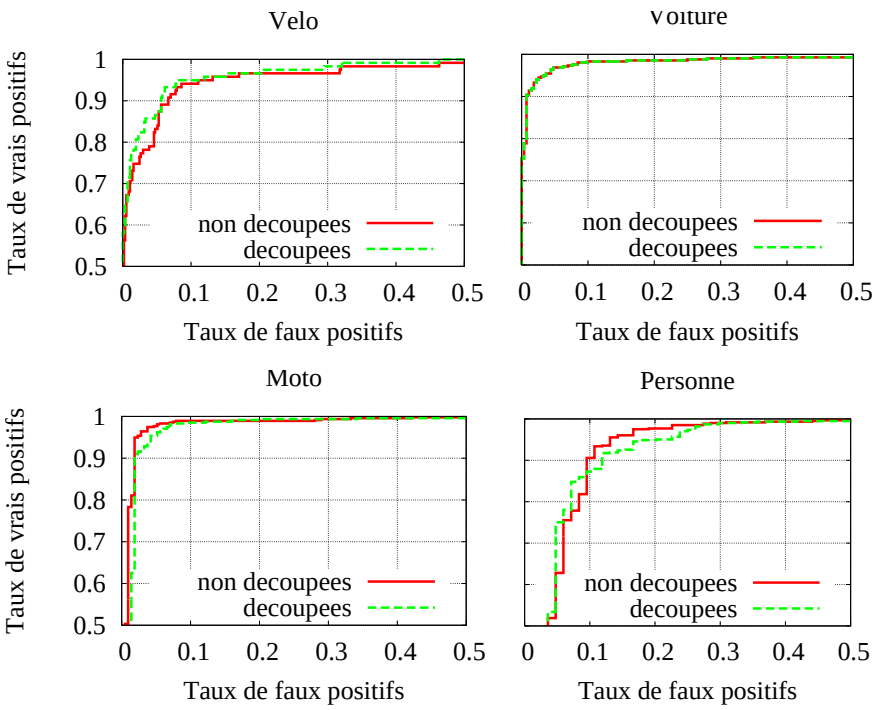


FIG. 2.9 – *Performances du classifieur selon que les images d'apprentissage sont découpées (pour ne contenir que les objets) ou non*

Nombre de mots du vocabulaire

La taille du vocabulaire choisi est relativement grande pour toutes les expériences proposées. Ici, nous cherchons à observer l'influence du nombre de mots sur les résultats de la classification. La figure 2.10 présente l'évolution des performances pour 3 approches différentes.

Dans tous les cas, nous partons d'un vocabulaire initial de 4000 mots, dont nous allons réduire la taille. La première courbe représente les histogrammes recalculés par affectation complète pour le cardinal du vocabulaire choisi. La deuxième courbe représente l'histogramme calculé par affectation totale pour les 4000 éléments du vocabulaire, tout simplement tronqué au nombre d'éléments voulus. Ceci est une approximation discutable, dont on souhaite mesurer les effets. La dernière courbe part des histogrammes sur 4000 mots où l'affectation des points aux centres se fait uniquement à l'intérieur de la sphère d'influence. Ainsi, seules les colonnes de l'histogramme choisies peuvent être sélectionnées. Cette représentation est exacte sans aucun calcul supplémentaire.

La figure 2.10 présente les résultats obtenus. Les meilleurs résultats sont bien entendu obtenus pour les histogrammes entièrement recalculés, mais l'approximation proposée est acceptable. L'utilisation de rayons donne les plus mauvais résultats, c'est cependant l'approche que nous utiliserons quand nous auront à sélectionner certaines primitives (dans la section 2.4.4), afin d'avoir des histogrammes exacts.

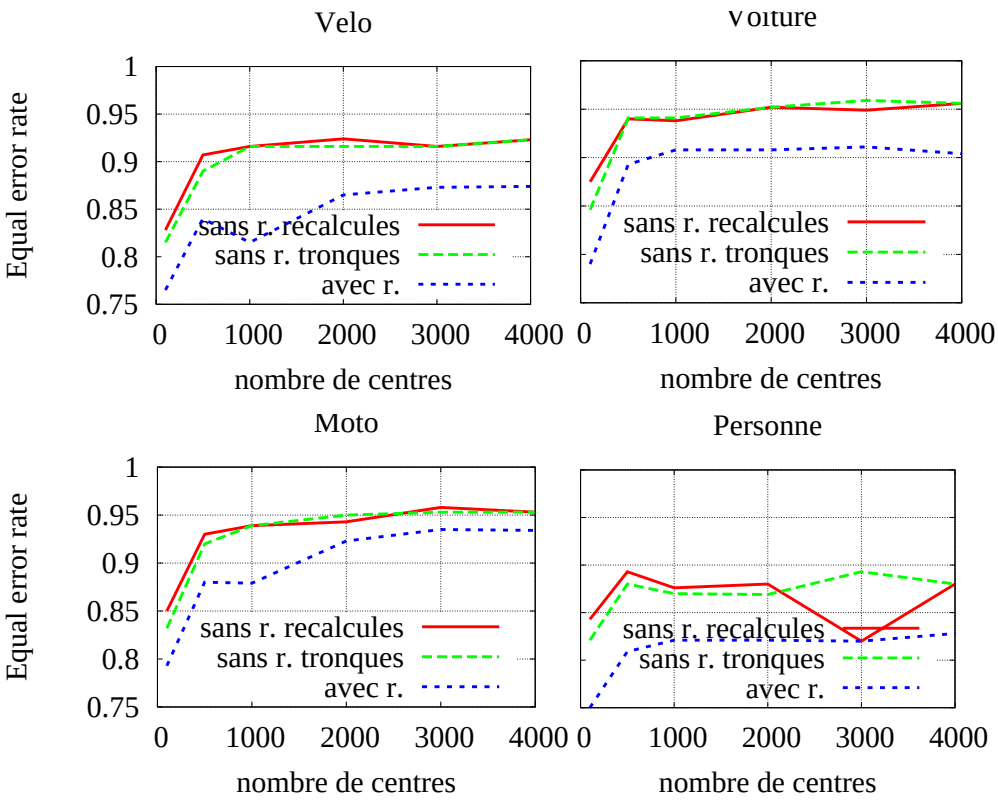


FIG. 2.10 – *Performances en fonction du nombre de mots produits pour le vocabulaire (voir texte).*

2.4.3 Comparaison avec différentes façons de créer un vocabulaire visuel

Nous avons comparé la méthode proposée avec des façons plus classiques de créer un vocabulaire visuel. Tout d'abord nous avons comparé avec une méthode qui n'utilise pas d'échantillonnage biaisé, et ensuite avec un vocabulaire construit classiquement à l'aide le l'algorithme des k-moyennes.

Comparaison avec un échantillonnage non biaisé

La méthode générative utilisée est comparée à un algorithme comportant également deux phases : l'une d'échantillonnage et l'autre de clustering. La phase de clustering, également réalisée par l'algorithme online median, tient compte des centres déjà placés, comme précédemment. La seule différence est que l'échantillonnage est uniforme, et non plus biaisé par un rayon d'influence.

Les expériences permettent de montrer l'importance de l'échantillonnage biaisé, qui guide le clustering, et évite de marquer plusieurs fois des régions fortement peuplées. Il dirige les centres vers des régions rares, mais qui s'avèrent informatives. Les résultats sont donc meilleurs avec le réchantillonnage biaisé. Des valeurs numériques pour le taux de vrais positifs à l'EER sont données dans la table 2.1.

	vélo	voiture	moto	personne
Méthode avec réchantillonnage biaisé	0.923	0.956	0.954	0.881
Méthode avec réchantillonnage uniforme	0.90	0.947	0.952	0.869

TAB. 2.1 – *Taux de vrais positifs à l'EER, pour différents algorithme de clustering. La deuxième ligne est un algorithme à deux phases très proche de la méthode du chapitre, à la différence que le réchantillonnage n'est pas biaisé.*

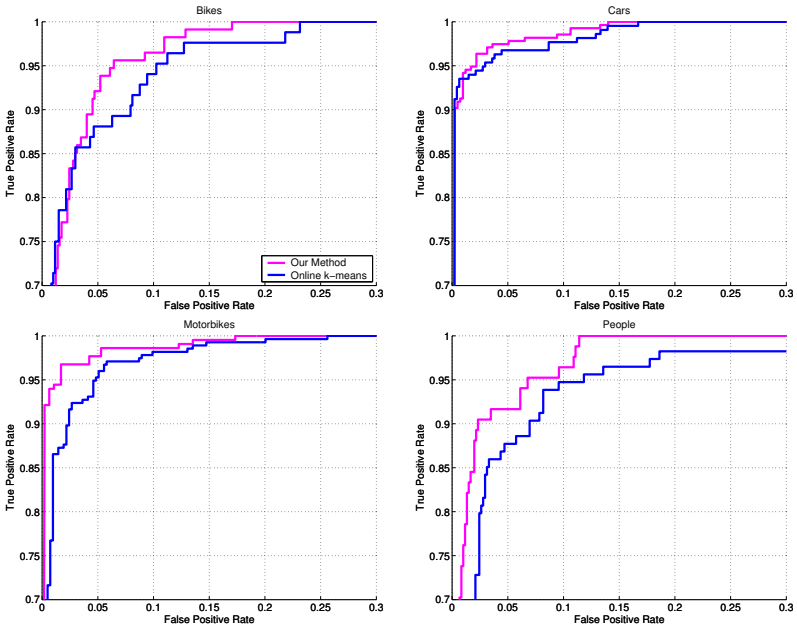


FIG. 2.11 – *Comparaison de notre méthode avec l'algorithme des k-moyennes.*

Comparaison avec l'algorithme des k-moyennes

Nous avons comparé les résultats de notre méthode avec une version « online » des k-moyennes. Les résultats présentés figure 2.11 montrent la supériorité de notre méthode.

Nous pouvons également comparer le temps nécessaire pour construire le vocabulaire pour chacune des méthodes. Pour notre implémentation, avec deux ensembles de paramètres différents (20 centres par itérations et 3000 échantillons pour chaque étape de clustering pour le premier ensemble - 200 centres par itérations et 5000 centres pour le deuxième). Les résultats sont présentés sur la table 2.2.

nb mots	notre méthode #1	notre méthode #2	online k-means
100	261	-	2712
200	540	166	2895
500	1284	343	3397
1000	2715	801	4421
2000	5052	1518	6390
3000	7504	2163	8330

TAB. 2.2 – *Comparaison des temps d'exécution (en secondes) pour deux paramétrages différents de notre algorithme et de l'algorithme des k-moyennes.*

2.4.4 Réduction de la dimension

La figure 2.12 présente les résultats de la classification en fonction du nombre d'éléments du vocabulaire, après sélection des mots les plus pertinents.

Cette fois les meilleurs éléments parmi les 4000 produits sont choisis, à l'aide de 3 méthodes différentes : l'information mutuelle, l'*odds ratio* (ou rapport des chances), et une sélection directement faite par le classifieur SVM. Cette dernière méthode donne généralement les meilleurs résultats.

Ces expériences ont été réalisées avec la méthode avec réchantillonnage biaisé, en utilisant le rayon d'influence pour le calcul des histogrammes. Nous pouvons ainsi extraire les coordonnées de l'histogramme correspondant aux éléments du vocabulaire sélectionné, pour obtenir les histogrammes du vocabulaire réduit, sans aucun autre calcul.

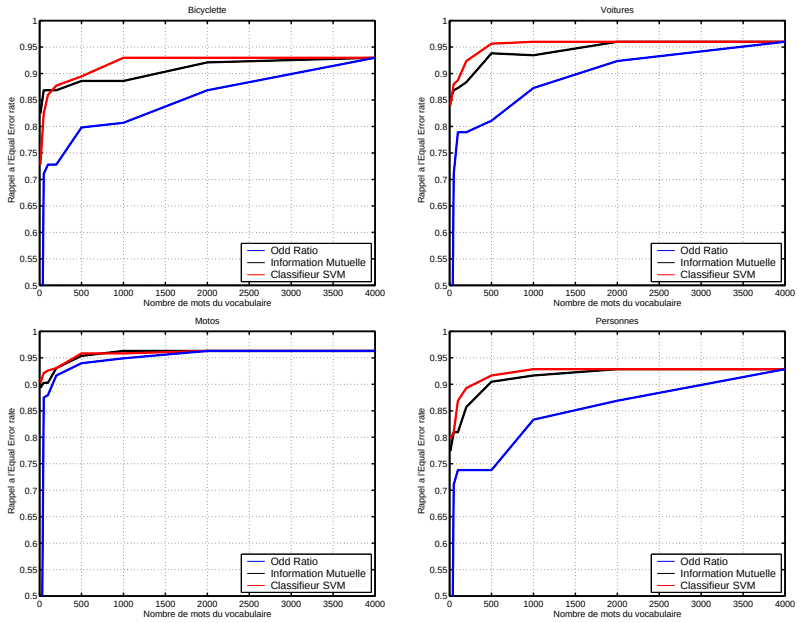


FIG. 2.12 – *Sélection de primitives par trois méthodes différentes : l'information mutuelle, l'odds ratio et le classifieur SVM (voir texte).*

Il s'avère que très peu de primitives, judicieusement choisies, suffisent à s'approcher des résultats obtenus par le vocabulaire complet.

2.4.5 Comparaison sur des bases standard

Pascal VOC 2005

Pour la compétition Pascal VOC 2005, toutes les meilleures méthodes sont basées sur l'approche sac-de-mots. Nous avons obtenu les meilleurs résultats sur presque toutes les classes, ce qui montre que l'extraction dense de descripteurs, combinée à un vocabulaire visuel créé de façon à faire face aux grandes quantités de données déséquilibrées fonctionne mieux que les méthodes basées sur des points d'intérêt. La valeur d'EER obtenue sur les 4 classes, pour les meilleures méthodes, ainsi que la courbe ROC pour la catégorie vélo pour tous les participants sont présentés table 2.3 et figure 2.13. Tous les résultats sont disponibles dans [16].

	moto	vélos	personnes	voitures
INRIA_L (Notre méthode)	**0.977**	**0.930**	**0.901**	**0.938**
INRIA_Z	0.964	0.930	0.917	0.937
Southampton	0.972	0.895	0.881	0.913

TAB. 2.3 – *Résultats d'EER pour la base Pascal VOC 2005 obtenus par les meilleures méthodes ayant participé à la compétition.*

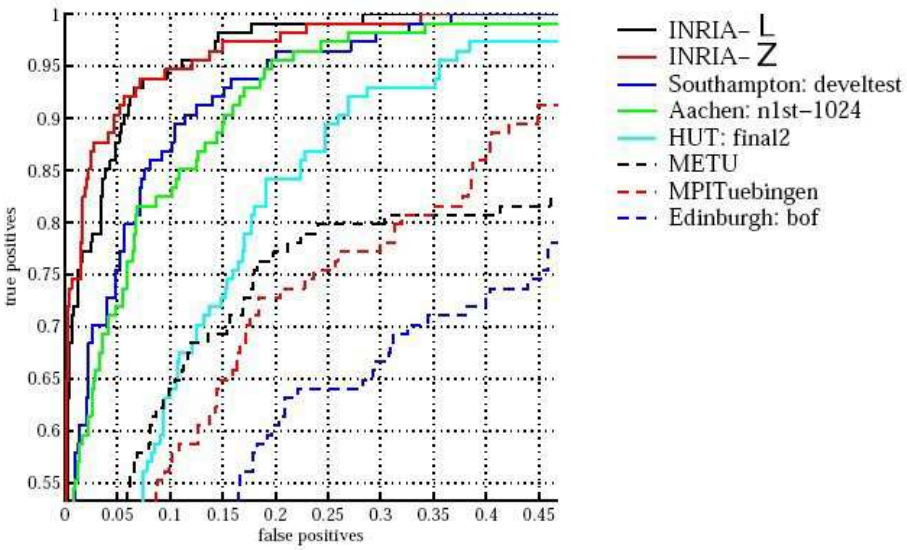


FIG. 2.13 – *La courbe ROC représentant tous les participants du Pascal Challenge 2005, pour la catégorie vélo. La figure provient du document [16].*

	Vélo vs Fond	Voiture vs Fond	Personnes vs Fond
Opelt *et al* [68]	0.765	0.702	0.81
Moosmann *et al* [62]	0.844	0.799	-
Notre méthode	0.867	0.8	0.88

TAB. 2.4 – *Résultats pour la base TU-Graz02, pour 6000 mots*

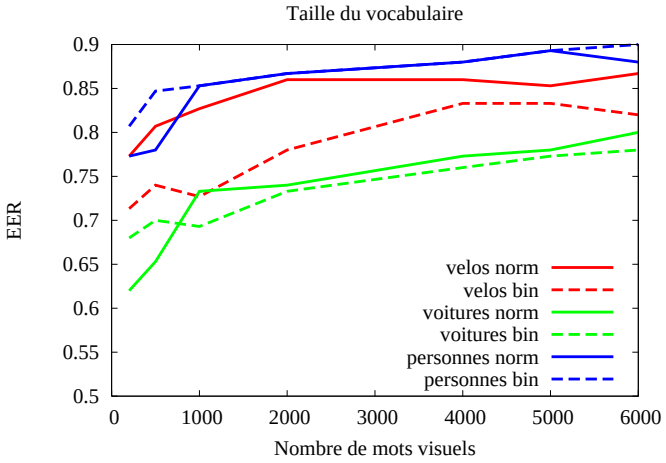


FIG. 2.14 – *Influence de la taille du vocabulaire sur la base TU-Graz02, pour deux normalisations différentes.*

Base TU-Graz 02

Nous avons complété notre étude par des expériences réalisées sur la base TU-Graz02 (voir section 1.3.2). Les descripteurs sont extraits selon une grille dense en position et en échelle, en suivant les mêmes paramètres que précédemment. Le vocabulaire visuel est construit en utilisant les paramètres de l'algorithme donnant les meilleurs résultats sur notre précédent ensemble d'expérience (rayon d'influence de 0.6, 20 centres ajoutés par itération, et 3000 échantillons).

La table 2.4 montre que notre méthode surpasse les résultats de la méthode initialement proposée sur cette base par Opelt *et al* [68] ainsi que des résultats récents obtenus par Moosmann *et al* [62].

Les résultats de la courbe de la figure 2.14 confirment qu'augmenter la taille du vocabulaire améliore les résultats.

Conclusion

Nous avons proposé dans ce chapitre une méthode efficace pour la production des vocabulaires visuels, lorsque les descripteurs sont échantillonnés densément dans les images. Elle compense le déséquilibre présenté par les données et permet une construction rapide du vocabulaire visuel malgré la prise en compte de millions de descripteurs. C'est pourquoi cette méthode est utilisée avec succès dans le chapitre 6, lorsqu'il s'agit de créer des vocabulaires visuels de 10 000 mots.

Cependant, les clusters étant définis comme des sphères autour des centres représentant les mots visuels, l'association d'un descripteur à un mot

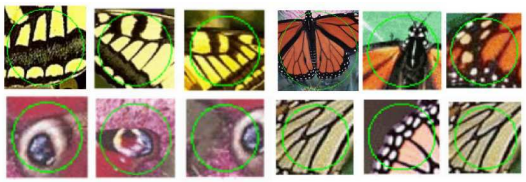
est basée sur des calculs de distance dans des espaces de grande dimension et la complexité associée augmente linéairement avec la taille du vocabulaire. Ainsi, l'affectation des descripteurs aux mots visuel est un processus coûteux. Nous verrons dans le chapitre 4 une autre méthode pour créer des vocabulaires visuels larges. Basée sur des arbres, elle permet de s'affranchir des calculs de distance entre les descripteurs et des centres de cluster. Une comparaison entre ces deux approches est proposée dans la section 4.5.4.

Vocabulaires discriminants pour la classification d'images

3

Sommaire

Les mots d'un vocabulaire visuel sont en général choisis pour modéliser au mieux les descripteurs locaux extraits des images. En ce sens, ils ne sont pas forcément adaptés à la tâche de classification. Fort de cette constatation, nous présentons dans ce chapitre un algorithme de quantification conçu pour produire des vocabulaires adaptés à la classification des images. Pour ce faire, le vocabulaire est décrit comme une partie du modèle de classes, et est appris pendant son estimation.



3.1 Description du problème

Nous nous intéressons dans ce chapitre, comme dans le chapitre précédent, à la tâche de classification d'images. Rappelons que l'objectif est d'affecter à une image le label d'une classe, si au moins une occurrence d'un objet de cette classe est présent. Nous avons déjà présenté les travaux marquants de ce domaine dans le chapitre précédent (section 2.2) et parlé du modèle par *sac-de-mots*.

Dans les méthodes dont nous avons parlé, et quelque soit l'algorithme utilisé, le vocabulaire visuel est construit lors d'un processus distinct et indépendamment des labels des images. Or, contrairement au texte, les vocabulaires visuels sont des concepts artificiels (dans le sens où le vocabulaire n'est jamais explicitement connu), et ne sont pas définis de façon unique. Pourtant, la représentation des images et les performances de classification dépendent fortement de leur choix. Dans ce chapitre, nous proposons de définir des vocabulaires adaptés aux classes d'images à reconnaître.

D'autres auteurs se sont penchés sur cette question. Dans [98], Winn *et al* s'intéressent à ce problème et suggèrent de partir d'un vocabulaire universel de grande taille, puis de construire un vocabulaire compact et discriminant en regroupant les paires de mots jouant le même rôle. Une limitation de cette approche est que si des descripteurs visuels représentant des informations distinctes sont groupés par le vocabulaire initial, ils ne peuvent être séparés ultérieurement.

La construction de vocabulaires visuels adaptés a aussi été explorée par Perronnin *et al* [70]. Ils abordent ce problème en combinant un vocabulaire visuel universel avec des vocabulaires spécifiques à chaque classe. Le vocabulaire visuel universel décrit le contenu visuel associé à toutes les classes, tandis que le vocabulaire spécifique à chaque classe est obtenu en adaptant le vocabulaire universel à une classe particulière afin de mettre en valeur les spécificités de cette classe. Ces vocabulaires spécialisés ont pour but de mettre en valeur les différences non pas entre les classes, mais entre l'apparence moyenne codée par le vocabulaire universel et l'apparence d'une classe donnée. C'est pourquoi, si deux classes sont visuellement très similaires, il n'y pas de garantie d'obtenir des mots visuels permettant de les discriminer.

Notre approche s'inscrit dans le cadre des modèles à variables latentes d'aspect (introduits dans la section1.2.3). Ces modèles sont des extensions du modèle sac-de-mots. Citons parmi les plus marquants le modèle pLSA (*probabilistic Latent Semantic Analysis* [31]) ou bien le modèle LDA (*Latent Dirichlet Analysis* [4]). Ces modèles représentent les images comme des combinaisons de distributions spécifiques sur des variables latentes d'aspect, les *topics*. Ils nécessitent que les images soient transformées en mots visuels, qui dépendent, comme nous l'avons vu, des descripteurs visuels choisis (pour faire le lien entre les pixels et le contenu de l'image) et de la méthode de quantification utilisée. Cependant, pour toutes les applications de ces modèles aux tâches de classification d'objets [81] ou de scènes [72], le vocabulaire visuel est construit par un processus distinct, non supervisé (dans le sens où il est indépendant de la connaissance des classes d'objets).

L'approche proposée dans ce chapitre essaie d'aller plus loin encore dans la recherche d'un vocabulaire discriminant. Inspirés par [72, 81], nous proposons un modèle génératif basé sur des variables latentes d'aspect, qui mo-

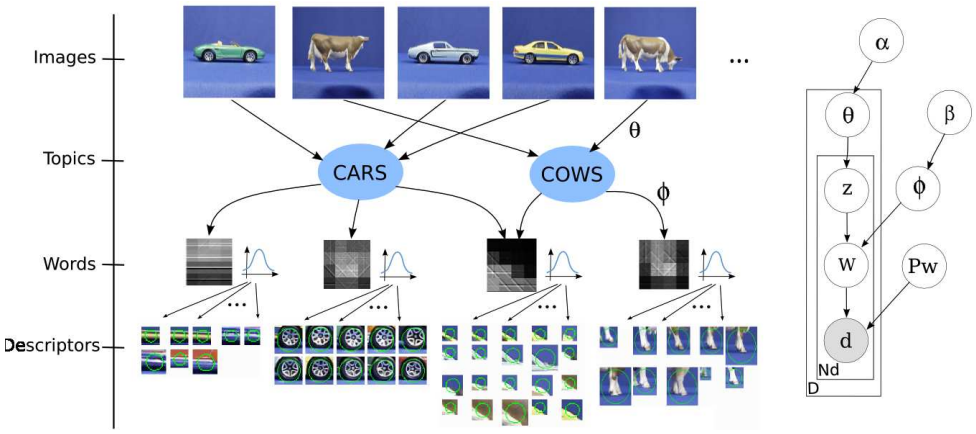


FIG. 3.1 – *Aperçu de la méthode et modèle graphique correspondant. Notre méthode modélise les images par un processus génératif. Chaque image génère des topics, qui représentent les catégories d'objets. Ces topics choisissent des mots visuels, qui eux même génèrent des descripteurs.*

délise les occurrences des descripteurs dans les images. Au lieu d'utiliser le vocabulaire visuel comme une étape de pré-traitement, celui-ci est un composant intégré à notre modèle, appris en même temps que les autres paramètres. En effet, nous considérons les images comme des distributions de *topics*, les topics comme des distributions de *mots visuels*, et les mots comme des distributions gaussiennes dans l'espace des descripteurs. Les variables latentes sont donc ici à la fois les topics et les mots visuels. Le processus génératif associé à une image est illustré figure 3.1.

Une loi de Dirichlet est utilisée comme a priori pour les distributions sur les topics et sur les mots, et incite le modèle à produire peu de topics par image, quelques mots spécifiques pour chaque classe, ainsi que des mots génériques partagés entre les classes.

Notons que notre modèle peut être appris sans aucune supervision, mais nous verrons plus loin qu'un peu de supervision rend l'estimation du modèle plus stable.

Le reste du chapitre est organisé comme suit. La section 3.2 présente le modèle proposé ainsi que la façon de l'estimer. La section 3.3 explique comment les paramètres du modèle sont utilisés pour faire de la classification d'images. Notre modèle peut être utilisé avec différentes stratégies de classification, qui sont décrites dans cette section. Enfin, dans la section 3.4 nous présentons les expériences, puis nous concluons le chapitre.

3.2 Modélisation des statistiques d'apparence locale

3.2.1 Modèle génératif

Les images sont considérées comme des ensembles non-ordonnés de descripteurs visuels, choisis à l'aide d'un détecteur de points d'intérêt, ou échantillonnés uniformément dans les images. En pratique, nous avons choisi d'extraire les patchs selon une grille régulière multi-échelle, comme suggéré par [36] et [98]. Nous avons également choisi d'utiliser des descripteurs SIFT (voir définition dans la section 2.1.1), dans un espace de dimension 128,

mais d'autres descripteurs auraient très bien pu être utilisés. La position et l'échelle de ces descripteurs ne sont pas retenues par la suite.

Nous avons utilisé une version simplifiée du modèle GM-LDA (Gaussian-Multimodal LDA [98]), qui est un modèle à variables latentes d'aspect, expliquant statistiquement la génération des descripteurs dans les images. Les descripteurs visuels proviennent de deux facteurs cachés (les variables latentes), que nous appellerons *topics* et *mots*. Les images sont modélisées comme des combinaisons de T topics possibles qui produisent eux-mêmes N mots visuels. Les mots visuels sont des distributions gaussiennes sur l'espace des descripteurs SIFT. Les T distributions des topics sur les mots (ϕ) sont échantillonnées selon une distribution de Dirichlet de paramètre β, et les distributions des images sur les topics (θ) sont échantillonnées selon une loi de Dirichlet de paramètre α.

Modéliser une image I avec notre modèle suppose donc qu'elle ait été construite à partir du processus génératif suivant.

1. Échantillonnage d'une valeur $\theta \sim Dir(\alpha)$, où $Dir(\alpha)$ est une loi de Dirichlet d'hyper-paramètre α, fournissant ainsi une distribution sur les variables latentes *topics*.
2. Pour chaque descripteur d'image d_i :
   (a) échantillonnage d'un topic z_i depuis la distribution multinomiale de paramètre θ : $z_i \sim Mult(\theta)$;
   (b) échantillonnage d'un mot visuel w_i conditionnellement au topic z_i à partir de la loi multinomiale de paramètre ϕ_{z_i}, $w_i \sim Mult(\phi_{z_i})$;
   (c) échantillonnage d'un descripteur d_i conditionnellement à w_i, $d_i \sim \mathcal{N}(P_{w_i})$, où $\mathcal{N}(P_{w_i})$ désigne la distribution gaussienne de paramètre P_{w_i}.

La distribution conjointe sur les descripteurs visuels d'une image I est donnée par la formule suivante :

$$p(d|P,\phi,\alpha,\beta,I) = \int \prod_{d_i \in I} \sum_{j=1}^{N} \sum_{k=1}^{T} p(d_i|w_j,P)p(w_j|z_k,\phi)p(z_k|\theta)p(\theta|\alpha)d\theta \tag{3.1}$$

Comparée aux méthodes proposée dans [18, 19, 72, 81], notre modèle possède une couche supplémentaire qui explique la génération des descripteurs à partir des mots visuels. Cette couche est le composant clef de notre modèle, puisqu'il permet un apprentissage simultané du modèle et du vocabulaire visuel.

La représentation graphique associée à notre modèle est présentée figure 3.1.

3.2.2 Estimation du modèle

L'apprentissage d'un modèle est fait par une maximisation de la vraisemblance, qui est obtenue par l'estimation des paramètres optimaux pour les variables α, β, ϕ et θ, pour un ensemble d'images donné.

Les hyper-paramètres α et β jouent un rôle important puisqu'ils permettent, par l'utilisation de valeurs particulières, de contrôler les distributions sur les topics et les mots, afin de les rendre éparses et donc spécialisées.

C'est pourquoi, tout comme [26], nous préférons ne pas les estimer, et utiliser des hyper-paramètres fixes.

Comme l'intégrale dans l'équation 3.1 rend l'optimisation directe de la vraisemblance impossible, nous estimons les variables par la technique d'approximation itérative appelée *échantillonnage de Gibbs* (ou *Gibbs sampling*). C'est un cas particulier de la méthode de chaîne de Markov Monte Carlo (MCMC), où une chaîne de Markov est construite de façon à converger vers une distribution cible. L'état suivant de cette chaîne est atteint par un échantillonnage séquentiel de toutes les variables à partir de leur distribution, conditionnellement à la valeur courante de toutes les autres variables et aux données.

Dans notre modèle, nous utilisons le fait que les a priori (α et β) sont conjugués aux distributions multinomiales ϕ et θ, afin de considérer la distribution a posteriori sur les affectations des topics $p(z|w)$. Une justification complète peut être trouvée dans [26] pour le modèle LDA.

Le processus d'estimation est fait par échantillonnage des distributions. Notre méthode d'estimation est très similaire à [26][1] qui présente un algorithme efficace pour l'estimation du modèle LDA.

Le procédé d'échantillonnage est fait par échantillonnage des distributions $p(z_i|z_{-i}, w)$ et $p(w_i|d_i, w_{-i}, z, P)$ pour toutes les observations d_i, où z_{-i} représente toutes les valeurs z sauf z_i. La première distribution est obtenue via la formule suivante :

$$p(z_i = j|z_{-i}, w) \propto \frac{(n^{w_i}_{-i,j} + \beta)(n^I_{-i,j} + \alpha)}{(n^{(.)}_{-i,j} + W\beta)(n^I_{-i,.} + T\alpha)} \tag{3.2}$$

avec $n^w_{-i,j}$ représentant le nombre de fois qu'un mot w a été affecté au topic j en excluant l'observation considérée i, et $n^I_{-i,j}$ étant le nombre de fois qu'un mot de l'image I a été affecté au topic j. $n^{(.)}_{-i,j}$ représente le nombre total de mots affectés au topic j et $n^I_{-i,.}$ est le nombre d'observations dans l'image I, en excluant le descripteur d_i.

La deuxième distribution vient de l'équation :

$$p(w_i|d_i, w_{-i}, z, P) \propto p(d_i|w_i, P)p(w_i|t_i) \tag{3.3}$$

où $p(w_i|t_i)$ correspond à ϕ^w_j pour $w_i = w$ et $t_i = j$ qui peut être obtenu par $\frac{n^w_{-i,j}+\beta}{n^{(.)}_{-i,j}+W\beta}$. D'après le modèle, la distribution $p(d_i|w_i, P)$ est égale à $\mathcal{N}(P_{w_i})$. P correspond aux paramètres de la mixture de gaussiennes qui décrit les mots et est réestimée à chaque itération en utilisant des techniques d'échantillonnage classiques. Le processus d'estimation itératif est initialisé en utilisant des distributions équiprobables sur les topics, et un algorithme des k-moyennes (*k-means*) est utilisé pour créer les mots visuels initiaux.

Les paramètres du modèle peuvent être estimés sans aucune supervision, c'est-à-dire en utilisant des images d'apprentissage non labellisées. Si nous marginalisons la probabilité $P(\theta, \phi, P, d_i)$ sur les variables ϕ, P et d_i, $P(\theta)$ devrait avoir des modes corrélés aux vraies classes, permettant ainsi d'obtenir des mots visuels spécialisés pour les classes. Malheureusement, nous avons observé en pratique que ce n'est le cas que lorsque le fond des images

[1] les justifications et les détails d'implémentation sont disponibles dans cette référence.

n'est pas trop encombré et que l'objet occupe une partie importante de l'image. Dans le cas contraire, l'échantillonneur de Gibbs reste bloqué dans des modes non désirés (minima locaux), qui dépendent fortement de l'initialisation. Nous avons observé que ce comportement indésirable pouvait être réduit, voire supprimé, en ajoutant de la supervision, c'est-à-dire en indiquant pour quelques images d'apprentissage, leur appartenance aux classes considérées. Dans tous les cas, nous avons observé expérimentalement que ce type de supervision rend l'estimation plus précise et permet d'améliorer considérablement les performances de la méthode.

3.3 Utilisation du modèle pour la classification

Une fois le modèle appris, différents types d'information peuvent être utilisés pour classifier les images, selon la représentation choisie.

3.3.1 Classification par maximum de vraisemblance basée sur les topics

Les topics ont été construits de façon à être de bons représentants du contenu des images. Une façon toute naturelle de classifier les images est donc la règle du maximum de vraisemblance, utilisant ces topics. La façon la plus directe d'implémenter cette règle est de choisir autant de topics que de classes et de supposer que les probabilités sur les classes sont les probabilités sur les topics, pour une image donnée.

Par exemple, si la classe C_i est représentée par le topic z_i dans l'image I, nous avons $p(C_i|I) = p(z_i|I) = \theta_i$. Nous avons observé expérimentalement que la chaîne de Markov générée par l'échantillonneur de Gibbs pour la distribution θ tend à converger rapidement vers des modes précis et stables. La sortie de l'échantillonneur est utilisée et l'intégrale est approximée par une somme de valeurs discrètes. Cette loi est appelée TOPIC-BAYES.

3.3.2 Classification par classifieur SVM entraîné sur les topics

Si nous voulons plus de topics que de classes d'objets, la règle précédente ne peut plus être appliquée. Nous avons adapté la règle de classification proposée par [72] à notre modèle. Leur protocole consiste à entraîner un classifieur sur les variables latentes associées à chaque image. Ceci ne peut être fait directement en utilisant notre modèle « LDA + gaussiennes », qui n'estime pas explicitement de valeurs numériques pour les variables latentes, mais des densités de probabilité.

Comme précédemment, nous utilisons la sortie de l'échantillonneur pour approximer une valeur correspondant à la distribution sur les topics de chaque image. Chaque image est ainsi représentée par un vecteur de probabilité de topics qui peut être directement utilisé par un classifieur SVM. Ce classifieur est appelé TOPIC-SVM.

3.3.3 Classification par sac-de-mots et classifieur SVM

Au lieu de classer les images à partir de leur distribution sur les topics, les images peuvent aussi être classées selon leurs statistiques sur les mots

visuels, comme cela est fait dans une approche traditionnelle par sac-de-mots. Comparer le modèle par sac-de-mots avec une modélisation basée sur les topics à partir du même modèle est un problème intéressant. Ce classifieur est notée LDA-VOC-SVM.

3.4 Expériences

Dans cette section, nous allons mettre en évidence l'intérêt des vocabulaires construits avec notre méthode. Les expériences sont divisées en deux : la catégorisation à partir des topics, qui sont les variables latentes de plus haut niveau de notre modèle, et la catégorisation d'images à partir des mots visuels selon un cadre sac-de-mots classique. Le même modèle est utilisé dans les deux cas, mais ce sont des paramètres différents qui sont considérés par le classifieur. Pour chacun des problèmes, nous allons comparer les résultats obtenus par notre méthode aux méthodes standards de la littérature. Nous allons également comparer les performances de notre classifieur basé sur les topics avec celui basé sur les mots, pour différents niveaux de supervision.

Deux méthodes ont été implémentées pour permettre la comparaison avec l'existant : une approche sac-de-mots, basée sur un vocabulaire visuel construit avec l'algorithme des k-moyennes, et un modèle LDA standard, utilisant également les k-moyennes pour construire le vocabulaire.

3.4.1 Bases de données

Pour les expériences suivantes, nous utiliserons la base ETH-80 (section 1.3.1), la base des oiseaux (section 1.3.3), la base des papillons (section 1.3.4), et la base Pascal VOC 2005 (section 1.3.5) qui ont toutes été décrites en détail dans l'introduction de ce mémoire.

3.4.2 Choix des paramètres

Pour toutes les expériences présentées ici, les descripteurs locaux sont extraits à partir d'une grille dense, en position et en échelle. Nous avons observé que lorsque notre approche était basée sur des détecteurs de points d'intérêt, celle-ci donnait de moins bons résultats. Les paramètres utilisés fournissent environ 800 patchs par image pour la base ETH, et 1500 patchs par image pour les bases d'oiseaux et de papillons, ainsi que pour la base Pascal. Chaque patch est représenté par un vecteur SIFT [54] de dimension 128.

Nous avons supposé des lois a priori de Dirichlet symétriques, α et β, ayant une valeur scalaire fixe. Cette connaissance a priori sur les distributions multinomiales contrôle le mélange des poids des multinomiales. L'utilisation d'hyper-paramètres de valeurs faibles encourage les distributions à être éparses. Les images choisissent donc en général un faible nombre de topics, et les topics un faible nombre de mots. Nous avons utilisé les valeurs $\alpha_i = \beta_i = 0.5, \forall i \in \{1, .., T\}$.

Nous avons observé que l'échantillonneur de Gibbs converge après moins de 50 itérations, ce qui est le nombre utilisé dans nos expériences. Cela prend jusqu'à 12 heures pour traiter les bases les plus importantes avec les vocabulaires les plus larges. Notons également que pour réduire le coût

de stockage des descripteurs, la valeur de leur représentation SIFT a été quantifiée dans un espace très large mais discret.

Lorsque rien n'est précisé, les résultats reportés sont des performances multi-classes, obtenues en combinant des classifieurs SVM 1 contre 1. Nous reportons la moyenne et la variance de 5 exécutions avec différentes initialisations aléatoires. Sauf lorsqu'une valeur différente est spécifiée, le vocabulaire visuel comporte 1000 mots.

3.4.3 Classification basée sur les topics

De façon idéale, les méthodes basées sur des variables latentes d'aspect peuvent être complètement non-supervisées, comme cela a été montré par Sivic *et al* [81]. Le nombre de topics peut être fixé comme égal au nombre de catégories, et chaque catégorie n'est représentée que par un seul topic.

Cependant, nous sommes convaincus que les classes sont un concept hautement sémantique, qui repose plus sur une connaissance humaine, que sur des caractéristiques visuelles bas-niveau. Pour illustrer notre propos, la figure 3.2 propose une estimation non supervisée de notre modèle avec 10 topics sur la base ETH-80. Les 8 images correspondant le mieux au topic (*e.g.* possédant la plus forte probabilité de générer le topic considéré) sont présentées pour chacun de ces 10 topics. Nous observons que les topics ont été découpés entres les différentes classes mais aussi entre les différentes vues des classes. Et ces différentes vues, pour les voitures par exemple, sont visuellement très différentes. Il semble peu probable qu'elles soient groupées de façon complétement non supervisée en un semble topic semble.

De plus, nous avons observé pendant les expériences que même dans les cas simples, les topics coïncident difficilement avec les vraies classes. Plus précisément, il y a de nombreux minima locaux, qui rendent l'issue du processus d'estimation très dépendante de l'initialisation. Une solution peut être d'utiliser les topics dans un cadre (faiblement) supervisé. Dans ce cas, des labels de classes sont utilisés pour réduire le nombre de paramètres du modèle et encourager les topics à correspondre aux classes. Ensuite nous avons utilisé soit un simple classifieur bayésien pour affecter les images au label du topic le plus probable, noté TOPIC-BAYES, soit un classifieur qui considère les vecteurs de topics comme des vecteurs représentants d'image. Cette deuxième façon de faire est notée TOPIC-SVM.

En utilisant ces deux stratégies basées sur les topics, les topics produits par notre modèle (noté LDA-VOC) ont été comparés à ceux produits par une méthode de référence - le modèle LDA - qui n'apprend pas le vocabulaire (notée STD-LDA, pour LDA standard).

La table 3.1 résume les expériences conduites sur les bases d'images ETH-80 et d'oiseaux. Chaque ligne correspond à différents niveaux de supervision, depuis 0 image labellisée (cas complètement non-supervisé) jusqu'à un plus grand nombre. Notons que le cas non-supervisé n'est pas applicable avec la stratégie TOPIC-SVM qui nécessite au moins 1 image labellisée par classe. Sans aucune supervision, la variance est très grande dans le meilleur des cas (base ETH-80), tandis que pour les cas plus difficiles (base d'oiseaux) la classification n'est pas possible puisque les topics ne correspondent pas du tout aux classes. La supervision aide le système à produire de meilleurs résultats,

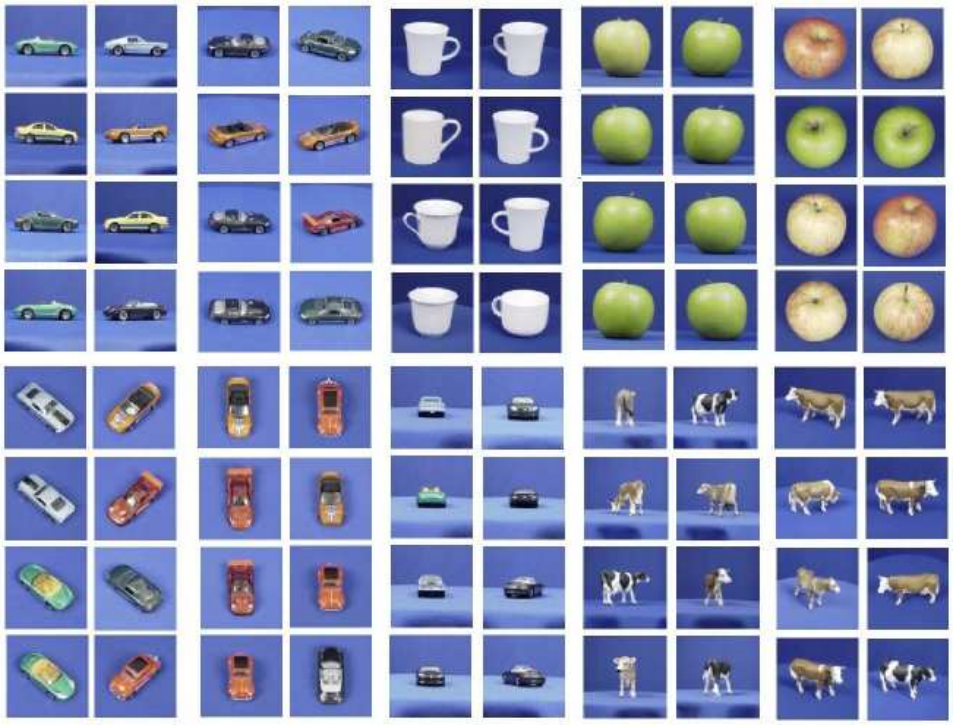


FIG. 3.2 – *Notre modèle a été appris avec 10 topics, sur les 4 classes de la base ETH-80, de façon totalement non supervisée. Chaque image montre les 8 images choisissant le plus probablement l'un des topics : les topics découpent entre les classes d'objets, mais aussi entre les points de vue des objets. Grouper ces différentes vues, sans supervision, à l'aide de descripteurs locaux uniquement, nous semble un résultat intéressant.*

ETH-80	TOPIC-BAYES				TOPIC-SVM			
nombre d'image labellisées	LDA-VOC		STD-LDA		LDA-VOC		STD-LDA	
	Moy.	Var.	Moy.	Var.	Moy.	Var.	Moy.	Var.
0	88.92%	12.43	-	-				
8	96.42%	1.53	94.62%	0.05	96.8%	1.12	94.6%	0.18
176	98.73%	0.08	97.16%	0.03	98.72%	0.25	97.19%	0.15
Oiseaux	TOPIC-BAYES				TOPIC-SVM			
nombre d'image labellisées	LDA-VOC		STD-LDA		LDA-VOC		STD-LDA	
	Moy.	Var.	Moy.	Var.	Moy.	Var.	Moy.	Var.
0	-	-	-	-				
66	44.01%	0.21	-	-	43.6 %	0.26	39.1%	0.46
198	55.97%	0.2	50.3%	1.01	55.6%	0.22	50.3%	1.02
300	60.68%	0.72	54.5%	0.6	60.67%	0.75	54.4%	0.75

TAB. 3.1 – *Les résultats des stratégies TOPIC-BAYES et TOPIC-SVM pour la base ETH-80 et la base d'oiseaux. Chaque ligne représente une quantité différente de supervision (images labellisées). Les performances moyennes ainsi que la variance sont reportées. « - » signifie que les topics ne peuvent pas être associés aux classes.*

plus stables (variance plus faible) : elle ne devrait pas être considérée comme optionnelle.

Il est important de noter que pour les deux bases d'images, et sous les différentes configurations, LDA-VOC produit de meilleurs résultats que la méthode LDA standard. Avec un vocabulaire créé spécifiquement pour la tâche de classification, l'estimation des topics est donc bien meilleure. De plus, notons que les stratégies TOPIC-BAYES et TOPIC-SVM donnent les mêmes résultats.

Les résultats sur la base ETH-80 sont très satisfaisants ; malgré un très grand nombre de points de vue, donner seulement 2 labels d'image par catégorie est suffisant pour obtenir un groupement de tous les points de vue dans la même catégorie. La base d'oiseaux est bien plus difficile, car même avec un grand nombre de labels, les performances sont relativement faibles. Cela semble indiquer que les topics ne sont pas le meilleur niveau d'information à utiliser pour classifier les images, en particulier lorsque le fond est encombré.

3.4.4 Classification par sac-de-mots

Dans ces expériences, nous estimons le modèle exactement comme cela a été fait dans la section précédente. Seulement, au lieu de classifier les images en utilisant les distributions de topics, une approche par sac-de-mots est considérée, utilisant le vocabulaire visuel produit par notre approche ; elle est notée LDA-VOC-BOF. Cette méthode de classification est comparée à une

nombre de mots	Papillons		Oiseaux	
	LDA-VOC-BOF	KMEANS-BOF	LDA-VOC-BOF	KMEANS-BOF
200	76.2 %	67.89 %	74.6 %	65.33 %
500	83.83%	78.57 %	85.1 %	76.58 %
1000	88.56%	84.65 %	89.0 %	83.33 %
2000	90.38%	85.77 %	90.9 %	86.17 %

TAB. 3.2 – *Comparaison du vocabulaire produit par notre modèle (LDA-VOC-BOF) avec un vocabulaire obtenu par une quantification par les k-moyennes de l'espace des descripteurs. (KMEANS-BOF).*

approche par sac-de-mots standard, notée KMEANS-BOF, qui utilise un algorithme des k-moyennes pour la quantification de l'espace des descripteurs. Dans tous les cas, un classifieur SVM linéaire est utilisé.

Les bases d'images sont découpées en 2 parties. La première, la base d'apprentissage, est labellisée et constitue la partie supervisée sur laquelle le modèle est appris. C'est aussi l'ensemble d'images utilisé pour entraîner le classifieur SVM. La deuxième partie constitue l'ensemble sur lequel est évalué la méthode. Pour la base d'oiseaux, le découpage apprentissage/test suit les recommandations faites par [49], et comprend 300 images par ensemble. Pour la base de papillons, le découpage est décrit dans [48], et donne 182 images d'apprentissage et 427 images de test. Enfin, sur la base Pascal VOC 2005, nous avons utilisé le découpage proposé pour la compétition [16].

Les résultats sont reportés dans les tables 3.2 et 3.4. Ils comportent des classifications moyennes pour différentes tailles de vocabulaire.

	cl 1	cl 2	cl 3	cl 4	cl 5	cl 6	
cl 1	43	3	1	1	2	0	86%
cl 2	3	45	0	1	1	0	90%
cl 3	1	0	49	0	0	0	98%
cl 4	0	0	1	49	0	0	98%
cl 5	1	1	0	1	47	0	94%
cl 6	0	3	0	1	0	46	92%
						Moy.	93%

	cl 1	cl 2	cl 3	cl 4	cl 5	cl 6	cl 7	
cl 1	79	0	2	0	1	2	1	92.9%
cl 2	0	16	0	0	0	0	0	100%
cl 3	0	2	52	2	0	1	0	91.2%
cl 4	2	0	0	41	5	0	0	85.4%
cl 5	1	0	1	9	47	0	0	81%
cl 6	3	0	2	0	0	103	0	95.4%
cl 7	3	0	0	4	0	0	58	89.2%
							Moy.	90.61%

TAB. 3.3 – *Matrices de confusion pour la meilleure exécution sur les bases d'oiseaux et de papillons. Le nombre d'images et les pourcentages correspondants sont présentés.*

Tout d'abord, considérons la table 3.2 à partir de laquelle nous pouvons faire 3 remarques. Premièrement, le vocabulaire produit par notre modèle obtient des performances moyennes de classification qui sont jusqu'à presque 10 % meilleures qu'une quantification basique des données par l'algorithme de clustering (dans le cas des oiseaux, ou des papillons). Ensuite, notons que le classifieur entraîné sur le modèle par sac-de-mots donne de bien meilleures performances que celui considérant uniquement les distributions sur les topics. Pour la base des oiseaux, un gain de plus de 30% est obtenu, ce qui peut être expliqué par la précision plus faible du modèle de topics pour des images aussi complexes.

Enfin, les performances globales obtenues par notre système sont très similaires aux meilleures performances reportées sur la base des oiseaux [49] et sur la base des papillons [48], bien que nous n'utilisions ni information

nombre de mots	Pascal VOC 2005									
	LDA-VOC-BOF					KMEANS-BOF				
classes	cl 1	cl 2	cl 3	cl 4	Moy	cl 1	cl 2	cl 3	cl 4	Moy
200	87.8	90.2	93.4	86.9	89.6	86.8	88.4	89.6	83.3	87.0
500	89.6	91.5	95.3	90.5	91.7	86.0	91.3	96.6	82.1	89.0
1000	89.5	92.7	97.0	91.2	92.6	89.7	92.7	96.6	89.9	92.3

TAB. 3.4 – *Comparaison entre le vocabulaire produit par notre modèle (LDA-VOC-BOF) et un vocabulaire produit par les k-moyennes. (KMEANS-BOF).*

de couleur, ni géométrie. En guise d'illustration, les deux tableaux de la table 3.3 montrent pour les bases d'oiseaux et de papillons la matrice de confusion associée à la meilleure exécution.

Ces expériences confirment notre intuition que, dans certaines situations, la classification d'images en utilisant les statistiques de mots visuels peut s'avérer bien meilleure que celle basée sur les distributions de topics. Nous avons essayé d'aller plus loin et de mesurer les limitations de ces méthodes, en fonction du nombre d'images d'apprentissage.

Nos expériences, illustrées par la figure 3.3, montrent que le modèle par sac-de-mots peut atteindre de meilleures performances mais nécessite plus d'images d'apprentissage, puisque la dimension du vecteur de représentation est plus grande. Les résultats présentés dans cette figure utilisent un modèle appris avec 12 images labellisées, qui est considéré à la fois au niveau des topics et des mots. Nous avons ajouté un nombre variable d'images labellisées pour entraîner le classifieur ; lorsque suffisamment d'images d'apprentissage sont disponibles, le modèle par sac-de-mots donne de meilleurs résultats que le classifieur basé sur les topics.

Intéressons-nous maintenant à la table 3.4, qui donne les résultats obtenus sur la base Pascal VOC 2005. Ces résultats peuvent être comparés aux résultats présentés précédemment dans le chapitre 2, qui correspondent également à la soumission gagnante de la compétition. Là où la méthode ga-

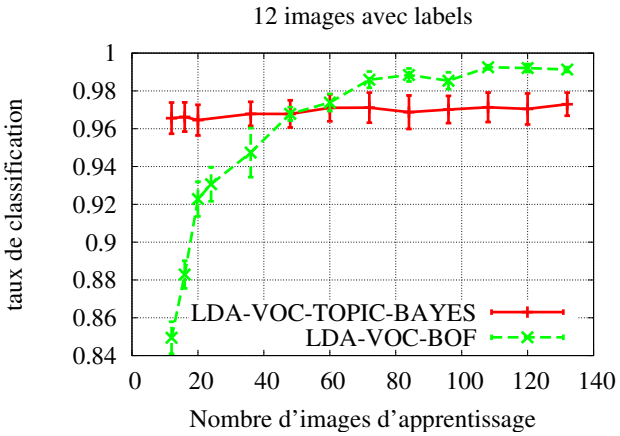


FIG. 3.3 – *ETH-80 : comparaison entre la classification basée sur les topics (TOPIC-BAYES) et celle basée sur les mots visuels (LDA-VOC-BOF), comme une fonction du nombre d'images d'apprentissage. Les deux représentations proviennent du même modèle.*

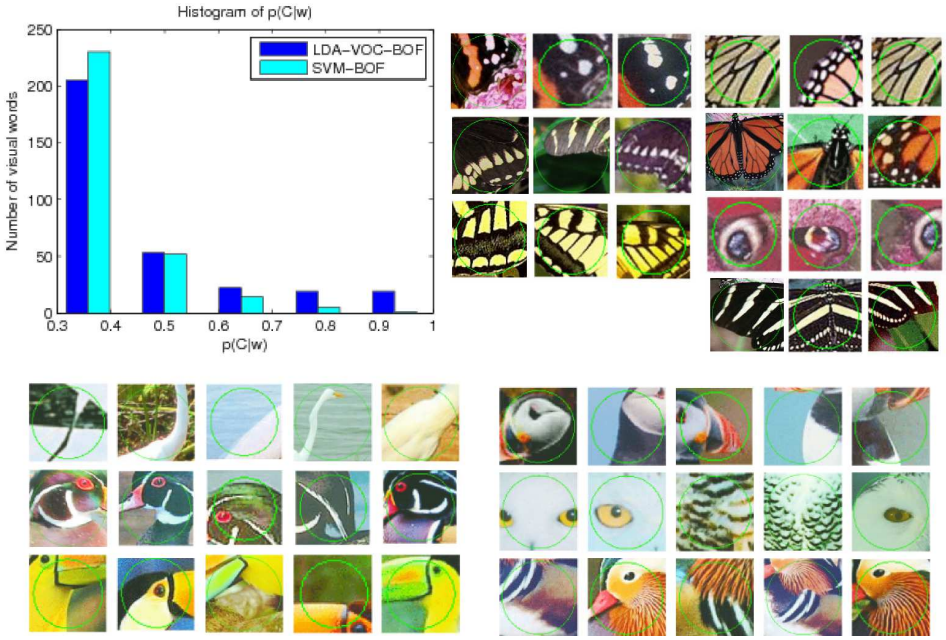


FIG. 3.4 – *Ligne du haut : à gauche, densité de probabilité sur les classes, conditionnellement aux mots visuels, et à droite, meilleurs mots par topics pour les papillons. Ligne du bas : 5 des meilleurs mots produits par topics pour les oiseaux.*

gnante obtenait 93% pour les vélos, 93.8% pour les voitures, 97.7% pour les motos et 90.1% pour les personnes, avec un vocabulaire de 4000 mots, nous obtenons ici, pour les même classes, respectivement 89.5%, 92.7%, 97% et 91.2% avec un vocabulaire de 1000 mots. Ce sont des résultats comparables, inférieurs pour trois classes, et meilleurs pour une classe. Nous obtenons donc des résultats comparables à la soumission gagnante du Pascal VOC 2005 (chapitre 2) en utilisant 4 fois moins de mots visuels. D'autres comparaisons sur la base Pascal V0C 2005 peuvent être faites en regardant la table 2.3.

3.4.5 Analyse statistique du vocabulaire

Notre motivation principale pour la création de vocabulaires simultanément avec d'autres paramètres était de produire des mots visuels qui soient plus adaptés aux catégories visuelles. Nous avons évalué cette adaptation en calculant $p(C|w)$, qui est la probabilité d'avoir une classe C lorsque le mot w est observé. Ces valeurs sont représentées sous forme d'histogrammes, pour chaque classe, pour tous les mots visuels. La figure 3.4 montre l'histogramme correspondant à la première catégorie des oiseaux, des résultats similaires ont été obtenus pour les autres classes. Nous observons que notre modèle a été capable de créer plus de 20 mots pour lesquels $p(C|w) > 0.9$, qui sont donc très spécifiques à la classe, alors que la quantification par les k-moyennes n'a créé qu'un seul mot aussi discriminant.

En guise d'illustration, la figure 3.4 montre également 5 des mots les plus discriminants par topic, pour les bases d'oiseaux et de papillons. Ces mots appartiennent aux animaux, et non au fond, de plus ils se focalisent sur les parties discriminantes, comme les yeux pour les hiboux, le cou pour les canards, etc. Nous pouvons voir ainsi la capacité de notre modèle à capturer les informations spécifiques aux classes.

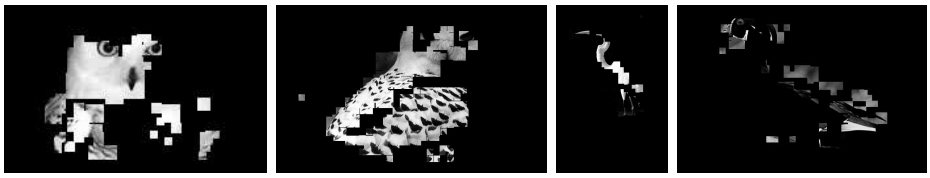


FIG. 3.5 – *Localisation des 5 mots les plus discriminants par image, dans des images de tests.*

Conclusion

Dans ce chapitre, nous avons proposé de créer des vocabulaires visuels adaptés à la classification d'objets. Le cœur de la méthode réside dans le fait que la création du vocabulaire fait partie intégrante du processus d'apprentissage. Il a été montré expérimentalement, et sur différentes bases, que ce modèle surpasse les méthodes traditionnelles, où le vocabulaire est créé séparément. Le nombre de mots utilisés pour obtenir de bonnes performances est plus petit que pour les approches par sac-de-mots traditionnelles. Cela est dû à la capacité de notre méthode à mieux quantifier l'espace des descripteurs que les méthodes de clustering standards. Les mots sont plus adaptés à la tâche et plus concentrés sur les informations de classe discriminantes. Nous obtenons donc une représentation très compacte et efficace. L'utilisation d'informations de classe lors de la création d'un vocabulaire visuel est une idée très prometteuse.

Une autre conclusion de ce travail est que l'approche par sac-de-mots donne de meilleurs résultats que le classifieur basé sur les topics, particulièrement lorsqu'un grand nombre de données d'apprentissage est disponible.

Cependant, comme toutes les observations (descripteurs visuels) sont directement utilisées pour apprendre notre modèle, son estimation est beaucoup plus coûteuse en temps que le modèle LDA standard.

Enfin, notons que les mots visuels ainsi produits sont particulièrement intéressants pour localiser les objets présents dans les images. La figure 3.5 localise les 5 mots les plus discriminants par classe dans les images proposées. Nous voyons que même s'il est encore difficile de localiser précisément les objets, ces informations sont suffisantes pour reconnaître leur catégorie.

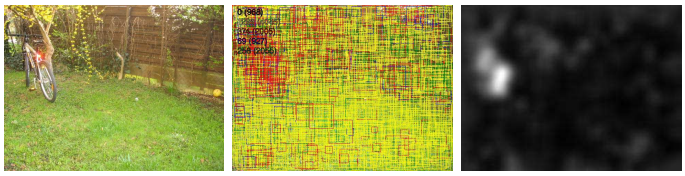
Ces mots les plus discriminants peuvent permettre de localiser les objets dans les images. Nous verrons dans les chapitres 5 et 6 comment combiner le modèle sac-de-mots avec des informations spatiales pour segmenter les objets.

Utilisation de forêts aléatoires pour la construction de cartes de saillance

4

Sommaire

Les forêts aléatoires constituent un moyen rapide et efficace de prédire la classe d'un patch extrait individuellement dans une image. Ces ensembles d'arbres peuvent être vus selon deux angles différents. Tout d'abord, les feuilles d'un arbre réalisent une quantification de l'espace des descripteurs, remplaçant ainsi un vocabulaire visuel. De plus la classification individuelle des patchs peut conduire à des hypothèses quant à la position des objets ; la carte de saillance ainsi obtenue peut être utilisée pour échantillonner les régions de l'image contenant probablement des objets et choisir plus efficacement les prochains descripteurs à extraire dans l'image.



4.1 Introduction

Dans les deux chapitres précédents, nous nous sommes intéressés à l'étude d'algorithmes de construction de vocabulaires visuels. Le chapitre 2 s'intéressait à l'extraction dense de descripteurs et à leur quantification en utilisant une combinaison d'échantillonnage biaisé et d'algorithme de clustering. Le chapitre 3 s'est attaché à rendre le vocabulaire discriminant, en le spécialisant aux classes à reconnaître. Dans ce cas, nous avons vu que ce travail, très prometteur en terme des résultats obtenus, s'était fait au prix d'une complexité accrue. Dans ce chapitre nous allons explorer une autre piste pour la construction de représentations discriminantes des images, capable de travailler avec des descripteurs denses. Comme la plupart des méthodes de classification, et comme pour les deux chapitres précédents, nous nous baserons sur des primitives locales.

La première contribution de ce chapitre est l'utilisation d'arbres de décision aléatoires afin de créer une quantification discriminante de l'espace des descripteurs.

Dans ce chapitre, nous nous intéressons également à la question de la saillance visuelle. Nous partons du principe que si nous disposons d'hypothèses sur la présence des objets dans une image, nous allons pouvoir restreindre son échantillonnage, c'est-à-dire le calcul de descripteurs locaux, aux régions « intéressantes ».

Des modèles de saillance visuelle biologiquement plausibles ont été largement étudiés [32] mais ont en général peu de connexion avec les méthodes de reconnaissance complexes.

Dans le domaine de la recherche visuelle (ou attention visuelle), les travaux publiés montrant l'utilité de telles méthodes pour des problèmes de reconnaissance complexes sont encore plus rares. La *recherche visuelle* ou *attention visuelle* est le processus de sélection d'informations, basé sur la saillance (processus ascendant ou *bottom-up*) et sur les connaissances a priori sur les objets (processus descendant ou *top-down*) [2, 103]. Pour la vision humaine, il semble que les mécanismes d'attention visuelle et de fixation oculaire constituent un élément indispensable [101]. Cependant, les systèmes de vision biologiquement inspirés mélangeant processus ascendant et descendant n'ont jamais été capables de faire mieux en terme d'erreur de classification que les méthodes purement ascendantes, quand il s'agit de problèmes de reconnaissance complexes.

De plus il a été montré que le concept d'attention visuelle dépendait fortement de la tâche à résoudre dans le cas du système de vision humain. En effet, Yarbus [101] a montré que les fixations oculaires sur une scène dépendent de la tâche demandée (voir l'illustration de l'expérience dans l'annexe A.1).

Ainsi nous proposons un système où la saillance est définie comme un ensemble d'attributs qui distingue au mieux un concept (la catégorie d'objet à reconnaître dans notre cas) des autres concepts. Cet ensemble d'attributs dépend du concept, et il devrait être appris pour chaque concept donné.

La deuxième contribution de ce chapitre est une combinaison de processus ascendant et descendant de façon à ce que les erreurs de classification soient moins nombreuses que pour un processus ascendant seul. Dans notre approche le processus ascendant utilise les informations de saillance telles que nous venons de les définir. Le processus descendant est basé sur une es-

timation en ligne[1] d'une fonction de densité de probabilité de présence d'une partie d'objet en fonction de la position et de l'échelle donnée dans l'image.

Ainsi, la méthode de classification que nous proposons dans ce chapitre combine une carte de saillance avec un classifieur de parties d'objet. La connaissance a priori est stockée par le classifieur et utilisée pour construire la carte de saillance qui fournit ainsi de l'information sur la classe des objets.

Nous avons évalué l'efficacité de la méthode proposée en mesurant la quantité d'information extraite d'une image, par rapport aux performances de classification.

Après une description des méthodes dont l'approche est similaire à la notre, en section 4.2, la section 4.3 présente un algorithme basé sur des arbres, utilisé pour la classification. La section 4.4 explique ensuite comment des informations de saillance spécifiques aux objets sont utilisées pour construire une carte d'attention visuelle, et comment celle-ci est combinée au classifieur. Enfin, nous présentons nos expériences dans la section 4.5.

4.2 État de l'art

4.2.1 Catégorisation d'objets

Nous avons déjà longuement parlé des différentes méthodes utilisées pour la catégorisation d'images. Dans le chapitre 2, nous avons justifié l'utilisation de descripteurs extraits de manière dense, par le manque de répétabilité des points d'intérêt à l'échelle de la catégorie. Pour gérer la quantité d'information résultant de cette extraction dense, nous avons proposé une méthode de clustering, permettant de trouver les clusters dans ces conditions particulières de représentation de l'espace des descripteurs.

Ici nous étudions une deuxième classe de méthode qui modélise directement les frontières entre les clusters de mots visuels. Ces frontières peuvent être apprises par l'intermédiaire d'arbres de décisions.

Les arbres de décisions ont été utilisés comme classifieur de patchs par Marée *et al*, [55]. Dans leur méthode, la catégorisation d'objets est faite en classifiant un ensemble de patchs choisis aléatoirement. Ces patchs sont classifiés indépendamment, c'est-à-dire que chacun détermine s'il peut être une partie locale d'un objet appartenant à une des catégories considérées. Ensuite les différents patchs votent, et la catégorie la plus probable gagne.

D'autre part, aucune des méthodes citées, que cela soit dans le cadre de la première famille de stratégies (exposée chapitre 2) ou dans le cadre de la deuxième (ce chapitre), n'utilisent de recherche visuelle : les primitives d'images sont extraites une fois pour toute au début du processus, et la façon dont ces primitives sont choisies provient d'un mécanisme purement ascendant.

4.2.2 Saillance visuelle

Même si la recherche visuelle n'est pas utilisée couramment pour la catégorisation d'objets, elle fait partie intégrante des systèmes de visions complexes comme le système de vision humain.

[1] mise à jour après chaque patch observé, lors de la classification d'une image

Nous pouvons distinguer différents types de primitives saillantes utilisées dans la littérature. Premièrement celles basées sur des détecteurs de points d'intérêt [29, 54], qui extraient des régions texturées de l'image. Ces détecteurs peuvent également être définis pour être invariants aux changements d'échelle, aux rotations, et même aux transformations affines [59]. Comme ils permettent de résumer les images par un petit nombre d'informations locales, ils rendent plus léger la suite du processus de classification. C'est pourquoi, les algorithmes de catégorisation d'objets [13, 19, 51, 52] utilisent très couramment ces détecteurs au début du processus. Dans [77], la saillance est définie comme la somme des valeurs absolues des décompositions locales en ondelettes. Kadir*et al* [37] proposent une mesure basée sur l'entropie de la distribution des intensités locales.

Les points d'intérêt sont très efficaces pour détecter les structures locales similaires de façon répétable, mais ils ne peuvent détecter de façon certaine les structures pertinentes. Il est clair que saillance ne signifie pas forcément complexité, ni répétabilité, mais que celle-ci est plus liée à la capacité de déterminer quelle information distingue le mieux un concept (ou un objet) d'autres concepts (ou objets) possibles. Ainsi, Walker *et al* [94] définissent les primitives saillantes comme celles qui ont une probabilité faible d'être confondues avec une autre primitive, et Vidal-Naquet *et al* [92] proposent de sélectionner les fragments d'images qui maximisent l'information mutuelle entre le fragment et la classe d'objet. Fritz *et al* [21] décrivent un système dans lequel des régions discriminantes sont produites par une mesure d'entropie conditionnelle. Il a été également signalé par Serre *et al* [78] que les primitives faiblement spécifiques ne sont pas aussi efficaces que les primitives apprises spécifiquement, pour la reconnaissance d'objets. La saillance peut également être définie par un critère de rareté. Dans [27] la saillance est définie comme étant inversement proportionnelle à la densité dans l'espace de descripteurs.

Toutes ces définitions sont dérivées à partir de critères définis empiriquement, et ne sont pas biologiquement plausibles. Les modèles de vision biologiquement inspirés sont attrayants de part leurs racines. En effet, les systèmes biologiques sont les seuls systèmes connus qui fonctionnent parfaitement. Des systèmes comme ceux proposés par Itti *et al* [32] ont donné des comportements intéressants, mais, comme montré par Gao et Vasconcelos [22], le manque d'un critère optimal clairement définit pour la saillance constitue une limitation importante de ces méthodes. L'approche proposée par Walther *et al* [95] se doit d'être soulignée, ils ont montré que le modèle de Koch et Ullman [38], qui est un modèle d'attention basé sur la saillance, peut améliorer les performances en reconnaissance d'objets, puisqu'elle prouve expérimentalement la validité des modèles biologiquement inspirés basés sur la saillance.

4.2.3 Recherche visuelle

La saillance visuelle et la recherche visuelle sont deux concepts différents mais fortement liés. L'exploration d'images à travers une séquence de points de fixation est supposée rendre la tâche d'interprétation plus légère en utilisant des informations selon un procédé descendant. Dans l'introduction, nous avons avons déjà mentionné les travaux de Yarbus [101], qui montrent que

les fixations dépendent de la tâche à résoudre. Encore une fois, l'objet ou le concept que l'on cherche devrait influer sur le critère de recherche. Malgré la pertinence de ce concept, l'attention visuelle n'est utilisée dans aucun des système de catégorisation d'objets présentés précédemment.

Cependant, des systèmes basés sur l'attention visuelle ont déjà été proposés par le passé. Navalpakkam et Itti [63] ont proposé une recherche visuelle biaisée par un processus descendant. Dans le cadre de la détection d'objets simples, ce biais permet de réduire par deux le nombre de fixations. Dans [5], Bonaitu et Itti montrent qu'un élagage rapide des espaces de recherche pour la reconnaissance, améliore la rapidité de la reconnaissance d'objets. Avraham *et al*[2] utilisent une carte de priorité dynamique pour la catégorisation d'une liste de régions.

4.2.4 Positionnement de notre approche

L'approche proposée ici peut être considérée comme une extension de la méthode proposée par Marée *et al*, dans [55]. Dans ce travail, les patchs d'images sont sélectionnés aléatoirement et classifiées comme appartenant à une catégorie d'objets ou au fond. Nous proposons de biaiser cette sélection aléatoire et d'utiliser une carte de saillance construite en ligne. Cette carte sera ainsi capable de s'adapter à l'objet que l'on souhaite reconnaître. Nous améliorons également la classification basée sur des arbres en remplaçant le simple vote des feuilles par une combinaison des votes à l'aide d'un classifieur SVM.

4.3 Classification d'images à l'aide d'arbres

Le cadre de travail que nous avons utilisé pour la classification d'images est basé sur les travaux de Marée *et al* [55]. Leur méthode comporte les étapes suivantes :

- ▷ Tout d'abord des patchs sont échantillonnés aléatoirement dans les images d'apprentissage.
- ▷ Ensuite des arbres de décision aléatoires sont construits comme des classifieurs sur ces patchs.
- ▷ Sur les images de tests, les patchs sont également échantillonnés aléatoirement, et chaque patch est classifié par les arbres de décision.

Dans cette section, nous allons d'abord décrire le processus d'échantillonnage des patchs et d'encodage de l'information, ensuite nous décrirons les « arbres de décision extrêmement aléatoires » qui sont utilisés pour la classification de chaque patch, puis nous introduirons le classifieur SVM qui remplace le vote final et qui apprend l'importance des feuilles dans les arbres de décision.

4.3.1 Extraction des primitives

Afin d'obtenir des primitives à partir des images, nous n'utilisons pas un détecteur de points d'intérêt qui se limite à certaines régions texturées de l'image, mais à la place les patchs sont échantillonnés à des positions et des échelles aléatoires.

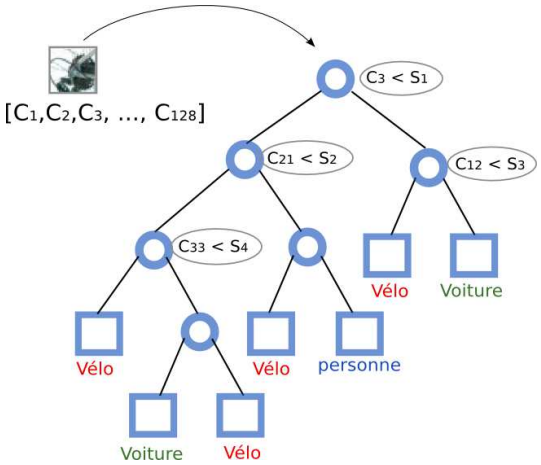


FIG. 4.1 – *Mécanisme de classification par un arbre de décision. Pour chaque nœud, un attribut C_i est comparé à un seuil S_x. Chaque feuille prédit une classe.*

Chaque patch est représenté par un descripteur. Nous en avons essayé plusieurs, avec pour conclusion que le meilleur choix dépend de la base d'images. Parmi ceux ci, citons le descripteur couleur qui donne directement les valeurs de couleur dans l'espace HSL du patch redimensionné en 16x16 pixels, et retourne un vecteur de dimension 768 (16x16x3). Le descripteur d'ondelettes couleur calcule une transformation en ondelettes de Harr [85] pour chaque canal de couleur et retourne également un vecteur de dimension 768. Le dernier descripteur est le populaire descripteur SIFT [54], qui retourne un histogramme 4 par 4 de 8 orientations (vecteur de dimension 128). Dans les expériences, nous préciserons à chaque fois le descripteur utilisé.

4.3.2 Arbres de décision

Après l'extraction des primitives, nous devons choisir une méthode qui peut faire face aux centaines de milliers de primitives dans un espace de grande dimension. De plus, la méthode doit être capable de classifier les patchs individuellement afin de permettre la construction de la carte de saillance, mais aussi d'utiliser l'image dans sa globalité pour la tâche de catégorisation. Dans ce chapitre, nous avons choisi de considérer les arbres extrêmement aléatoires (EXTremely RAndomize Trees, ou EXTRA-Tree) de Marée *et al* [55]. En les combinant avec un classifieur SVM, nous obtenons des résultats intéressants. Cela nous donne un système capable de classifier les patchs individuellement mais aussi l'image entière. Les arbres sont des classifieurs de descripteurs qui prédisent ainsi une classe pour chaque patch. Mais les feuilles peuvent être considérées comme une quantification de l'espace des descripteurs, qui remplace avantageusement un vocabulaire visuel.

Les arbres aléatoires sont très semblables aux arbres de décision standards. La classification d'un descripteur à n valeurs (attributs) consiste en un parcours de l'arbre, représenté figure 4.1. Pour chaque nœud rencontré, une comparaison est faite entre un attribut et un seuil. En fonction du résultat de cette comparaison, le parcours de l'arbre continue soit par le fils droit, soit par le fils gauche de ce nœud. Un test est donc défini par deux attributs : une coordonnée et un seuil. Ce parcours se termine lorsqu'une

feuille est rencontrée. Chaque feuille est associée à un label de classe, ainsi une primitive est classifiée par l'arbre en lui affectant le label de la feuille atteinte. La seule différence entre les arbres de décision standards et les arbres aléatoires, est que la construction de l'arbre met en œuvre des décisions aléatoires. Ceci accélère énormément le processus d'apprentissage par rapport à un arbre standard, en particulier pour de grandes dimensions, car ce dernier cherche le meilleur découpage possible de chaque nœud de façon exhaustive.

Chaque classifieur de nœud est choisi comme le meilleur parmi un ensemble de couples attribut/seuil possibles, générés aléatoirement. Une justification des arbres aléatoires est disponible dans les travaux de Breiman *et al* [8]. Le choix se fait selon un critère qui correspond au gain en log-vraisemblance produit par le test, qui est proportionnel à l'information mutuelle $I_{S,C}$. La variable S peut prendre deux valeurs selon le résultat du test, et la variable C prend comme valeur un des labels de catégories. Le critère s'écrit :

$$I_{S,C} = H(S) + H(C) - H(S,C) \tag{4.1}$$

où $H(S)$ est l'entropie du découpage en termes de population, $H(C)$ en termes de classe et $H(S,C)$ l'entropie conjointe. Il s'agit donc de répartir les patchs de telle manière que ceux provenant d'une même catégorie occupent les mêmes feuilles, tout en obtenant des découpages équilibrés en termes de population. Une description complète de l'algorithme peut être trouvée dans la thèse de Geurts [25].

L'accélération de la procédure d'apprentissage par l'introduction de l'aléatoire a certains inconvénients. Comparés aux arbres de décision standards, les arbres aléatoires sont plus larges et ont une variance plus grande. Le premier point n'est pas réellement un problème puisque les arbres sont très rapides pour la classification. Par contre, la forte variance décroît les performances de classification. Dans son manuscrit de thèse [25], Geurts donne un bon aperçu des méthodes qui diminuent la variance des arbres de décision aléatoires. Il y a principalement deux possibilités pour réduire la variance. La première est de faire de l'élagage (*pruning*), l'autre est de construire plusieurs arbres et d'utiliser l'ensemble des arbres (forêt) pour la classification. Les expériences montrent que lorsque le nombre d'arbres augmente, l'erreur de classification s'en trouve considérablement réduite, mais la complexité augmente également. Nous utiliserons dans nos expériences ces deux techniques de réduction de la variance.

4.3.3 Apprentissage de l'importance des feuilles

Afin de classifier une image, des patchs sont une fois encore échantillonnés aléatoirement dans l'image, et chaque patch est ensuite classifié à l'aide de l'ensemble d'arbres de décision. Les votes des arbres sont comptés, et l'image est affectée à la classe ayant le plus de votes.

Une propriété des arbres de décision est de s'adapter parfaitement aux données. Ils généralisent donc assez mal, mais certaines feuilles sont plus fiables que d'autres. Utilisées directement, les feuilles de l'arbre qui ne produisent pas de bons résultats de classification influencent le vote autant que les feuilles importantes de l'arbre. C'est pourquoi nous proposons une méthode pour apprendre l'importance des décisions prises par les feuilles. Pour

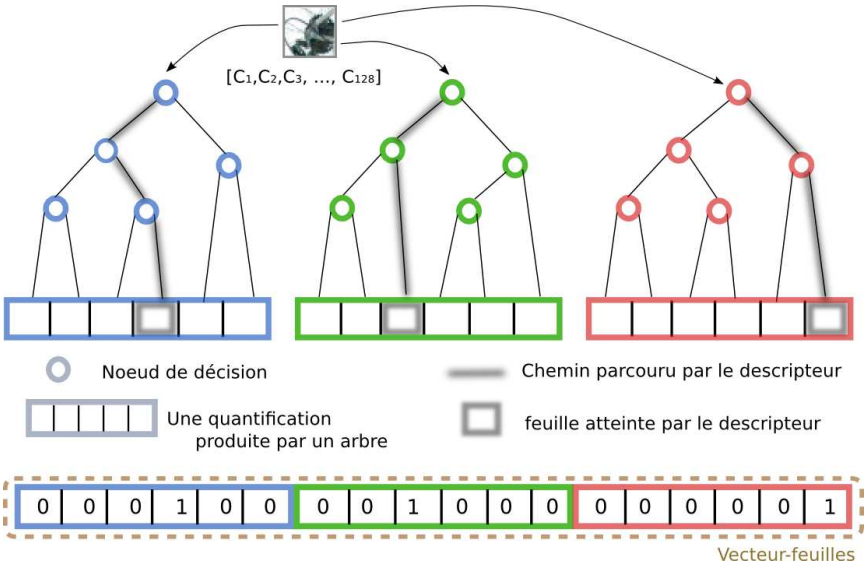


FIG. 4.2 – *Mécanisme de création du vecteur de représentation par image (vecteur-feuilles) en utilisant les arbres de décision. L'exemple ne considère qu'un seul descripteur.*

cela, nous nous appuyons sur une méthode de classification qui généralise mieux et a la capacité d'ignorer les données moins importantes, le classifieur SVM.

Pour apprendre l'importance des feuilles, chaque image est représentée par un vecteur binaire dont la taille est égale au nombre total de feuilles. Au début du processus de classification d'une image, toutes les entrées du vecteur sont à 0. Lors de la classification d'un patch de cette image, toutes les feuilles atteintes sont considérées, et les entrées correspondantes dans le vecteur sont mises à 1. La figure 4.2 illustre le procédé. Notons qu'une entrée choisie plusieurs fois au cours de la classification d'une image reste à 1. Une fois tous les patchs classifiés, les entrées à 1 dans le vecteur représentent toutes les feuilles qui sont responsables de la classification d'une image.

Ensuite, un classifieur SVM est utilisé pour apprendre l'importance des feuilles. Pour cela, le procédé de construction de vecteurs précédemment décrit est appliqué aux images d'apprentissage. Cela nous donne autant de vecteur-feuilles que d'images d'apprentissage. Ces vecteurs servent à entraîner un classifieur SVM linéaire. Afin de donner plus d'information au classifieur SVM, nous augmentons la dimension du vecteur-feuilles d'une unité afin d'ajouter le label de catégorie retourné par la classification par votes.

Lorsqu'une image est classifiée, les patchs sont échantillonnés comme décrit précédemment, puis classifiés à l'aide des arbres de décisions, et tous ces patchs sont utilisés pour construire un vecteur-feuilles qui est classifié par le classifieur SVM. La sortie du classifieur SVM est notre décision finale sur la catégorie de l'image. Les expériences montrent que cette méthode améliore de façon significative les performances de classification, par rapport au cas du vote simple.

L'entraînement d'un classifieur SVM linéaire revient à apprendre l'importance des feuilles puisque le problème d'optimisation qui est résolu pendant l'apprentissage calcule l'équation d'un hyperplan qui sépare au mieux les classes, et ainsi pondère les dimensions dans l'espace du vecteur-feuilles.

Notons que cette façon d'utiliser les arbres aléatoires revient à considérer que chaque arbre est une quantification de l'espace des descripteurs et donc simule le rôle d'un vocabulaire visuel. Ainsi le classifieur SVM est entraîné sur un vecteur qui peut être vu comme la concaténation d'histogrammes d'occurrences de mots visuels binarisés (chaque entrée de l'histogramme vaut 0 ou 1). La façon dont sont créés les arbres les rend intrinsèquement discriminants, ce qui place ce travail dans la continuité du chapitre précédent (voir discussion dans la section 4.5.4).

Dans cette section, nous avons décrit la méthode de classification que nous avons utilisée et proposé des extensions par rapport au travail de Marée *et al* qui améliorent de façon significative les performances, comme le prouvent nos expériences. Dans la section suivante, nous allons décrire une deuxième extension de la méthode qui utilise des cartes de saillance pour guider le processus d'échantillonnage des descripteurs.

4.4 Construction des cartes de saillance

Dans cette section, nous introduisons des cartes de saillance qui sont adaptées à nos besoins. Tout d'abord, nous définissons la saillance telle qu'elle sera utilisée. Puis nous expliquons comment biaiser l'échantillonnage aléatoire des patchs à l'aide de la carte de saillance, afin d'améliorer les performances de classification. Enfin, nous montrons que ces cartes de saillance peuvent être construites de façon efficace : en ligne pendant la classification.

4.4.1 Lien avec la probabilité de trouver les objets

Une carte de saillance décrit les zones d'une image susceptibles de contenir des informations pertinentes pour la tâche courante. Ces zones sont celles où se trouvent des primitives considérées comme rares ou informatives selon le critère de saillance utilisé (voir section 4.2). Ici, nous cherchons à reconnaître des objets : les régions susceptibles de contenir des informations intéressantes, c'est-à-dire les zones à forte saillance, correspondent donc aux endroits dans l'image où ces objets sont le plus souvent situés. Les régions de saillance plus faible sont, elles, associées au fond (tout ce qui n'est pas objet).

Comme montré dans la section 4.2, la plupart des approches sont jusque là basées sur des processus ascendants, indépendants de la tâche. Nous plaidons qu'il est utile pour un système de classification d'avoir une connaissance a priori sur l'endroit où il peut trouver des primitives discriminantes pour identifier les objets.

Le classifieur que nous avons introduit extrait des patchs à des positions aléatoires avec une taille aléatoire et les classifie individuellement. Pour améliorer les performances du procédé de classification, nous pouvons utiliser l'information a priori sur la position des objets pour échantillonner plus intensivement ces régions. L'échantillonnage aléatoire d'un patch correspond maintenant à choisir un point dans un espace 3D (position x échelle) en fonction d'une densité de probabilité qui représente la saillance.

Dans cette section, nous introduisons donc la saillance comme une densité de probabilité 3D, pour laquelle un point est défini comme saillant si le classifieur classifie le patch correspondant comme un objet. Une carte de

saillance contient des informations qui peuvent être exprimées comme la probabilité estimée $\hat{p}(O|X)$ de trouver un objet O à la position X, où $X = (x, y, s)$ indique la position (x, y) et l'échelle s. Ce type d'approche a déjà été mentionnée par [102]. La carte de saillance, qui représente la densité de probabilité, peut donc être utilisée comme une connaissance a priori pour biaiser l'échantillonnage.

4.4.2 Classification active d'images avec cartes de saillance

Jusque là, nous avons introduit une définition de la carte de saillance et expliqué comment elle peut être utilisée pour biaiser le processus d'échantillonnage. Maintenant nous montrons comment ces cartes peuvent être construites en ligne et utilisées en même temps que le procédé de classification.

Notre but est de guider le processus d'échantillonnage de façon à ce que suffisamment de patchs soient échantillonnés sur l'objet. Pour l'atteindre, nous avons introduit une densité de probabilité pour l'échantillonnage qui combine la probabilité pour un objet d'être présent à une position et une échelle donnée, avec la probabilité d'avoir déjà exploré une région donnée de l'espace de l'image. Les probabilités suivantes sont considérées :

- ▷ $\hat{p}(O|X, Z_{1:n-1})$ est la probabilité d'avoir un objet à la position X, sachant les $n-1$ dernières mesures Z. Il s'agit de la carte de saillance dont nous avons parlé précédemment.
- ▷ $p(E|X, S_{1:n-1})$ exprime le degré de nécessité pour l'exploration sachant les $n-1$ derniers échantillons S. Elle modélise l'information sur la position où les patchs ont déjà été échantillonnés à l'aide d'une densité de probabilité.

De façon à échantillonner les patchs, nous combinons ces deux informations pour obtenir une seule densité de probabilité. En multipliant les deux densités de probabilité, la densité de probabilité d'échantillonnage du prochain patch est donnée par :

$$p(S_n = X) = \frac{\hat{p}(O|X, Z_{1:n-1}) \cdot p(E|X, S_{1:n-1})}{\sum_X \hat{p}(O|X, Z_{1:n-1}) \cdot p(E|X, S_{1:n-1})} \tag{4.2}$$

Les patchs qui ont une probabilité forte d'être sur un objet et qui appartiennent à une région de l'image qui n'a pas encore été explorée vont être sélectionnés avec une forte probabilité.

Pour estimer $\hat{p}(O|X)$ nous initialisons notre densité de probabilité discrète avec une distribution uniforme. Pour chaque patch échantillonné, nous ajustons cette distribution selon le résultat de classification, avant d'échantillonner le patch suivant. Si un patch extrait au point (x, y, s) est classifié comme un objet, la densité de probabilité de son voisinage $(x \pm c, y \pm c, s \pm c)$ est augmentée d'une valeur constante et la densité de probabilité est ensuite normalisée. Si ce patch est classifié comme un élément du fond, la densité de probabilité est diminuée dans le même voisinage. Le rayon du voisinage c qui a été utilisé dans nos tests est de 5% de la taille de l'image. Incrémenter la carte de saillance d'une valeur constante dans une région cubique peut sembler être une solution simpliste à première vue, mais lorsque l'opération est réalisée plusieurs fois, les densités obtenues sont relativement lisses, comme le montre les résultats (figure 4.4).

Le besoin en exploration $p(E|X, S_{1:n-1})$ est estimé de façon très similaire à $\hat{p}(O|X)$ à la différence qu'il est indépendant des résultats de la classification du patch. Il est également initialisé à l'aide d'une distribution uniforme. Puis chaque fois que cette distribution est échantillonnée pour obtenir un patch, le besoin en exploration pour cette région est réduit. Nous fixons alors la probabilité pour ce point à 0 et réduisons la valeur de ses voisins de la même façon que nous avons augmenté/diminué la densité $\hat{p}(O|X)$. Cependant, le rayon utilisé est plus petit (3%).

Nous avons présenté dans cette section une définition de la saillance basée sur le degré de discrimination des primitives. Nous avons décrit comment ces cartes de saillance peuvent être construites en ligne pendant le processus de classification. Dans la section suivante, nous allons montrer des résultats produits par les méthodes que nous avons proposées.

4.5 Expériences

Nos expériences visent à mesurer l'impact de la combinaison des mécanismes ascendant et descendant sur les performances de catégorisation visuelle. La tâche considérée vise à déterminer la catégorie de chaque image, selon les objets qu'elle contient.

Les expériences ont été réalisées sur différentes bases d'images : la base Tu-Graz02 (section 1.3.2), la base pascal VOC 2005 (section 1.3.5) et la base des chevaux (section 4.5.3). Comme cette méthode inclue une grande quantité d'aléatoire, les résultats diffèrent d'une exécution à l'autre. Nous avons effectué les tests plusieurs fois, et présentons les résultats sous forme de moyenne et de variance. Nous avons mesuré les performances avec des courbes ROC et leur EER associé. Cette mesure de performance est définie dans la section 2.4.1.

4.5.1 Expériences sur la base TU-Graz 02

La base TU-Graz 02 contient 4 catégories : des images de vélos, de voitures, de personnes et des images de fond.

Pour les tests, nous avons utilisé les 300 images segmentées de chaque catégorie (voir section 1.3.2 pour plus de détails sur la base) découpées en un ensemble d'apprentissage et un ensemble de test. Différents tests ont été menés selon deux configurations. Dans la configuration #1, nous n'avons pas utilisé les masques de segmentation et entraîné notre méthode sur toute l'image ce qui est aussi la configuration utilisé par [68]. Dans la configuration #2, nous avons utilisé les masques de segmentation et appris les modèles d'objets uniquement sur les objets eux-mêmes. Dans les deux configurations, seule une des catégories d'objets a été testée contre les images de fond.

Nous avons décidé de faire des tests intensifs sur deux des catégories : les vélos et les voitures. Le descripteur utilisé est une transformation en ondelettes dans l'espace de couleur HSL.

Les paramètres de notre méthode sont étudiés sur cette base ainsi que sur la base des chevaux. Les résultats sont présentés sur la figure 4.3.

En augmentant le nombre de patchs échantillonnés dans les images d'apprentissage, le taux de classification augmente également. Il est particulièrement intéressant de regarder les valeurs numériques : avec 100 patchs par

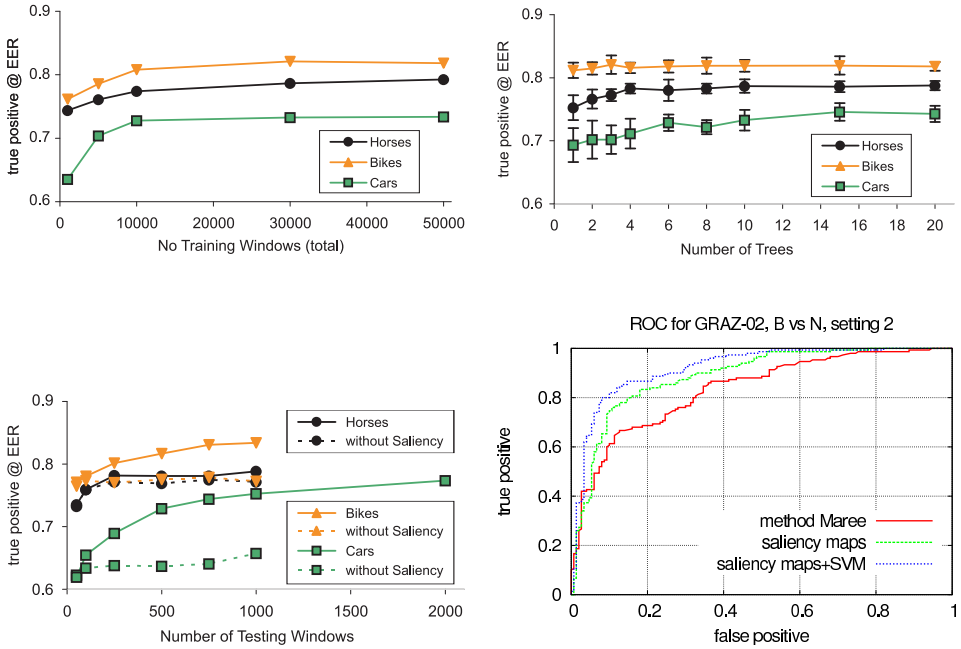


FIG. 4.3 – *Évaluation des paramètres sur différentes bases. Les boîtes indiquent l'écart type.*

image d'apprentissage, les résultats sont déjà stabilisés et ne sont plus vraiment améliorés.

Quant au nombre d'arbres de décision, nous pouvons observer que lorsque le nombre d'arbres augmente, la variance s'en trouve réduite. Les deux paramètres ont malheureusement en commun la propriété d'accroître le temps de calcul.

Le troisième graphe montre comment les taux de classification augmentent lorsque le nombre de patchs extraits par image sur les images de test augmente. Les pointillées indiquent ici les tests sans carte de saillance. Il est aisé de noter que les cartes de saillance améliorent les résultats lorsque un nombre important de patchs est échantillonné. Cela est dû au fait que ces cartes sont construites en ligne et ont donc besoin d'un peu de temps pour accumuler suffisamment de connaissance pour influer sur les résultats.

Aucun des résultats présentés sur ces 3 graphes n'utilise de classifieur SVM qui améliore encore les résultats. Cette amélioration est mise en évidence sur le dernier graphe qui montre les courbes ROC obtenues pour des tests réalisés sur la base TU-Graz02, selon la configuration #2. Ce graphe compare les résultats obtenues par notre méthode (avec et sans le classifieur SVM) avec ceux obtenus avec la méthode proposée par Marée *et al* [55], sur le même ensemble de test. Même si nous utilisons seulement 5 arbres, et 30000 patchs d'apprentissage (100 par image), comparé aux 10 arbres et 100000 descripteurs de Marée, nous obtenons de meilleurs résultats.

La performance globale sur les images de tests est donnée par la table 4.1. Ces résultats peuvent être encore améliorés en utilisant plus de 500 patchs pour les images de tests. Cependant, cela augmente les temps de calcul. Pour obtenir les résultats présentés dans le graphe, notre méthode n'a besoin que d'une à deux secondes par image.

Pour illustrer les résultats de ces tests, nous avons sélectionné des images de la base de test, classifiées avec la configuration #2. La figure 4.4 contient

	configuration #1		configuration #2	
	B vs. N	C vs. N	B vs. N	C vs. N
Résultats obtenus par Opelt *et al* [68]	0.765	0.707	-	-
Méthode de Marée *et al*	-	-	0.736	0.621
avec cartes de saillance	0.75	0.663	0.821	0.728
avec cartes de saillance + SVM	0.844	0.799	0.841	0.798

TAB. 4.1 – *Taux de classification à l'EER, pour la base TU-Graz02. B dénote les vélos, C les voitures, et N le fond.*

ces images ainsi que la carte de saillance associée à chacune d'entre elles. Les images représentant les cartes de saillance, créées au départ dans un espace à 3 dimensions (position et échelle) sont obtenues par une projection en 2 dimensions.

Pour illustrer les capacités de généralisation de notre méthode, nous avons entraîné le modèle sur les 3 catégories de la base TU-Graz, et testé sur des images additionnelles, récupérées sur internet (c'est ce que nous appellerons la configuration #3). Ce test inclut l'utilisation des masques de segmentation pour l'apprentissage. La figure 4.5 montre des résultats pour cette configuration de test. Sur l'image la plus à gauche, la méthode sélectionne comme régions saillantes les vélos et même les camions, classe sur laquelle nous n'avons réalisé aucun entraînement. La méthode généralise même dans les images qui ne contiennent aucun des objets sur lesquels elle a été entraînée, et identifie correctement les fonds qui ont été appris. Pour obtenir ces résultats de la configuration #3, nous avons utilisé 60000 patchs d'apprentissage au total (100 par image).

4.5.2 Expériences sur la base Pascal VOC 2005

Pour comparer les performances de classification de notre méthode avec l'existant, nous avons fait des tests selon les directives données pour la compétition Pascal Challenge 2005 (introduite section 1.3.5). Les tests sont faits selon la configuration#1, c'est à dire qu'on ne connaît pas la position des objets dans les images. Seul 73 patchs par image ont été utilisés pour l'apprentissage. Ces patchs sont représentés par des descripteurs SIFT. La classification utilise 4 arbres, et 10000 patchs par image pour construire les vecteurs-feuilles. Ces vecteurs feuilles sont utilisés pour entraîner le classifieur SVM dans le cas des images d'apprentissage, et pour classifier les images de test. Les résultats sont présentés dans le tableau 4.2. Les valeurs d'EER obtenues en moyenne sont de 0.958 sur les motos, 0.901 sur les vélos, 0.94 sur les personnes et 0.96 sur les voitures. Ces résultats sont comparables aux meilleures méthodes qui ont participé au Pascal Challenge mais il faut noter que, contrairement à elles, notre méthode utilise moins d'information et nécessite un temps de calcul plus faible.

4.5.3 Expériences sur la base des chevaux

Cette base a été introduite dans [35] et contient 2 sortes d'images : avec chevaux et sans chevaux. Les images ont été choisies sur internet, et sont donc moins biaisées. Les chevaux peuvent être petits, sous différentes poses,

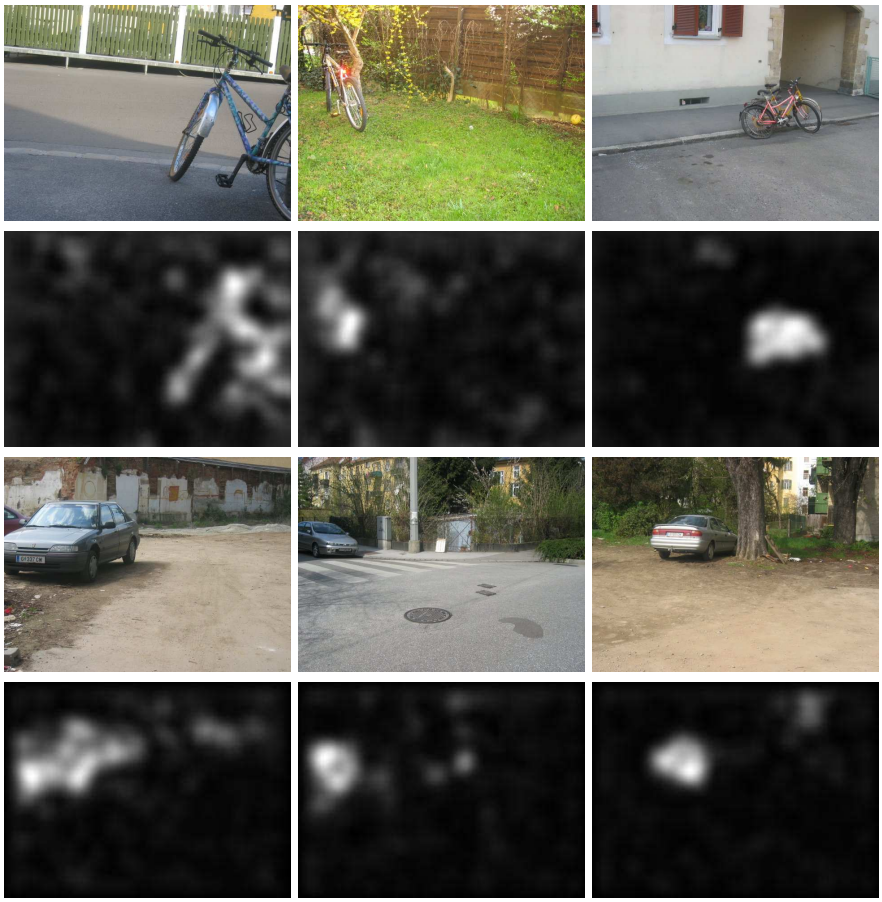


FIG. 4.4 – *Cartes de saillance obtenues pour la base TU-Graz02, pour la configuration #2. Chaque image est présentée avec sa carte de saillance associée.*

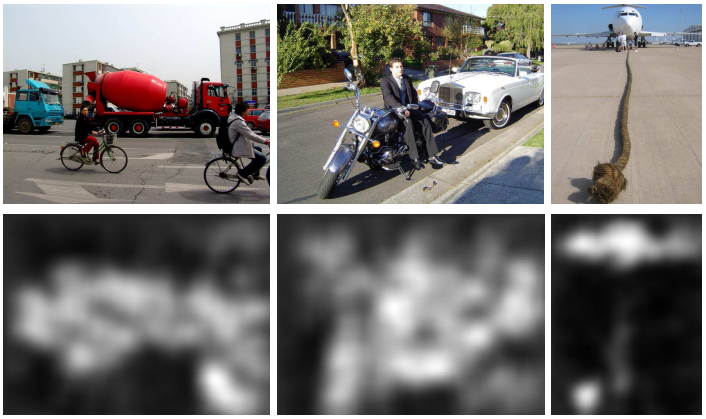


FIG. 4.5 – *Cartes de saillance obtenues pour de nouvelles images, trouvées sur internet, avec un entraînement sur les images TU-Graz02.*

	EER pour la classification de la base 'test 1'			
	motos	vélos	personnes	voitures
soumission INRIA_L (chap 2)	**0.977**	**0.930**	**0.901**	**0.938**
soumission INRIA_Z	0.964	0.930	0.917	0.937
vocabulaire discriminant (chap 3)	**0.97**	**0.895**	**0.912**	**0.927**
Méthode de Marée	0.864	0.808	0.774	0.871
avec la saillance (chap 4	0.890	0.737	0.770	0.836
avec la saillance + SVM	**0.958**	**0.901**	**0.940**	**0.960**

TAB. 4.2 – *Résultats de classification pour la compétition Pascal VOC 2005*

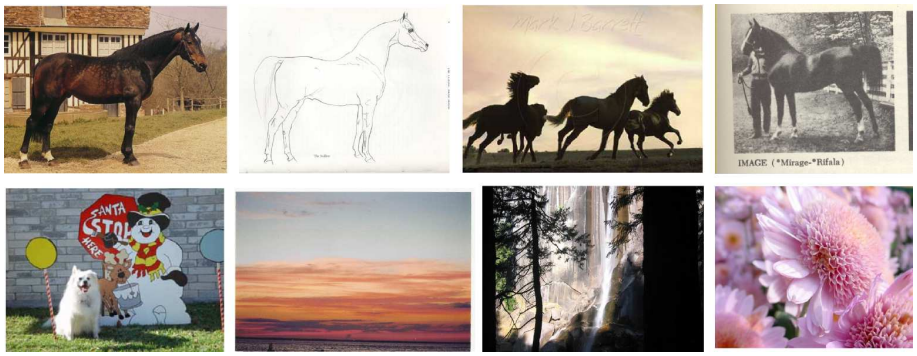


FIG. 4.6 – *Exemples d'images de la base des chevaux. La première ligne montre des chevaux, la deuxième montre des exemples négatifs.*

et peuvent être occultés. Parfois, il s'agit seulement d'un croquis. Des images sont disponibles sur la figure 4.6.

En utilisant des descripteurs SIFT, nous obtenons un taux de classification de 0.853 à l'EER, ce qui peut être considéré comme un bon résultat. Il ne peut être comparé avec d'autres méthodes, cette base étant utilisée en général pour des tâches de détection.

4.5.4 Comparaison avec les chapitres 2 et 3

Ce chapitre présente une nouvelle méthode de classification d'images basée sur des forêts aléatoires et un classifieur SVM. Les vecteurs-feuilles construits peuvent être vus comme des histogrammes d'occurrences de mots visuels où les mots de vocabulaires sont représentés ici par les feuilles des arbres. Sous cet angle de vue, nous pouvons comparer l'utilisation de ces forêts avec les deux méthodes de clustering proposées dans les chapitres précédents.

Dans le chapitre 2, le problème considéré est celui de la modélisation de la densité dans l'espace de représentation des descripteurs de patchs. Nous avons vu que cette modélisation est rendue difficile dans le cas d'un échantillonnage dense des images, car les données extraites sont déséquilibrées ce qui met en difficulté les méthodes de clustering classiques. Nous avons donc proposé une méthode qui tient compte de cette spécificité tout en étant très efficace pour l'apprentissage sur des millions de descripteurs. Dans ce chapitre, la difficulté est contournée puisqu'il ne s'agit pas de repérer les zones denses de l'espace de descripteurs mais de décider directement de frontières entre les régions de cet espace qui vont représenter les mots visuels. Ces frontières sont définies par les classifieurs unitaires de l'arbre. Une telle approche est discriminative.

Le chapitre 3 s'intéresse à la création de vocabulaires visuels discriminants et compacts. Le but est d'améliorer les performances de classification du vocabulaire, et donc de minimiser la confusion entre les classes des mots visuels. Les arbres de décision proposent une quantification qui n'est pas du tout compacte, par contre ils s'intéressent également à l'utilisation des informations de classe. En effet, les différentes décisions prises dans les nœuds de l'arbre visent à optimiser l'information mutuelle entre les décisions et les catégories, et ainsi séparent judicieusement les différentes classes. Cela fonctionne très bien lorsque les labels de chaque patch sont disponibles. Ce n'est pas toujours le cas. Si l'on assigne aux patchs les labels de l'image, alors les patchs apparaissant sur le fond sont labellisés comme » objet « . Si le fond est le même pour plusieurs catégories, les patchs de fond seront associés à des labels différents, selon l'image où ils ont été extraits. Ainsi, les décisions cherchent parfois à différencier des patchs qui ne peuvent pas l'être. Cependant, les expériences montrent que ces confusions n'entraînent aucune baisse de performance (voir table 4.1). Cela est probablement du à la quantité importante d'informations présentes dans le descripteur, ainsi qu'à la capacité du classifieur SVM à sélectionner automatiquement les informations utiles pour la classification.

La méthode proposée dans le chapitre précédent s'attaque à ce problème en accumulant des statistiques sur plusieurs images de la même catégorie et entre les images des différentes catégories. Le but est d'identifier ce qui est caractéristique d'une classe (ce qui revient souvent dans les images d'une catégorie, mais jamais dans les autres) et ce qui est caractéristique du fond (ce qui apparaît dans toutes les catégories d'images). Ainsi sont créés certains mots spécialisés pour une classe, et d'autres génériques et partagés entre les classes.

Il faut noter que les approches présentées dans ces deux chapitres produisent des tailles de vocabulaire très différentes. Là où la méthode du chapitre précédent créait quelques milliers de mots visuels, nous obtenons suivant les arbres quelques dizaines voire centaines de milliers de feuilles (avec les paramètres utilisés).

Enfin, la complexité de l'algorithme d'affectation d'un descripteur à un mot visuel est beaucoup plus faible dans le cas des forêt aléatoire qu'avec les méthodes standards de type k-moyennes. La complexité est en $log(N)$ au lieu de N, où N est le nombre de mots visuels, ce qui fait une énorme différence pour les vocabulaires de grande taille. De plus, les méthodes de clustering se basent sur des mesures de distance entre les descripteurs et les centres des clusters et, même si des structures de recherche rapide des plus proches voisins sont possibles, des distances dans des espaces de grandes dimensions doivent tout de même être calculées. Dans le cas des arbres, seul un petit nombre de comparaisons entre une composante du descripteur et un seuil suffit à déterminer la feuille atteinte. Pour ces raisons, cette méthode est particulièrement efficace.

Conclusion

Dans ce chapitre, nous avons présenté une approche qui permet de résoudre efficacement la tâche de classification d'images. Elle utilise des arbres de décision aléatoires afin de classifier les apparences locales d'images. Les résultats

de cette classification sont utilisés pour créer un vecteur de représentation de l'image entière. Un classifieur SVM permet d'apprendre l'importance des décisions prises par les différentes feuilles.

Les résultats de la classification des patchs sont également utilisés pour construire itérativement une carte de saillance, permettant ainsi de guider le processus d'échantillonnage des patchs dans l'image. La méthode proposée est capable de classifier les images en un temps plus court, grâce à cette sélection efficace de l'information pertinente, tout en produisant des résultats comparables aux meilleures méthodes de classification. Les expériences montrent visuellement que ces cartes de saillance permettent de localiser les objets.

Cette méthode qui s'avère efficace pour la classification des images, pourrait être transposée à la localisation ou la segmentation des objets dans les images. Dans le chapitre 6, ces arbres sont utilisés avec succès en combinaison avec d'autres mécanismes pour segmenter des catégories d'objets dans les images.

Un modèle LDA étendu pour la segmentation de catégories d'objets

5

Sommaire

Le modèle génératif proposé chapitre 3 décrit les images comme des collections non ordonnées de concepts visuels, les topics. Rappelons que ce modèle suppose que les topics conditionnent statistiquement la génération des mots visuels, et que les mots visuels eux mêmes conditionnent la génération des patchs. Une fois les topics identifiés, il devient possible de les lier aux patchs des images à travers les mots visuels. Notre modèle peut donc être étendu naturellement à des applications de segmentation objet/fond. C'est ce que nous nous proposons de faire dans ce chapitre, au moyen d'un modèle à variables latentes d'aspect combinant des mécanismes ascendants et descendants.

5.1 Introduction

Le problème de la segmentation d'images ou de l'étiquetage de régions d'images est un problème clef de la vision par ordinateur. Il consiste à séparer ou à grouper les pixels d'une image en régions cohérentes, qui sont en général des éléments que les humains considèrent comme des instances d'objets ou de parties d'objets. Ce problème a reçu beaucoup d'attention par le passé. Beaucoup d'approches différentes ont été développées, utilisant diverses propriétés des images comme la couleur, la texture, les contours, le mouvement, etc. [28].

La segmentation d'images a très longtemps été considérée comme un problème non-supervisé. Or, comme nous l'avons vu dans l'introduction (section 1.1.3), ce problème est étroitement lié à celui de l'interprétation des images. Aussi, après avoir été délaissée quelques temps, la segmentation d'images bénéficie d'un regain d'intérêt, tirant parti des récents progrès obtenus en reconnaissance d'objet et en apprentissage machine.

Les algorithmes proposés ici ont pour but de reconnaître, localiser et segmenter les objets simultanément. La position, l'échelle et l'orientation des objets peuvent être quelconques. Il ne s'agit pas d'une segmentation telle qu'elle est généralement envisagée, mais d'une segmentation fond/forme (c'est-à-dire déterminer quels sont les pixels qui appartiennent aux objets d'intérêt).

(a) (b) (c)

FIG. 5.1 – *Cette figure illustre la tâche étudiée dans ce chapitre et dans le suivant. L'image (a) se trouve segmentée en (b) par une méthode classique, alors que le vélo qui est un élément important dans la compréhension de la scène devrait faire l'objet d'une unique région, comme dans (c).*

Cette tâche de segmentation de catégories d'objets est illustrée figure 5.1. Une segmentation de l'image (a) utilisant uniquement des informations bas niveaux - ici la couleur (image (b)) - n'a capturé quasiment aucune des informations sémantiques de l'image. En particulier, nous remarquons que le vélo est absorbé par les autres régions. Si nous sommes intéressés par la catégorie des vélos, une segmentation comme (c) serait la solution idéale attendue pour cette tâche.

Pour réaliser cette tâche, nous supposons l'existence de catégories d'objets. Nous allons construire automatiquement des modèles de ces catégories à partir d'exemples d'apprentissage, et utiliser les modèles lors de la segmentation. C'est en cela que cette segmentation est différente de la segmentation d'images habituellement envisagée (limitation à une interprétation bas niveau de l'image). Grâce aux modèles d'apparence ainsi appris pour chaque catégorie la compréhension des images est possible, et la segmentation se trouve enrichie par des critères de plus haut niveau qui guident le regroupement des pixels.

Dans ce chapitre, nous proposons une méthode de segmentation de catégories d'objets, inspirée du modèle LDA [4], et dans le prochain chapitre nous présenterons un modèle combinant un sac-de-mots avec un champ de Markov.

Le reste du chapitre est organisé comme suit. Nous présentons d'abord dans la section 5.2 l'état de l'art commun aux méthodes de ce chapitre et du suivant, en les positionnant par rapport aux travaux existants. Puis dans la section 5.3 nous décrivons le premier modèle proposé et son estimation. Ensuite, la section 5.4 est consacrée aux expériences menées sur une base d'images naturelles. Enfin nous concluons le chapitre.

5.2 État de l'art

Il a été montré récemment [19] que les modèles considérant les images comme des ensembles de mots visuels, modèles que nous avons utilisés jusqu'à maintenant pour la classification d'images, peuvent être appliqués avec succès à la localisation de classes d'objets dans les images. Ce type de méthode est particulièrement adapté à nos besoins puisqu'il permet de gérer de très fortes variations d'apparence. En revanche, il ne permet qu'une localisation grossière des objets, se limitant à une « boîte englobante » dans l'image. Ces modèles peuvent également être combinés à un processus de Dirichlet, permettant de produire des clusters de localisation spatiale [86]. Ce modèle définit des distributions gaussiennes sur les positions des mots visuels, chaque gaussienne pouvant être interprétée comme un cluster de mots visuels associé à un objet. Le processus de Dirichlet permet l'estimation automatique de la complexité du modèle, c'est à dire du nombre d'objets ou de clusters présents dans la scène. Bien que la modélisation de la forme des objets avec des gaussiennes permette de les localiser, la segmentation produite est très grossière, et la frontière entre ces objets très floue.

Cao *et al* [9] ont essayé de surmonter cette limitation du modèle en proposant un modèle à variable latente spatialement cohérent. La représentation des images utilise l'association entre des régions segmentées homogènes et des mots visuels extraits à l'aide d'un détecteur de points d'intérêt. Les régions sont obtenues en utilisant une méthode de segmentation pour sursegmenter l'image en un grand nombre de régions. Ainsi les régions produites ne contiennent en général pas de frontière d'objet. La classe de chaque région est déterminée par les mots visuels qu'elle contient. Les régions permettent un bon point de départ pour la tâche de segmentation, ce qui constitue un avantage de la méthode. Cependant, il faut faire un compromis entre la taille et la qualité des régions initiales produites. D'un coté, peu de régions permettent une estimation plus rapide et plus fiable des classes, grâce à un plus grand nombre de mots visuels extraits. De l'autre, cela augmente le risque que certaines des régions contiennent des parties provenant d'objets différents, et ces erreurs ne peuvent être corrigées par la suite. Nos deux approches ne sont pas sujettes à ce compromis puisque dans notre cas, nous ne dépendons d'aucune segmentation initiale.

D'une manière générale, les champs de Markov (MRF) [23] et leurs variantes (CRF pour *Conditional Random Field* [40, 80, 91], DRF pour *Discriminative Random Field* [79]) ont une longue histoire liée à celle de la segmentation des images. Un des avantages majeurs des MRF est la *régu-*

larisation qu'ils offrent. En effet ces modèles définissent des distributions de probabilité sur les labels des pixels (ou sur les labels des patchs). Ces distributions suivent une relation d'indépendance conditionnelle. Intuitivement le label d'un pixel (ou d'un patch) ne dépend que des labels de ses voisins directs, généralement définis par une grille régulière sur l'image, de connectivité 4 ou 8 le plus souvent. La distribution du champ aléatoire est ensuite combinée à des évidences locales extraites des images ; par exemple le mot visuel associé à un patch pourra augmenter la vraisemblance d'avoir une certaine classe en une certaine position dans l'image. Les contraintes de voisinage peuvent ensuite résoudre les ambiguïtés provenant des évidences locales en propageant spatialement les preuves locales dans l'image.

Cette notion de propagation spatiale est présente dans les deux méthodes. Pour la première, elle est permise par l'utilisation de nombreuses régions qui se chevauchent et qui se partagent l'information entre les patchs (ce mécanisme est détaillé section 5.3). Dans le cas de la deuxième méthode, un MRF est intégré au modèle.

Shotton *et al* [80] proposent d'utiliser un CRF pour apprendre un modèle discriminatif des classes, qui incorpore des informations d'apparence, de forme et de contexte. Notre deuxième modèle est assez similaire, bien que dans leur cas, ce soient des scènes et non des objets qui soient segmentées. En particulier, au lieu de considérer un modèle de fond parfaitement générique, leur méthode interprète l'image en définissant des sous-catégories de fond apprises (comme l'herbe, le ciel, la route, etc.). A la différence de leur modèle, le notre est capable de modéliser séparément les différentes instances d'objet, même si elles appartiennent à la même classe. Ainsi, un modèle d'apparence propre à chaque instance peut être développé et combiné au modèle générique partagé par toutes les instances de la classe.

Traitant du même problème de l'interprétation de scène, Verbeek et Triggs [90] ont proposé de combiner un modèle de champ de Markov, pour les dépendances locales, avec un modèle de topics, estimé au niveau de l'image. Par rapport à un simple modèle MRF, le modèle de topics supprime les petites régions qui appartiennent à des catégories minoritaires de l'image, permettant une bonne cohérence au niveau de l'image. Par rapport à un modèle pLSA, les patchs appartenant aux différentes classes sont regroupés en zones connexes, grâce aux contraintes de voisinages imposées par le MRF. Les objets sont donc définis plus précisément. Le modèle que nous proposons dans ce chapitre se base sur le modèle LDA, très similaire à pLSA. Cependant, si les objets à segmenter sont trop petits, ils seront manqués par [90], qui applique pLSA à toute l'image. C'est pourquoi dans la méthode proposée, la distribution sur les topics n'est pas la même sur toute l'image, mais chaque région de l'image possède sa propre distribution sur les topics. De plus, les segmentations produites par [90] restent relativement grossières, car les gradients de l'image, et la cohérence d'apparence des objets ne sont pas pris en compte. Ces deux critères sont intégrés au modèle du chapitre suivant.

Dans le même esprit, un modèle combinant un MRF et un processus de Dirichlet a été proposé dans [69]. Leur modèle de mixture est associé à un processus de Dirichlet, estimé de façon non supervisé qui permet de sélectionner automatiquement le nombre d'éléments de la mixture, c'est à dire le nombre de régions qui interviennent dans le champ de Markov. Leur modèle

a été appliqué à la segmentation non-supervisée d'images SAR, RADAR et MR. Dans notre deuxième modèle, nous utilisons le même principe, mais ici les composants de la mixture représentent les différents objets à segmenter.

Winn et Shotton [100] ont utilisé un CRF amélioré basé sur un ordre spatial entre les parties d'objet pour gérer les occultations. Tout comme notre deuxième modèle, les différentes instances de la même catégorie sont distinguées et les occultations entre objets sont explicitement modélisées. Cette modélisation est basée sur l'organisation relative des objets. Cependant, dans sa forme actuelle, le modèle gère uniquement des changements d'échelles très limités et il ne modélise qu'une seule catégorie d'objets (mais plusieurs objets) par image.

Parallèlement à cette première ligne de recherche, des méthodes permettant une segmentation précise (sans reconnaissance des objets cette fois) ont été développées. Notons qu'elles obtiennent des segmentations d'objets de grande qualité [73, 53] dans un contexte où la position des objets est supposée interactivement définie par l'utilisateur, par une boîte englobante ou par un tracé grossier des parties de l'objet. Dans le cas de la méthode [73], le principe est le suivant : les distributions de couleur de l'image sont modélisées pour le premier plan et pour le fond à l'aide d'une mixture de gaussiennes. Ces distributions sont réestimées itérativement, et après chaque itération une minimisation d'énergie par *graph-cut* est effectuée afin de séparer le premier plan et le fond. La fonction d'énergie est définie par un MRF pour un champ de label premier-plan/fond donné, et dépend de la similarité entre pixels voisins qui ne possèdent pas le même label, et de la vraisemblance de la couleur des pixels étant donné le modèle de couleur pour le premier plan et le fond.

Ces algorithmes interactifs obtiennent des résultats précis, et l'étape suivante est de s'affranchir de l'intervention de l'utilisateur. Notre but est de segmenter les objets qui se trouvent dans les images en spécifiant uniquement la ou les catégories d'objets que l'on souhaite extraire de l'image, par exemple : « segmente-moi tous les moutons présents dans cette image ».

Cependant, les segmentations obtenues avec un MRF sans aucun modèle de forme des objets produisent rarement des segmentations correctes, c'est pourquoi plusieurs auteurs ont tenté de fusionner ces deux concepts. Nous terminons donc cette discussion sur les méthodes existantes par une présentation des différentes façon d'intégrer un modèle de forme dans le processus de segmentation. Citons par exemple Kumar *et al* [39] qui proposent une méthodologie pour combiner un CRF et un modèle *pictoriel* de structure. La partie CRF produit une segmentation objet/fond tandis que le modèle pictoriel encourage le CRF à suivre la forme de l'objet.

D'autres approches se focalisent uniquement sur le modèle décrivant l'apparence des objets dans les images, sans faire appel à aucune technique de régularisation. Leibe et Schiele [51], qui ont été parmi les tous premiers à faire de la segmentation d'objets, utilisent des images segmentées à la main pour apprendre des masques de segmentation correspondant aux mots d'un vocabulaire visuel. Ensuite, un modèle implicite de forme permet de localiser les objets et de segmenter l'image en combinant des masques de segmentation locaux correspondant aux entrées du vocabulaire visuel. Un système de vote dans un espace de Hough sur la position, l'échelle et la rotation de l'objet est utilisé pour obtenir un ensemble de primitives locales cohérentes

qui s'accordent sur la segmentation de l'objet. Les primitives locales erronées sont ainsi filtrées.

La méthode proposée par [6] s'intéresse également à la tâche de segmentation objet/fond. Elle apprend une forme spécifique et est invariante à la texture. Une segmentation multi-échelle ascendante (*bottom-up*) est combinée à des « prototypes » (*templates*) de forme afin de produire la segmentation finale. Cette méthode repose elle aussi fortement sur la segmentation initiale.

Winn *et al* ont proposé la méthode [99] où la forme et l'apparence des catégories d'objets sont apprises à partir d'un ensemble d'images d'apprentissage et les nouveaux objets sont segmentés en adaptant une version déformée de ce modèle à géométrie rigide à l'image considérée.

Bien que les méthodes basées sur un modèle de forme de l'objet soient robustes à de petites variations locales de forme, les contraintes géométriques fortes dont elles dépendent les rendent mal adaptées à des classes d'objets faiblement structurées ou à de larges changement de point de vue. De telles classes nécessitent des modèles plus flexibles, ce qui est le cas des deux contributions proposées.

5.2.1 Description de l'approche

Notre approche possède plusieurs caractéristiques communes avec les approches mentionnées plus haut. Tout d'abord elle combine des stratégies ascendante et descendante.

Le processus ascendant est très similaire aux méthodes considérées jusqu'à maintenant. Il consiste à échantillonner et à normaliser en taille des patchs d'images, qui sont représentés par des descripteurs SIFT [54]. Ces descripteurs sont ensuite quantifiés en un ensemble discret de labels, les *mots visuels*. Chaque patch est décrit par le mot le plus proche. Ce processus est illustré figure 5.2. À partir de cette étape, les images sont vues comme des ensembles d'occurrences de mots visuels. Comme le processus affecte des labels objet/fond à chaque patch, la segmentation au niveau du pixel nécessite un processus additionnel, responsable de la combinaison des labels de patchs en hypothèses au niveau du pixel (décrit section 5.3.3).

Le processus descendant intègre des modèles d'objets et les utilise pour obtenir une cohérence globale, en combinant des informations locales, fournies par le processus ascendant. La plupart des méthodes citées précédemment ne peuvent être utilisées ici, à cause des fortes variations d'apparence d'objet. Les modèles géométriques comme le modèle implicite de forme (ISM [51]) nécessiteraient un grand nombre d'images d'apprentissage afin de capturer les larges variations d'apparence. Les approches basées sur des caractéristiques de contours [7] ne sont utilisables que lorsque les contours d'objets sont suffisamment stables. Par conséquent, nous allons adopter une approche plus flexible.

Pour la reconnaissance de catégories complexes d'objet, nous avons vu précédemment que le modèle par sac-de-mots [13] est considéré comme l'un des plus efficaces. Nous avons également considéré les modèles à variables latentes d'aspect, en particulier les modèles pLSA [31] et LDA [4], et nous avons montré dans le chapitre 3 qu'ils permettent de modéliser l'information sur les classes.

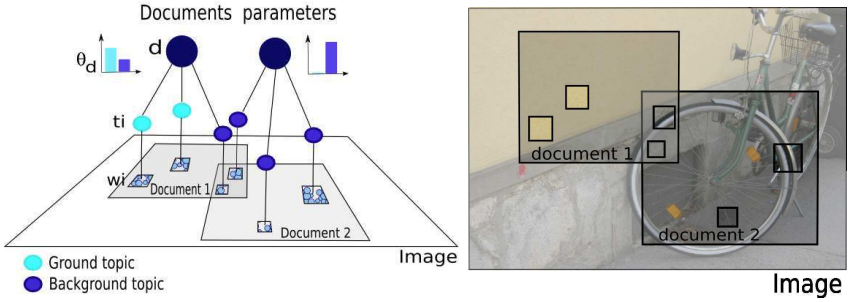


FIG. 5.2 – *Deux documents qui se chevauchent dans une image. Deux des patchs, vus par leur mot visuel (w), appartiennent au topic d'objet (t) et quatre d'entre eux appartiennent au fond. La distribution des documents sur les topics est représentée par les histogrammes.*

L'utilisation de méthodes basées sur les variables latentes d'aspect pour la segmentation d'images s'avère attractive pour différentes raisons. Tout d'abord, l'apparence des objets (représentée par des topics) peut être automatiquement découverte et apprise, limitant la quantité de supervision requise. De plus, la flexibilité d'un tel cadre de travail permet de faire face aux larges variations d'apparence et de forme que présentent les catégories d'objets. Cependant, comme déjà mentionné dans la section précédente, les objets nécessitent de couvrir des parties importantes de l'image de façon à constituer un topic dominant. Ceci n'est pas le cas lorsque les objets sont petits (voir par exemples les images de la base TU-Graz02, figure 1.6). De plus, comme il n'y a aucune contrainte géométrique, ces méthodes ne sont a priori pas bien adaptées aux tâches de détection et de segmentation.

La contribution de ce chapitre est un nouveau modèle graphique, illustré figure 5.3, qui permet de représenter les images et les objets, de façon à résoudre les deux problèmes soulevés. Notre modèle, illustré par la figure 5.2, consiste à décrire les images par des ensembles de documents locaux, multi-échelles qui se chevauchent. Dans ce cas, même les petits objets constituent le topic principal d'au moins un petit nombre de documents, et peuvent ainsi être découverts. Chaque patch d'image (représenté par un mot visuel) appartient à plusieurs documents. Le processus d'affectation d'un label (objet ou fond) aux patchs est fait au niveau du document, qui est un niveau semi global. Cependant, les documents se chevauchent et partagent les patchs d'images, donc les décisions semi-locales sont propagées tout le long de l'image, comme le permettent les champs de Markov.

Une étape d'apprentissage est utilisée pour calculer les a priori sur les apparences des classes d'objet. Cette étape considère un certain nombre d'exemples d'objets.

5.3 Modèle multi-documents

5.3.1 Description

Les images sont décrites comme des ensembles non ordonnés de patchs. Chaque patch est lui-même représenté par un *mot visuel* ainsi que par sa position dans l'image (voir figure 5.2). Notre approche est générative : les mots visuels sont supposés provenir de facteurs sous-jacents, nommés *topics*

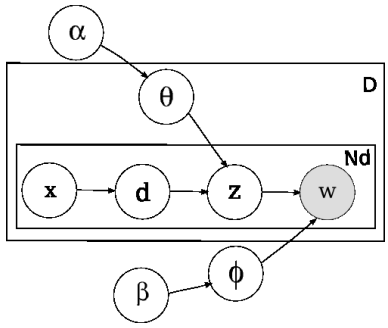


FIG. 5.3 – *Modèle graphique associé à la méthode proposée*

qui décrivent les aspects latents des images, comme avec le modèle LDA [4]. En pratique, nous allons utiliser uniquement deux topics, un pour représenter l'objet, et l'autre pour représenter le fond.

Une image est décrite comme une multitude de documents différents ($d \in D$) correspondant à des régions qui se chevauchent. Les documents sont choisis pour couvrir uniformément les images. Chaque document possède sa propre distribution sur les topics, notée θ_d. Par contre, au sein de tous les documents, la probabilité pour un topic de générer un mot est la même. La distribution des topics sur les mots, notée ϕ est générée suivant une distribution de Dirichlet d'hyper-paramètre β (comme dans le cas du modèle présenté chapitre 3) . Le modèle est illustré figure 5.2.

Modéliser une image I avec notre modèle suppose qu'elle ait été construite à partir du processus génératif suivant :

1. Tout d'abord, la distribution $\theta_d \sim Dir(\alpha)$ est échantillonnée pour chaque document d, où $Dir(\alpha)$ est une distribution de Dirichlet d'hyper-paramètre α, fournissant les distributions sur les variables latentes *topics*,
2. Pour chaque observation (c'est à dire un patch associé à un mot visuel w et une position x) :
   (a) choisir un document d, équiprobablement parmi l'ensemble de documents contenant x. $p(d|x) = 0$ si $x \notin d$ et $p(d|x) = \frac{1}{N}$ si $x \in d$, où N est le nombre total de documents contenant x.
   (b) échantillonner un topic z à partir de la distribution multinomiale de paramètre θ_d : $z \sim Mul(\theta_d)$
   (c) enfin générer un mot visuel w conditionnellement à z à partir de la distribution multinomiale ϕ, $w \sim Mult(\phi)$.

La distribution conjointe $p(w, d, z, x)$ prend la forme du modèle graphique présenté figure 5.2. La marginalisation sur les topics z et les documents d permettent de récrire la probabilité $p(w|x, \phi, \alpha, \beta, I)$:

$$p(w|x, \phi, \alpha, \beta, I) = \sum_{d \in D} \int_{\theta} \sum_{z \in Z} p(w|z, \phi) p(z|d, \theta) p(d|x) d\theta \qquad (5.1)$$

où w représente le mot visuel, x sa position, Z l'ensemble de topics, D l'ensemble de documents, I l'image, et θ et ϕ sont les distributions multinomiales mentionnées plus haut.

5.3.2 Estimation du modèle

Dans le modèle que nous venons de décrire, la position des patchs x et les mots visuels correspondants w peuvent être observés directement. Les hyper-paramètres α et β prennent des valeurs fixes. L'estimation du modèle consiste à calculer les distributions multinomiales θ et ϕ en fonction de leur a priori de Dirichlet, de paramètres α et β, et sachant l'ensemble de x et w observés dans les images.

L'estimation est faite selon le critère de maximum de vraisemblance : M patchs sont collectés de façon à obtenir l'ensemble $(x_1, w_1), \ldots, (x_M, w_M)$. Nous voulons calculer θ et ϕ qui maximisent $p((x_1, w_1), \ldots, (x_M, w_M)|\theta, \phi, \alpha, \beta)$.

Le modèle donné par l'équation (5.1) est trop compliqué pour être estimé directement, nous avons utilisé une technique d'échantillonnage de Gibbs pour l'estimation. Pendant l'estimation, les affectations aux topics (variables cachées de notre modèle) sont estimées conjointement avec θ et ϕ. Le processus d'estimation est très similaire à celui décrit chapitre 3, dans la section 3.2.2. En effet, il intègre les mêmes distributions de probabilité. Comme précédemment, les justifications et détails théoriques sur cette façon efficace d'estimer le modèle peuvent être trouvés dans [26]. Le processus de Gibbs fonctionne de la façon suivante. Les documents sont initialisés comme possédant une distribution équiprobable sur les topics, ensuite nous itérons pour estimer la probabilité a posteriori $p(z|w)$.

Notons également que pour rendre l'estimation possible, seule une image est traitée à la fois. Nous avons typiquement quelques milliers de documents par image. L'estimation simultanée de toutes les images serait impossible. Par conséquent, les documents de différentes images sont supposés indépendants pour l'estimation.

Comme nous l'avons dit précédemment, une étape d'apprentissage est utilisée pour acquérir un fort a priori sur les distributions des topics sur les mots (ϕ). Cela donne une bonne initialisation des affectations des mots aux topics, et guide ainsi de façon efficace le processus d'estimation complet. Néanmoins, la distribution ϕ peut être adaptée à chaque image particulière, le modèle apprend à quoi ressemble chaque topic de façon spécifique à chaque image.

5.3.3 Des labels de patchs aux pixels

À la fin du processus d'estimation, tous les patchs possèdent une probabilité d'avoir été générés par l'un des topics de classes. Ces patchs correspondent à des fenêtres d'image carrées, utilisées pour calculer le mot visuel. De façon à calculer la probabilité pour un pixel p_x d'appartenir à une classe d'objet (correspondant au topic z), il faut accumuler la connaissance sur les patchs $\mathcal{P}$ contenant ce pixel. Cela est modélisé par un modèle de mixture, où les poids (probabilité pour un pixel d'avoir été généré par un patch $p(p|\mathcal{P})$) sont des fonctions de la distance entre le pixel et le centre du patch.

$$p(class(p_x) = z) \propto \sum_{\mathcal{P}_i \ni p_x} p(t_i = z) p(p_x|\mathcal{P}_i) \tag{5.2}$$

où t_i dénote le topic du patch $\mathcal{P}_i$.

Cela peut être vu comme une synthèse de tous les labels correspondant au même pixel. Dans les régions où les patchs voisins ne s'accordent pas, la confiance sera faible, au contraire, lorsque les patchs voisins s'accordent, la probabilité pour un pixel d'appartenir à une classe sera plus forte.

5.4 Résultats expérimentaux

5.4.1 Bases d'images

Les expériences ont été menées sur 3 bases d'images différentes. La première est la base TU-Graz02, présentée section 1.3.2, dont on considérera la classe des vélos. On considérera également la base des oiseaux 1.3.3 et la base des papillons 1.3.4.

5.4.2 Paramètres expérimentaux

Pour toutes les expériences présentées, les descripteurs locaux sont extraits selon une grille dense, à différentes échelles. Nous extrayons environ 10 000 patchs par image. Ainsi l'échantillonnage est suffisamment dense pour que chaque pixel appartienne à suffisamment de patchs. Chaque patch est représenté par un descripteur SIFT [54], de dimension 128.

Seuls deux topics sont considérés, l'un pour la classe d'objet et l'autre pour le fond. Durant l'étape d'apprentissage, un vocabulaire visuel est construit, et de forts a priori pour la forme des distributions multinomiales des topics sur les mots (ϕ), codés dans la variable β sont calculés. L'hyper-paramètre α est, comme pour le chapitre 3, composé des valeurs $\alpha_i = 0.5, \forall i \in \{1, .., T\}$. Durant l'étape de test, l'échantillonneur de Gibbs procède à 50 itérations.

Nous considérons tout d'abord des résultats qualitatifs sur les bases des oiseaux et des papillons, puis nous donnons des résultats quantitatifs sur la base Graz, où une évaluation au niveau des pixels est possible.

5.4.3 Résultats qualitatifs

Les expériences sont faites sur les bases des oiseaux et des papillons. Les images sont représentées avec un vocabulaire visuel de 2000 mots, construits à partir de notre modèle de vocabulaire à variables latentes, présenté dans le chapitre 3. Nous avons utilisé les boîtes englobantes pour indiquer approximativement dans quelle partie de l'image l'objet est localisé, et ainsi estimer les topics d'objet et de fond.

La méthode estime pour chaque patch d'une image de test, sa probabilité d'appartenir à une classe d'objet. Par accumulation de la connaissance obtenue sur tous les patchs, nous pouvons estimer une carte de probabilité au niveau du pixel. La figure 5.4 montre quelques exemples de carte de probabilité obtenus au niveau du pixel, à l'aide de notre méthode de segmentation. Ces cartes de probabilité sont données sous forme de masques de transparence, de sorte que plus un pixel est susceptible d'appartenir à un objet, plus le masque est transparent. Nous observons qu'en dépit de l'importante variation d'apparence et de forme, les cartes de probabilité permettent de donner la position et la forme des instances d'objet.

FIG. 5.4 – *Exemples de cartes de probabilité obtenues par notre méthode. L'image est opaque lorsque les pixels ont une faible probabilité d'appartenir à la classe à segmenter.*

Cependant, sur certaines images, les segmentations sont vraiment mauvaises, et ne se concentrent que sur la partie la plus discriminante de l'animal, le reste de l'image ne choisissant pas clairement une classe. Cela peut être expliqué par deux raisons. Tout d'abord, la supervision est faible, les boîtes englobantes contenant l'animal peuvent ne pas être suffisamment informatives pour permettre une estimation précise des topics. Ensuite, les animaux sont souvent observés sur le même fond (leur habitat naturel) et le fond risque donc d'être appris comme une parti du modèle de chaque classe.

Résultats quantitatifs

Nous avons évalué notre méthode en comparant les segmentations qu'elle produit avec la vérité terrain. Pour la base TU-Graz02, les vérités terrains sont disponibles pour 300 des images de vélos. Elle prend la forme de masques de segmentation au niveau du pixel, fait à la main (voir des exemples figure 1.6, du chapitre 1 et figure 5.5). Ces masques permettent d'évaluer la qualité des segmentations produites. Nous allons les comparer aux cartes de probabilité produites par notre algorithme et calculer ainsi un score sur la précision (voir figure 5.5 pour des exemples).

Nous avons montré dans la section 5.3.3 que notre algorithme calcule la probabilité pour chaque pixel d'image d'appartenir à un objet d'une catégorie donnée (résumé par la carte de probabilité). D'un autre coté, nous connaissons les vrais labels de chaque pixel, grâce à la vérité terrain. Il est donc possible d'évaluer les performances de notre algorithme en calculant une courbe ROC pour chaque image. La courbe ROC a été définie dans la section 2.4.1.

Tout d'abord nous allons voir dans quelle mesure, l'utilisation d'une multitude de documents qui se chevauchent pour chaque image donne de meilleurs résultats qu'une méthode plus triviale. Les expériences sont

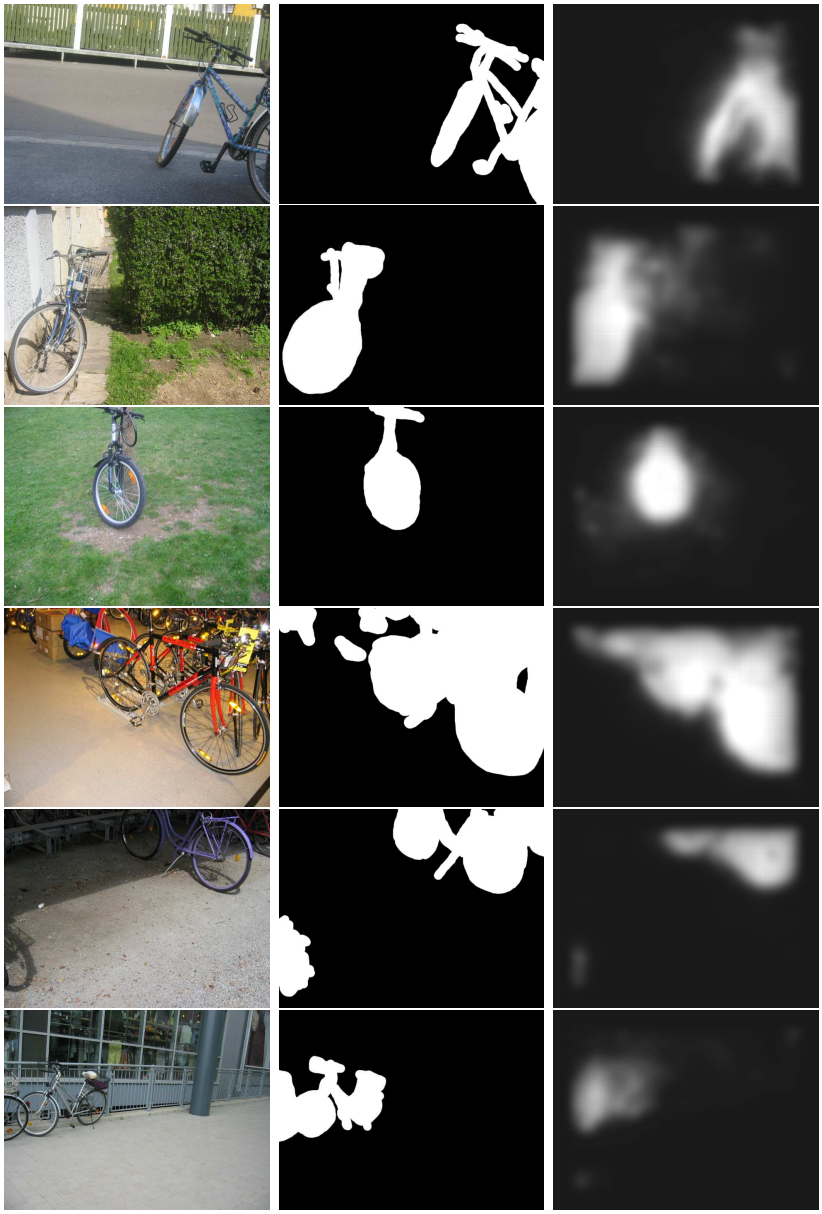


FIG. 5.5 – *Quelques images de vélos de la base TU-Graz02 (gauche), la vérité terrain (milieu) et la carte de probabilité obtenue (droite) en utilisant notre algorithme.*

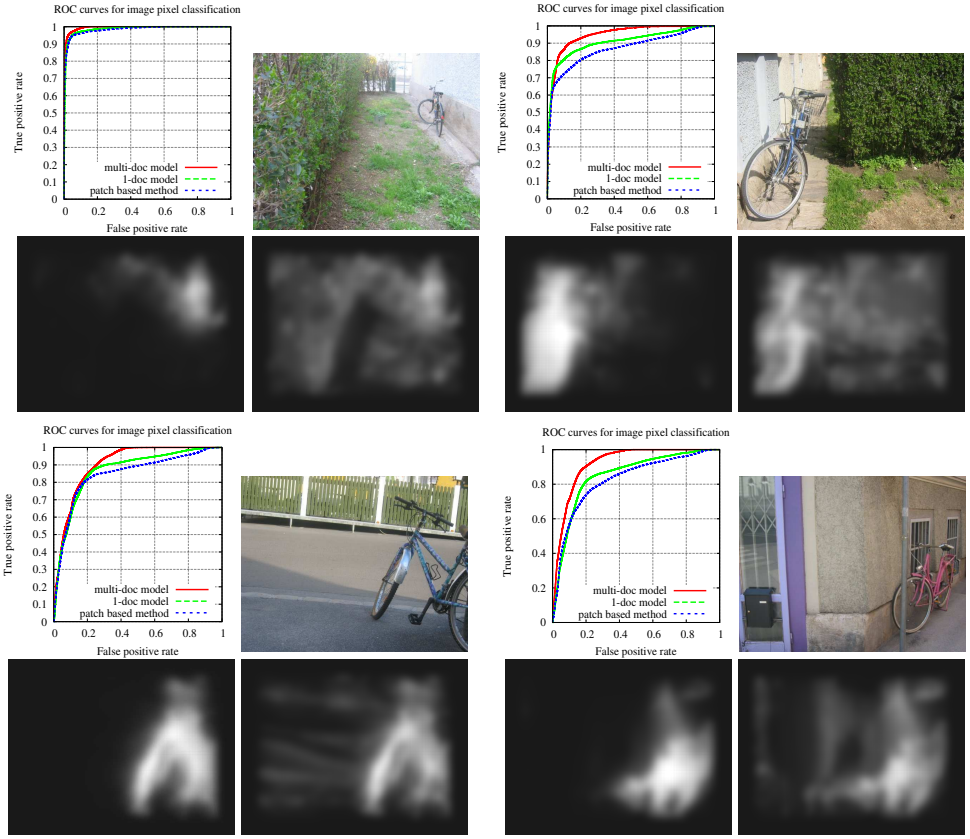


FIG. 5.6 – *Comparaison entre notre méthode et les deux méthodes de référence. Pour chaque image originale (en haut à droite), la courbe ROC est donnée pour notre méthode (en rouge) pour la méthode à un document (en vert) et pour une classification indépendante des patchs (en bleu). Sont également donnés le résultat obtenu par notre méthode et par la méthode à un document (respectivement en bas à gauche et à droite), la troisième méthode, non présentée ici, ayant des masques encore moins précis.*

conduites selon un scénario où tous les labels de patchs d'apprentissage sont connus grâce aux masques de segmentation des images d'apprentissage. Nous avons développé deux méthodes de référence dans un but de comparaison :

1. une *Méthode basée sur les patchs uniquement* : sur les images d'apprentissage, les masques sont utilisés pour fixer les affectations aux topics et ensuite estimer la probabilité pour chaque topic de générer un mot visuel particulier. Nous avons utilisé le théorème de Bayes, $p(t|w) = \frac{p(w|t)p(t)}{p(w)}$, pour calculer la probabilité pour un mot visuel observé d'appartenir à l'objet. Les affectations au niveau du pixel se font de la façon décrite dans la section 5.3.3.
2. une *Méthode basée sur un seul document* : l'image entière est considéré comme un seul document, ce qui est la façon traditionnelle d'appliquer les méthodes à variables latentes d'aspect. A part pour la mixture de documents, le reste de la méthode est la même.

Nous avons comparé notre méthode avec les deux méthodes de référence. Sur la figure 5.6, pour différentes images (partie haute droite), la partie haute-gauche présente les courbes ROC obtenues pour les 3 méthodes

Sup 1	Des masques de segmentation sont disponibles pour toutes les images	0.798
Sup 2	La moitié des images ont des masques de segmentation et la moité restante est annotée avec des boîtes englobantes	0.792
Sup 3	25% des images possèdent des masques de segmentation, et 75% des images des boîtes englobantes	0.787
Sup 4	Seul des boîtes englobantes sont disponibles pour toutes les images	0.761
Sup 5	La moitié des images ont des boîtes englobantes, la moitié restante sont annotées avec juste un label	0.729
Sup 6	25% des images sont annotées avec une boîte englobante, les 75% restantes, avec seulement un label	0.639

TAB. 5.1 – *Pour différentes configurations, correspondant à différents niveaux de supervision (Sup), ce tableau donne sa description ainsi que la moyenne des EER obtenus sur les différentes courbes ROC mesurant la classification au niveau du pixel.*

proposées. Nous montrons également la carte de probabilité obtenue par la méthode proposée (partie en bas à gauche) et pour l'une des 2 méthodes de référence : celle avec un document. L'autre méthode de référence est au mieux similaire. Les résultats montrent le gain fourni par la multitude de documents qui se chevauchent.

Il est intéressant d'analyser la quantité de supervision nécessaire pour avoir de bons résultats. Dans ce but, nous avons fait quelques expériences avec différents niveaux de supervision :

- ▷ des masques de segmentation qui permettent de marquer précisément les patchs d'images qui appartiennent aux objets
- ▷ des boîtes englobantes qui donnent des rectangles contenant les objets
- ▷ des labels au niveau de l'image. La seule information que nous avons est que les objets sont présents quelque part dans l'image, mais aucune information sur la position n'est donnée.

Les trois niveaux de supervision sont combinés en différentes configurations, résumées sur la table 5.1. La question est de savoir à quel point la qualité des résultats dépend de la supervision.

Pour chacune de ces configurations, nous avons produit des cartes de probabilité pour les images de test. Chaque carte de probabilité est utilisée pour calculer une courbe ROC et son EER associé. La moyenne des EER produits sur les différentes images de test est donnée sur la dernière colonne de la table 5.1 pour différents niveaux de supervision.

Comme prévu, nous avons observé que plus le niveau de supervision est élevé, plus les segmentations produites sont précises. La table montre que quelques masques sont suffisants pour assurer une estimation stable des topics d'images et ainsi produire des segmentations satisfaisantes. Les boîtes englobantes donnent des résultats raisonnables, même si la perte en précision est visible. En guise d'illustration, la figure 5.7 montre des masques pour les différents niveaux de supervision décrit dans la table 5.1.

Il est intéressant d'analyser les résultats obtenus sur les images de test. Nous calculons la courbe ROC moyenne pour chaque niveau de supervision, avec des barres d'erreur pour les 3 différentes configurations Sup 1, Sup 4, Sup 5. La courbe est présentée dans la figure 5.8. La déviation standard

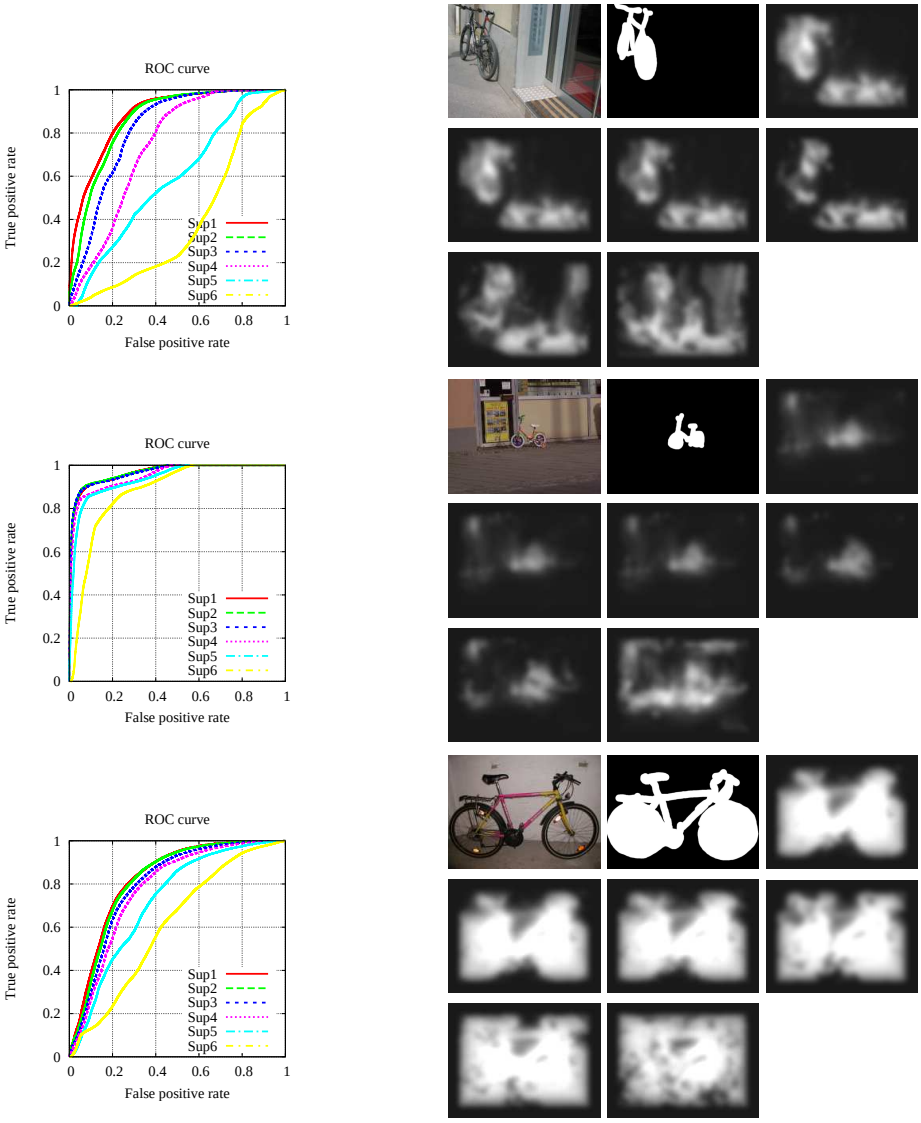


FIG. 5.7 – *Cartes de probabilité pour les différents niveaux de supervision, et les courbes ROC correspondantes sur des images de vélos. Dans l'ordre, sont présentées l'image originale, la vérité terrain, puis les cartes de probabilité pour les configuration 1 à 6.*

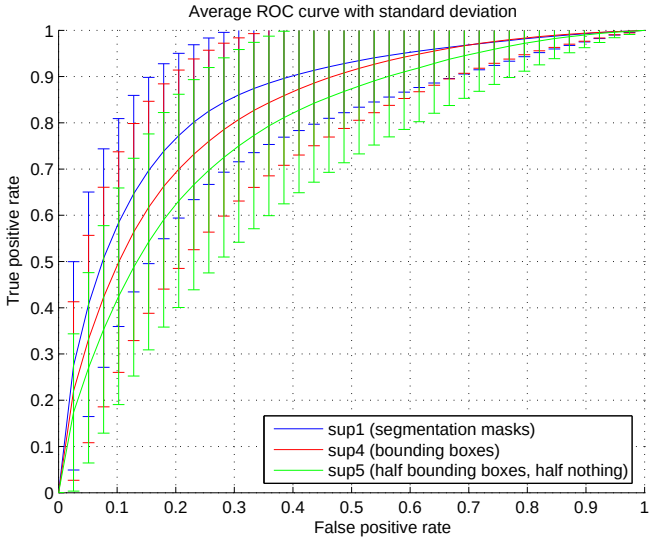


FIG. 5.8 – *Courbe ROC moyenne, obtenue à partir des courbes de toutes les images de test. La déviation standard est également donnée.*

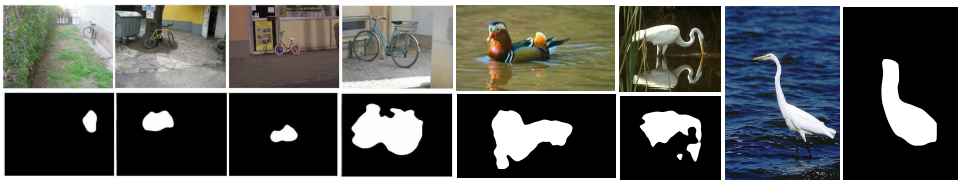


FIG. 5.9 – *Exemples de masques binaires obtenus en seuillant les cartes de probabilité générée par notre méthode sur la catégorie vélo de la base TU-Graz2 et sur la base d'oiseaux.*

peut paraître grande à première vue. Cela peut facilement être expliqué par deux facteurs. Tout d'abord, nous mesurons la performance avec des courbes ROC, comme le nombre de pixels d'objet est souvent très petit (parfois moins de 10% du nombre total de pixels) comparé au nombre de pixels de fond, même si une petite portion du nombre de pixels d'objet est mal classée, cela a un fort impact sur la courbe ROC. Le deuxième facteur est la variabilité et la difficulté des images de la base TU-Graz02 : certains objets sont à peine détectables, même pour des humains. A l'EER, le taux de vrais positifs est presque de 80% pour le plus fort niveau de supervision, et tombe à 76% pour les boîtes englobantes.

Enfin, nous montrons figure 5.9 des masques de segmentation binaires obtenus en seuillant la carte de probabilité. Nous pouvons voir la précision de notre méthode, considérant qu'aucun indice fort (couleur, texture, forme) n'a été utilisé ici pour classifier les pixels.

5.5 Conclusion

Dans ce chapitre nous avons vu comment utiliser un modèle à variables latentes d'aspect pour segmenter les objets dans les images. Contrairement aux modèles classiques, qui ne sont utilisés que pour la reconnaissance, la présence ici de plusieurs documents, de supports spatiaux plus réduits permet d'effectuer simultanément une tâche de reconnaissance et de localisation, qui

prédit la forme des objets. Les documents permettent de propager l'information à un niveau semi-global, et sont donc plus adaptés à la reconnaissance que les méthodes de type MRF où la propagation se fait de façon très locale. Nous obtenons des résultats binaires de segmentation très encourageants.

Cette méthode a cependant des limitations. Tout d'abord, les documents d'images sont nombreux et présentent parfois plusieurs topics dominants simultanés, comme c'est le cas pour les régions frontières entre l'objet et le fond. Ceux-ci couvrent toutes les positions et toutes les échelles possibles. Il serait intéressant de disposer de moins de documents, et que leur nombre, leur position et leur échelle soient automatiquement estimés. Ainsi ces documents seraient idéalement choisis de façon à ne contenir qu'une catégorie, donc un seul topic. Si le nombre de document est réduit, il faut envisager d'autres moyens de propager l'information dans l'image.

Une deuxième limitation est que la méthode n'utilise pas de contraintes de voisinage très locales s'appuyant sur les contours comme le font en général les méthodes de segmentation classiques. Ainsi pourraient être prises en compte des informations supplémentaires de l'image, qui permettraient, lorsque cela est possible, de rendre plus précises les frontières d'objet.

Ces deux limitations sont corrigées dans la méthode de segmentation proposée dans le chapitre suivant.

6 Segmentation de catégories d'objets par combinaison d'un modèle par mots visuels et d'un champ de Markov

Sommaire

Ce chapitre présente une méthode de segmentation de catégories d'objet, tout comme la méthode du chapitre précédent. Elle présente un certain nombre d'avantages par rapport à celle-ci. En particulier, elle permet d'adapter le nombre de documents (ici des *blobs*) automatiquement au nombre d'objets contenus dans l'image. Elle exploite également les dépendances entre patchs, et tient compte des contours de l'image.


« S'il vous plaît ... segmente-moi un mouton ! »

6.1 Introduction

Comme dans le chapitre précédent, nous nous intéressons à la tâche de segmentation de catégories d'objets. Dans le chapitre précédent nous avons utilisé un modèle LDA étendu et obtenu des résultats intéressants. Cependant, pour chaque image, quelques centaines de documents sont considérés. De plus, les informations de contours et de couleurs ne sont pas utilisées. La méthode proposée ici utilise ces informations.

La contribution de ce chapitre est un modèle permettant la segmentation précise des catégories d'objets dans les images, modèle qui tire parti de deux composants complémentaires.

- ▷ un modèle de champ de Markov ou MRF (*Markov Random Field*) utilisé pour sa capacité à produire des champs de labels localement cohérents et s'adaptant aux frontières bas-niveau de l'image.
- ▷ un modèle de type *sac-de-mots* qui permet la reconnaissance et la localisation des objets malgré de fortes variations de point de vue et qui assure une cohérence globale des informations visuelles.

Les frontières d'objets sont définies localement, mais les structures globales (comme les classes d'objets), qui sont primordiales dans la sémantique de l'image, assurent la cohérence de ces informations locales.

Il convient de noter que le modèle que nous proposons repose fortement sur la notion de *mots visuels*. Comme nous l'avons vu, les mots visuels constituent une quantification de descripteurs bas-niveau de l'image. Or, cette quantification locale peut être réalisée de différentes manières, conduisant à différents types de vocabulaires. Nous nous intéressons également à ce point.

Un état de l'art a déjà été proposé dans la section 5.2, qui positionne les travaux de ce chapitre par rapport à l'existant.

Le reste de ce chapitre est organisé comme suit. Nous présentons notre modèle de segmentation d'objets ainsi que les méthodes utilisées pour son estimation. Puis nous proposons une amélioration de cette méthode, basée sur des mots visuels plus discriminants, obtenus à l'aide d'arbres de décision aléatoires. Enfin, nous étudions les résultats expérimentaux, et présentons les conclusions de ces travaux.

6.2 Description du modèle

Nous supposons que chaque image est représentée par une collection de patchs qui se chevauchent fortement.

Le modèle que nous proposons dans ce chapitre a pour vocation de prédire le label de chacun de ces patchs, pour chaque image à interpréter. Les labels désignent une classe d'objets ou le fond. Nous verrons par la suite (section 6.4.2) comment traduire cette labellisation des patchs en une labellisation des pixels, du fait des chevauchements. Individuellement, ces patchs sont représentés par des *mots visuels* qui sont obtenus par quantification vectorielle d'un descripteur d'apparence des patchs.

Les mots visuels sont ensuite utilisés dans un modèle génératif d'objets dans lequel les objets sont définis par des régions elliptiques de l'image (dénotées *blobs*) de position, de taille et de catégories variées. Aucun a priori sur ces régions n'est utilisé, et elles doivent être estimées pour chaque image

à segmenter. Chacune de ces régions contient un ensemble de patchs. Parallèlement, la grille de patchs est vue comme un champ de labels auquel on applique une régularisation entre patchs voisins à l'aide d'une structure de champ de Markov qui permet d'avoir des champs de labels cohérents et qui suivent les contours de l'objet.

Comme nous l'avons expliqué dans l'introduction, la force de notre modèle repose sur la combinaison de ces deux composants différents mais complémentaires. Notre modèle est un modèle paramétrique, complètement spécifié par la probabilité conditionnelle de chacun de ses paramètres. La connaissance de ces lois permet d'échantillonner des valeurs de paramètres en accord avec la loi de probabilité jointe, grâce à un échantillonneur de Gibbs. Cette section décrit dans un premier temps les deux composants du modèle, à savoir la partie basée sur les blobs ainsi que la structure de champ de Markov, puis elle détaille l'estimation des paramètres.

6.2.1 Primitives visuelles

Deux types d'informations sont extraits des images à segmenter : un ensemble de n patchs et une carte de frontières.

Les patchs représentent des zones carrées et de même taille de l'image. Ils sont extraits selon une grille régulière posée sur l'image, de telle façon que les centres des patchs soient positionnés sur les nœuds de la grille. Les patchs se chevauchent fortement. Nous noterons l'ensemble des patchs produits $\mathcal{P} = \{\mathcal{P}_i, 1 \leq i \leq n\}$. Pour chaque patch $\mathcal{P}_i$, quatre caractéristiques différentes sont calculées.

- Tout d'abord, une représentation SIFT [54] est extraite du patch. On suppose qu'un vocabulaire visuel existe, qui aura été créé pendant une phase d'apprentissage, par quantification d'un grand nombre de descripteurs SIFT, représentatifs de ceux observés dans les images à segmenter. La représentation SIFT de notre patch $\mathcal{P}_i$ est associée au mot visuel le plus proche. Ce mot visuel est noté w_i^{sift} ; c'est la première de ces caractéristiques.
- Le patch est ensuite représenté par un descripteur couleur [88] qui est comparé à un second vocabulaire visuel (obtenu de la même façon que le vocabulaire visuel SIFT, mais sur ce nouveau type de descripteur). Ce descripteur est ensuite associé au mot visuel couleur le plus proche, noté w_i^{color}.
- D'autre part, une valeur RGB est obtenue en moyennant la valeur des pixels extraits tout près du centre du patch. Ce vecteur 3D est noté rgb_i.
- Enfin, les coordonnées $X_i = (x_i, y_i)$ du centre du patch dans l'image sont considérées.

Carte de frontières. Une image de contours $\mathcal{G}$ est extraite, donnant en chaque pixel la probabilité de trouver une frontière d'un segment de l'image. Cette carte est calculée avec l'algorithme [57], capable d'extraire les segments naturels de l'image. Il est conçu pour que le résultat soit aussi proche que possible des contours qui seraient tracés par un humain. Il prend en compte, entre autres, les changements d'illumination, de couleur et de texture associés aux frontières naturelles des objets. Voir la figure 6.1 pour un exemple d'image et sa carte de frontières associée.

FIG. 6.1 – *Exemple d'image extraite de la base TU-Graz02 et la carte de frontières associée.*

Au final, toute l'information contenue dans une image est résumée par les caractéristiques de l'ensemble de ses n patchs, *i.e.* $\{w_i^{sift}, w_i^{color}, rgb_i, X_i, 1 \leq i \leq n\}$, et par l'image de contours $\mathcal{G}$.

6.2.2 Modèle génératif

Cette section présente un modèle dont le but est de proposer une première segmentation grossière entre les objets et le fond. Notre modèle s'inspire de [87] et possède des structures spatiales explicites. Une image est supposée être composée de différentes régions que nous appellerons *blobs*. La génération des caractéristiques de patch dépend des paramètres associés à ces blobs. Intuitivement, si une image contient 3 objets, disons une voiture, un piéton et un vélo, cela devrait être modélisé par 4 blobs, un pour chaque objet ainsi qu'un autre pour le fond.

Étant donné les blobs et leurs paramètres, les patchs $\mathcal{P}$ d'une image sont supposés indépendants. Le processus génératif pour un patch est le suivant :

- ▷ sélection d'un blob,
- ▷ échantillonnage des caractéristiques du patch à partir des distributions associées au blob.

Le reste de cette section décrit ce processus génératif.

Le processus de Dirichlet [64] possède une propriété d'auto-renforcement : plus une valeur a été échantillonnée par le passé, plus elle est susceptible d'être échantillonnée de nouveau. Le processus de Dirichlet peut être vu comme la limite lorsque K tend vers l'infini d'un modèle de mixtures à K composants. Notons que, même pour une mixture avec un nombre infini de composants, mais possédant un nombre fini d'échantillons, les composants pouvant être associés sont en nombre fini. Dans notre cas, les blobs tiennent le rôle de composants dans la mixture, et leur nombre n'est ni connu, ni fixé à l'avance. Cela signifie que pour chaque patch nouvellement échantillonné :

- ▷ il peut soit appartenir à un des blobs déjà considérés (dont le nombre est fini), d'indice k avec $1 \leq k \leq K$, avec une probabilité $\frac{N_k}{n-1+\alpha}$ où N_k est le nombre d'échantillons déjà générés par ce blob, et n est le nombre total d'échantillons ;
- ▷ sinon, il peut être échantillonné par un nouveau blob, avec une probabilité $\frac{\alpha}{n-1+\alpha}$, avec α est le paramètre de concentration du processus de Dirichlet.

Chaque blob B_k est associé à un ensemble de paramètres $\Theta_k = \{\mu_k, \Sigma_k, C_k, l_k\}$. La densité sur la position spatiale X_i des patchs associés

est donnée par une loi gaussienne $p(X_i|\Theta_k) = \mathcal{N}(X_i, \mu_k, \Sigma_k)$. Le label de classe associé au blob est noté l_k, et C_k désigne les paramètres d'une mixture de gaussiennes sur les vecteurs couleurs rgb associés aux patchs. Le fond est défini par sa distribution couleur C_{bg} et son modèle spatial est uniforme sur la zone de l'image.

En plus des caractéristiques observées pour chaque patch $\mathcal{P}_i$ (*i.e.* $\{w_i^{sift}, w_i^{color}, rgb_i, X_i\}$) deux variables aléatoires supplémentaires lui sont associées : l'indice du blob qui a généré ce patch, noté b_i, et le composant qui a généré la valeur rgb_i dans la mixture de gaussiennes sur les valeurs RGB du blob, noté c_i.

Étant donné l'indice k du blob qui a généré le patch $\mathcal{P}_i$, les caractéristiques sont supposées indépendamment distribuées, ce qui donne l'équation suivante :

$$p(\mathcal{P}_i|b_i = k) = p(w_i^{sift}|\Theta_k)p(w_i^{color}|\Theta_k)p(rgb_i|\Theta_k)p(X_i|\Theta_k). \tag{6.1}$$

Le modèle de couleur de chaque blob capture les distributions de couleur spécifiques à chaque instance d'objet et à chaque fond d'image, comme dans le modèle Grab-Cut [73]. Cela aide à produire des segmentations d'objets dont l'apparence est cohérente, même si localement la reconnaissance est ambiguë. Notons que ce modèle de couleur joue un rôle très différent de celui du mot visuel couleur w_i^{color}, qui modélise l'information de couleur à l'échelle d'une catégorie d'objets.

Les probabilités de générer les mots visuels SIFT et couleur sont modélisées par des lois multinomiales associées à la catégorie d'un blob, c'est-à-dire $p(w_i^{sift}|\Theta_k) = p(w_i^{sift}|l_k)$ et $p(w_i^{color}|\Theta_k) = p(w_i^{color}|l_k)$.

Ce sont les seules informations qui ne sont pas estimées sur chaque image, mais partagées entre toutes les images à segmenter. Elles sont apprises à partir d'images d'entraînement annotées dans lesquelles les mots visuels sont extraits. Les distributions sont alors estimées par un processus de comptage : on compte le nombre de fois que chaque mot visuel apparaît sur chaque classe.

6.2.3 Champ de Markov sur les labels

Étant donné les catégories associées aux blobs, les affectations des patchs aux blobs $b = \{b_1, \ldots, b_n\}$ déterminent la segmentation de l'image. La qualité de cette segmentation est renforcée par le deuxième composant de notre modèle : un champ de Markov sur les affectations aux blobs. Le MRF modélise l'espérance des corrélations sur les affectations entre patchs voisins, et aligne les changements de labels avec les frontières probables de l'image, selon la carte de frontières. Le MRF est défini sur la grille régulière de patchs à travers une connectivité de 8 voisins.

Ci-dessus, nous avons défini un modèle génératif sur les patchs $p(\mathcal{P}, b|\Theta) = p(b)p(\mathcal{P}|b, \Theta)$, où l'a priori $p(b)$ avait été modélisé par un processus de Dirichlet. Ici, nous allons inclure un composant MRF dans $p(b)$ en définissant une nouvelle probabilité a priori comme le produit de celle provenant du MRF et de celle provenant du processus de Dirichlet. Nous obtenons :

$$p(\mathcal{P}, b|\Theta) \propto p_{dir}(b)p_{mrf}(b|\Theta)p(\mathcal{P}|b, \Theta). \tag{6.2}$$

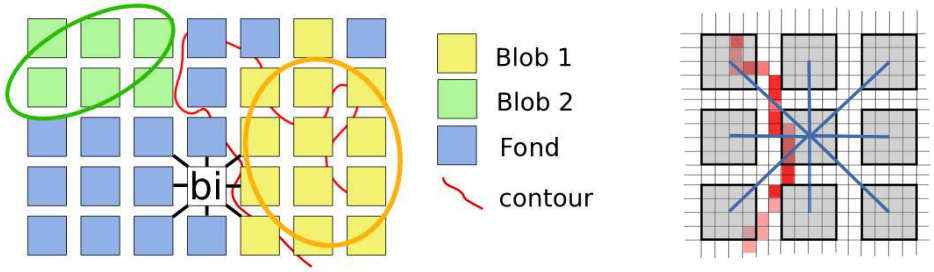


FIG. 6.2 – *Le modèle est influencé par un potentiel de type MRF et les modèles représentant les objets et le fond. Le potentiel MRF est basé sur la carte de frontières. La partie droite donne une vue plus précise d'un patch et de ses 8 voisins.*

Pour simplifier la formulation du MRF, Θ est omis de la notation, et la probabilité est réécrite comme $p(\mathcal{P}, b|\Theta) \propto \exp(-E(\mathcal{P}, b))$ en utilisant la fonction d'énergie :

$$E(\mathcal{P}, b) = U(\mathcal{P}, b) + \gamma \sum_{i,j \in \mathcal{C}} V_{i,j}(b_i, b_j), \tag{6.3}$$

où $\mathcal{C}$ représente l'ensemble des voisins (ou cliques) dans la grille de connectivité 8, γ est un paramètre qui équilibre les deux termes de la formule, et

$$U(\mathcal{P}, b) = -\log(p(\mathcal{P}|b, \Theta)p_{dir}(b)). \tag{6.4}$$

Le modèle du MRF, p_{mrf}, est représenté par le second terme de l'équation 6.3, et le potentiel sur les paires de voisins est défini par

$$V_{i,j}(b_i, b_j) = [l_{b_i} \neq l_{b_j}] \exp(-\beta \Phi_{i,j}), \tag{6.5}$$

où [.] est la fonction indicatrice.

Ce potentiel renforce la cohérence spatiale des labels du patch b_i, et encourage les changements de labels à se produire là où la carte de frontières $\mathcal{G}$ présente des valeurs élevées. La valeur maximale prise par $\mathcal{G}$ entre les centres de patchs $\mathcal{P}_i$ et $\mathcal{P}_j$ est notée $\Phi_{i,j}$, et β est l'inverse de la moyenne des valeurs prises par $\Phi_{i,j}$ dans l'image. Ainsi, $V_{i,j} = 0$ pour les couples de patchs qui sont affectés au même blob, sinon cette valeur représente une pénalité qui est d'autant plus forte que la probabilité d'une frontière entre les patchs diminue (donnée par $\mathcal{G}$). La figure 6.2 illustre cette propriété.

6.2.4 Estimation des affectations

Ci-dessus, nous avons défini un modèle qui combine un processus de Dirichlet et un MRF. Dans cette section, nous allons voir comment utiliser le modèle pour estimer les affectations des patchs aux blobs, notées b, pour une image, ainsi que les affectations des blobs aux classes, notées $l_k, 1 \leq k \leq K$. Ceci est réalisé par un échantillonneur de Gibbs, qui échantillonne successivement les différents paramètres des blobs Θ_k, et les variables associées aux patchs : b_i et c_i. Dans le reste de la section, nous allons considérer les distributions conditionnelles qui sont utilisées par l'échantillonneur de Gibbs.

Étant donné une affectation des patchs aux blobs fixée b, les paramètres de tous les blobs de l'image $\Theta_k = \{\mu_k, \Sigma_k, C_k, l_k\}$ sont indépendamment distribués. Nous supposons un a priori non-informatif sur Θ_k, et nous utilisons

le raccourci de notation suivant $\mathcal{B}_k = \{i \ : \ b_i = k\}$ afin de rendre plus lisible l'écriture des lois a posteriori sur les paramètres. Pour les paramètres décrivant le support spatial des blobs, μ_k et Σ_k, nous avons :

$$\mu_k \sim \mathcal{N}(Mean(\{X_i \ : \ i \in \mathcal{B}_k\}), \frac{1}{N_k} Cov(\{X_i \ : \ i \in \mathcal{B}_k\})), \quad (6.6)$$

$$\Sigma_k \sim \mathcal{W}(Cov(\{X_i \ : \ i \in \mathcal{B}_k\}), N_k - 1), \quad (6.7)$$

où $\mathcal{N}$ est la loi normale et $\mathcal{W}$ est une distribution de Wishart. Le paramètre C_k du modèle de couleur spécifique au blob est estimé en utilisant un algorithme EM (*Espérance-Maximisation* ou *Expectation-Maximisation*) stochastique, qui utilise des échantillons plutôt qu'une espérance dans l'étape E. Enfin, la multinomiale utilisée pour échantillonner le label de classe l_k est donné par :

$$p(l_k|b) \propto \prod_{i \in \mathcal{B}_k} p(w_i^{sift}|l_k) p(w_i^{color}|l_k). \quad (6.8)$$

La variable c_i, qui désigne le composant du modèle de mixtures associé à chaque patch, est obtenue directement par échantillonnage à partir de la loi a posteriori associée à la mixture du blob considéré.

L'affectation des patchs aux blobs b_i est échantillonnée séquentiellement, sachant les paramètres de blobs Θ_i et toutes les autres affectations de patch aux blobs, notées $b_{-i} = b \setminus \{b_i\}$. Il faut distinguer deux cas, l'échantillonnage d'une affectation à un blob qui est déjà associé à d'autres patchs, et d'une affectation à un nouveau blob.

$$p(b_i|b_{-i}, \Theta, \mathcal{P}) \propto \begin{cases} p(\mathcal{P}_i|b_i) \frac{N_{b_i}}{n-1+\alpha} \exp(-\gamma \sum_{i,j \in \mathcal{C}} V_{i,j}) & : \text{ blob existant} \\ p(\mathcal{P}_i|b_i) \frac{\alpha}{n-1+\alpha} \exp(-\gamma \sum_{i,j \in \mathcal{C}} V_{i,j}) & : \text{ nouveau blob} \end{cases} \quad (6.9)$$

Pour calculer l'équation (6.9) pour un nouveau blob, on échantillonne les paramètres du blob comme suit. Le label de catégorie l_k est échantillonné uniformément sur les catégories possibles. Le centre du blob μ_k est échantillonné uniformément sur le support de l'image. Σ_i est choisi isotropique avec un écart type correspondant à la moitié de la plus petite dimension de l'image. Enfin les paramètres de la mixture de couleur C_k sont choisis à partir des couleurs de tous les pixels de l'image.

6.3 Utilisation d'arbres de décision

Le modèle pour la segmentation qui a été présenté dans la section précédente repose sur la notion de vocabulaire visuel, dont le but est de représenter les patchs d'images. La raison principale qui pousse à quantifier les descripteurs de patchs est d'en simplifier la modélisation.

Conventionnellement, les vocabulaires visuels sont créés en utilisant de simples algorithmes de clustering basés sur des distances, comme par exemple l'algorithme des k-moyennes. Il s'agit de l'approche que nous avons eue dans la section précédente. Mais ce procédé est en général coûteux en terme de complexité pour, d'une part, créer le vocabulaire visuel et, d'autre part, associer chaque descripteur de patch à un mot. De plus, comme nous l'avons vu dans le chapitre 3, il n'y a aucune garantie que le vocabulaire ainsi obtenu

soit adapté à la discrimination entre les apparences visuelles des différentes classes. En effet, les mots les plus fréquents ne sont pas souvent spécifiques à un objet, mais partagés entre les classes.

Nous avons montré dans le chapitre 4 que les arbres de classification aléatoires constituent une alternative attractive aux méthodes de clustering standard pour la création de vocabulaires visuels. Ci-dessous, nous décrivons comment les vocabulaires visuels peuvent être remplacés par des arbres de décision dans notre modèle.

6.3.1 Application des arbres de décision à notre modèle

Les arbres de décision sont construits exactement de la même manière que celle décrite dans la section 4.3.2. La seule différence réside dans l'information stockée par les feuilles. Au lieu d'un unique label, comme cela était le cas précédemment, chaque feuille stocke une probabilité sur les classes, qui correspond à la probabilité pour un descripteur atteignant cette feuille d'appartenir à chacune des classes. Elle est simplement calculée pendant l'apprentissage comme le nombre de patchs d'une catégorie qui ont atteint cette feuille, divisé par le nombre total de patchs ayant atteint la feuille.

Comme pour la section 4.3.2, la variance est réduite par élagage (*pruning*) et par l'utilisation de plusieurs arbres et combinaison des résultats. Dans les expériences, l'élagage sera paramétré par un nombre maximum de feuilles par arbre.

Les paramètres qui doivent être spécifiés pour la construction des arbres sont le nombre de tests considérés à chaque nœud parmi lesquels le meilleur découpage est choisi, le nombre d'arbres et le nombre de feuilles par arbre après élagage. Nous étudions l'influence de ces paramètres dans les expériences (section 6.4.3).

Notons que les arbres de décision découpent l'espace des descripteurs, comme le fait un vocabulaire visuel obtenu par un algorithme de clustering. Lors de l'utilisation d'une forêt d'arbres de décisions, chaque arbre produit une quantification différente de l'espace. Comme les arbres de décisions élémentaires sont basés sur l'information mutuelle entre les descripteurs et les catégories, le vocabulaire produit est bien adapté à la discrimination entre les ensembles de catégories.

Rappelons que dans notre modèle original nous avons utilisé deux vocabulaires visuels : un pour les descripteurs SIFT et un pour les descripteurs couleurs. Lorsqu'une forêt d'arbres de décision aléatoires est utilisée, nous obtenons une collection de quantifications différentes. Comme précédemment, chaque patch $\mathcal{P}_i$ est représenté en utilisant la moyenne des valeurs RGB des pixels extraits en son centre rgb_i, et les coordonnées 2d de sa position dans l'image X_i. Pour s'adapter à des vocabulaires visuels définis de façon plus générale, les mots visuels w_i^{sift} et w_i^{color} sont remplacés par une collection de mots visuels w_i^j, avec $j \in \{1, \ldots, J\}$ représentant l'indice parmi J vocabulaires différents. Ces J vocabulaires visuels peuvent être construits en utilisant des techniques de clustering standard, ou bien des arbres de décision, et ils peuvent être basés sur un ou plusieurs descripteurs comme, par exemple, des descripteurs SIFT et des descripteurs couleur.

De façon à refléter ces changements dans notre modèle, nous remplaçons l'équation 6.1, qui donne la probabilité d'avoir un patch sachant les

affectations aux blobs et les paramètres du blob, par

$$p(\mathcal{P}_i|b_i = k) = p(rgb_i|\Theta_k)p(X_i|\Theta_k)\prod_{j=1}^{J} p(w_i^j|\Theta_k). \tag{6.10}$$

Une fois encore, les probabilités des mots visuels sachant les blobs ne dépendent que de la classe du blob, $p(w_i^j|\Theta_k) = p(w_i^j|l_k)$. Ces dernières sont obtenues de façon triviale par comptage et normalisation du nombre de fois qu'un mot visuel apparaît dans une classe, indifféremment de la façon dont a été créé le vocabulaire visuel (clustering ou arbre de décision).

L'échantillonneur de Gibbs ne diffère que pour la variable l_k, qui est maintenant échantillonnée avec :

$$p(l_k|b) \propto \prod_{i\in\mathcal{B}_k}\prod_{j=1}^{J} p(w_i^j|l_k). \tag{6.11}$$

6.4 Évaluation expérimentale

Dans cette section, nous présentons des résultats expérimentaux. Tout d'abord, dans la section 6.4.1, les bases d'images utilisées sont décrites. Dans la section 6.4.2, nous discutons de la façon dont l'estimation des catégories de patchs est utilisée pour obtenir des segmentations au niveau du pixel. Ensuite, dans la section 6.4.3, nous présentons une première série d'expériences qui propose une étude paramétrique de la méthode, avec notamment une investigation sur le rôle du vocabulaire visuel. Dans les sections 6.4.4 et 6.4.5 nous présentons nos résultats expérimentaux qualitatifs et quantitatifs, que nous comparons à l'état de l'art.

6.4.1 Bases d'images

Dans nos expériences, nous considérons des bases d'images difficiles, appropriées à la segmentation objet/fond. Nous considérons principalement les bases suivantes : la base TU Graz-02, les bases Pascal VOC 2006 et 2007 et la base Microsoft (MSRC). Elles ont été introduites respectivement dans les section 1.3.6, 1.3.7 et 1.3.8.

Ces bases contiennent des classes d'objets présentant une grande variation d'apparence, ainsi qu'un fond générique et encombré. De plus, les objets en question présentent des variations d'échelle, d'illumination, de points de vue ainsi que des occultations.

6.4.2 Des labels de patchs aux pixels

Les modèles que nous avons présentés dans ce chapitre se basent sur une représentation à base de patchs, cependant notre but est de produire des segmentations précises au niveau du pixel. En utilisant des patchs qui se chevauchent fortement, nous nous assurons que les segmentations produites seront tout de même très précises, grâce à une simple méthode de post-traitement, très similaire à celle utilisée dans le chapitre précédent (section 5.3.3).

L'échantillonneur de Gibbs décrit dans la section 6.2.4 permet d'obtenir l'estimation des probabilités a posteriori des affectations des patchs aux

blobs, ainsi qu'une probabilité sur les classes pour chaque blob. À partir de ces derniers, nous pouvons calculer la probabilité sur les classes pour chaque patch, en sommant les probabilités pour les blobs d'appartenir aux classes, pondérées par la probabilité pour chaque patch d'appartenir à un blob. C'est la première somme de la formule 6.12.

La probabilité pour un pixel p_x d'appartenir à un objet est obtenue par combinaison des labels de tous les patchs $\mathcal{P}_i$ contenant ce pixel, selon la même technique. Nous considérons un modèle de mixture où les poids sont proportionnels à la distance entre la position du pixel et le centre du patch. C'est la deuxième somme de l'équation 6.12.

$$p(class(p_x) = z) = \sum_{\mathcal{P}_i \ni p_x} \sum_{B_k} p(l_k = z) P(\mathcal{P}_i | B_k) p(p_x | \mathcal{P}_i) \tag{6.12}$$

où l_k désigne la classe du blob B_k.

6.4.3 Étude paramétrique

Dans cette section, nous évaluons les différents ensembles de primitives et les méthodes de construction du vocabulaire, pour la segmentation des images de la base TU-Graz02 avec notre méthode. Les images de cette base ne contiennent qu'une catégorie d'objets, donc la tâche de segmentation peut être vue comme un problème de classification binaire. Ainsi, la précision peut être mesurée en utilisant des courbes précision-rappel qui montrent combien de pixels appartenant à la catégorie d'objets (toutes images confondues) sont correctement classifiés. Pour chaque classe, nous utilisons 300 images annotées pour apprendre le modèle, tandis que la seconde moitié de la base est utilisée pour le test.

Influence des différentes primitives

Notre méthode repose sur l'utilisation de vocabulaires visuels, qui représentent des quantifications des descripteurs de patchs. Nous avons proposé deux façons de construire les vocabulaires visuels : la façon la plus classique (section 6.2), qui utilise un algorithme de clustering, et une technique alternative (section 6.3), qui permet d'avoir des vocabulaires discriminants en utilisant des arbres. Cependant, nous avons observé que le comportement des primitives visuelles est le même quelque soit le vocabulaire visuel utilisé, donc dans cette partie nous n'allons considérer que des résultats obtenus avec un vocabulaire visuel construit par un algorithme des k-moyennes. Dans la section suivante, nous comparons différentes façons de construire les vocabulaires visuels.

Plusieurs primitives sont calculées à partir de chaque patch : un descripteur SIFT, un descripteur couleur, la moyenne des valeurs RGB et la coordonnée 2d dans l'image. Ici, nous étudions l'impact des différentes primitives sur les résultats finaux de segmentation. Nous comparons le modèle complet, noté $w^{sift} + w^{color} + rgb + X$, avec d'autres modèles qui ne contiennent qu'un sous-ensemble de ces primitives. Le composant MRF est utilisé dans notre modèle uniquement dans les expériences qui utilisent la coordonnée spatiale dans l'image X. Des vocabulaires visuels de 5000 mots (resp. 100 mots) ont été créés pour les descripteurs SIFT (resp. couleur).

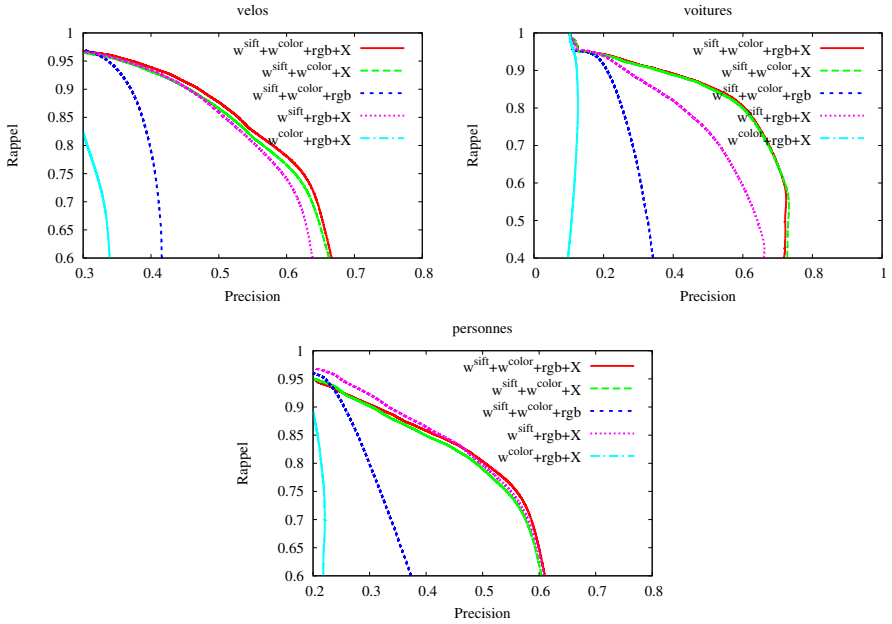


FIG. 6.3 – *L'importance relative des différentes caractéristiques des patchs est illustrée ici. Les mots visuels SIFT (w^{sift}) et couleur (w^{color}), la composante couleur (rgb) ainsi que la position (X) du patch sont combinés de différentes façons. (Voir le texte pour plus d'explications.)*

Le résultat de cette étude paramétrique est reporté dans la figure 6.3. Nous observons que les vocabulaires visuels w^{sift}, w^{color} sont essentiels. Si l'un des deux manque, les performances baissent (sauf pour la classe personne qui n'a besoin que du descripteur SIFT). Toutefois, le descripteur SIFT est plus critique que le descripteur couleur. Les résultats montrent que ces composants responsables de la reconnaissance sont cruciaux pour guider le processus de segmentation.

La régularisation spatiale utilisant le composant MRF et le modèle génératif de blobs améliorent les résultats considérablement, comme la comparaison entre les courbes rouge (toutes les primitives) et bleue (sans l'information spatiale) le montre. La régularisation apporte également une amélioration très notable visuellement.

La primitive *rgb*, utilisée pour les instances d'objets, donne une amélioration pour deux des trois classes. Nous avons observé, que si l'objet est correctement localisé, alors le composant couleur améliore considérablement la précision de la segmentation. Dans ce cas, certains patchs non discriminants peuvent être affectés à l'objet ou au fond, en fonction de leur couleur. Ce phénomène est illustré figure 6.4. Cela montre que le composant couleur RGB peut aider à segmenter une partie de l'objet qui était initialement affectée au fond, mais dont la couleur s'est avérée cohérente avec le modèle couleur RGB de cet objet particulier.

Cependant, il arrive parfois que l'objet ne soit pas localisé précisément, et que le modèle couleur soit bruité. Il peut même détériorer les résultats dans certains cas.

Nous pouvons comprendre le rôle des différents composants du modèle grâce aux différentes illustrations de la figure 6.5. Dans la partie gauche,

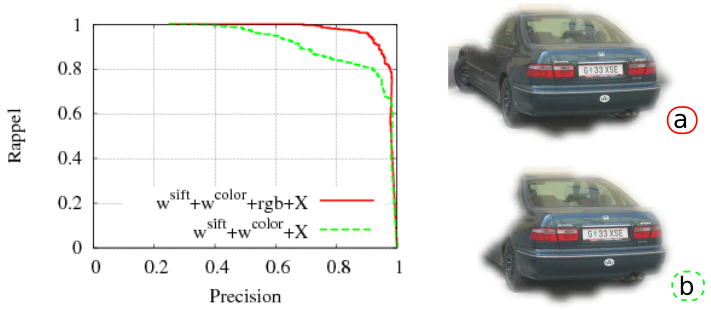


FIG. 6.4 – *Notre modèle avec (a) et sans (b) le composant RGB spécifique aux objets. Une courbe précision rappel est disponible dans la partie gauche, et les images correspondantes sont montrées dans la partie droite.*

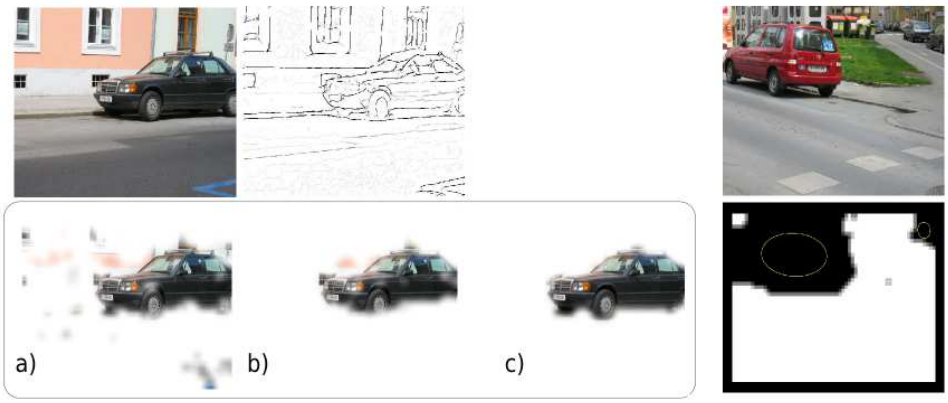


FIG. 6.5 – *Gauche : une image et sa carte de frontières associée, ainsi que les segmentations produites par a) un simple classifieur de patch ($w^{SIFT} + w^{color}$), b) le modèle génératif complet (avec processus de Dirichlet), c) le modèle complet. Droite : le processus de Dirichlet a prédit deux blobs pour différencier les modèles d'apparence de chaque instance.*

nous pouvons voir différentes segmentations obtenues sur la même image en utilisant a) un simple classifieur de patch (chaque mot visuel vote pour une catégorie), b) le modèle génératif seul avec le processus de Dirichlet comme a priori, c) le modèle complet avec le champ de Markov. La partie droite de la même figure illustre nos motivations à intégrer un processus de Dirichlet dans le modèle de génération des blobs. Le modèle a estimé que cette image était mieux décrite avec deux blobs d'objet, et qu'une configuration avec un seul blob serait moins probable. En effet, deux blobs permettent d'avoir deux modèles de couleurs spécifiques ce qui est plus précis qu'un modèle commun.

Comparaison entre les vocabulaires visuels

Dans cette section, nous évaluons la qualité des segmentations obtenues en utilisant un algorithme de clustering (k-moyennes) pour le vocabulaire, ou en le remplaçant par des arbres de décision.

Pour des raisons de simplicité, nous ne considérons ici qu'une seule sorte de descripteur, le descripteur SIFT, pour la partie reconnaissance de notre modèle. La figure 6.6 compare les deux types de vocabulaires visuels sur les

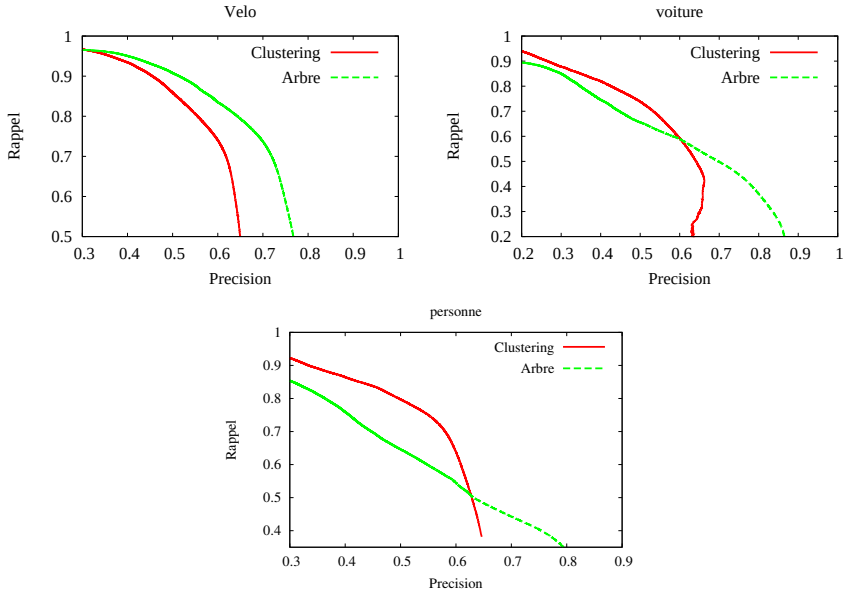


FIG. 6.6 – *Comparaison entre le vocabulaire obtenu avec les k-moyennes et le vocabulaire utilisant des arbres, pour les 3 classes de la base TU-Graz02.*

classes de la base TU-Graz02. Le modèle comprend dans les deux cas : le descripteur SIFT, le composant RGB et la position du patch. Le vocabulaire obtenu par l'algorithme des k-moyennes a 5000 mots visuels, tandis que les vocabulaires obtenus à partir des arbres possèdent 5000 feuilles par arbre. Dans ces expériences, nous utilisons 3 arbres et 50 tests par nœud. Les résultats montrent qu'avec ces paramètres, les vocabulaires basés sur les arbres conduisent à un modèle qui segmente mieux les images que l'algorithme des k-moyennes.

Les arbres aléatoires dépendent de plusieurs paramètres. Il est donc intéressant d'évaluer leur influence sur les résultats de segmentation. Tout d'abord, le nombre de feuilles par arbre est un paramètre important. Il permet de fixer la précision dans la quantification de l'espace. Nous suspections qu'un nombre trop élevé de clusters entraînerait un sur-apprentissage, mais ceci n'a pas été observé dans les expériences. Dans la partie gauche de la figure 6.7, les résultats montrent que la précision moyenne augmente lorsque le nombre de feuilles augmente (jusqu'à 5000) tout en gardant le nombre d'arbres fixé à 3. La partie droite de cette même figure montre l'influence du nombre d'arbres (pour 5000 feuilles). Augmenter le nombre d'arbres améliore légèrement la précision moyenne, mais les résultats dépendent moins du nombre d'arbres que du nombre de feuilles par arbre, ce qui est cohérent avec les résultats reportés dans [62].

Un autre paramètre essentiel est le nombre de tests effectués en chaque nœud afin de choisir le meilleur découpage. Ce paramètre contrôle la quantité d'aléatoire mais a également une incidence sur le temps nécessaire à la construction de l'arbre. La partie gauche de la figure 6.8 présente des courbes précision-rappel obtenues avec le modèle basé sur les arbres, en faisant varier la valeur de ce paramètre depuis 1 (arbre complètement aléatoire) jusqu'à 100 tests par nœud. La partie droite de la figure donne les temps de calculs associés. Dès 10 tests par nœud, les performances augmentent significative-

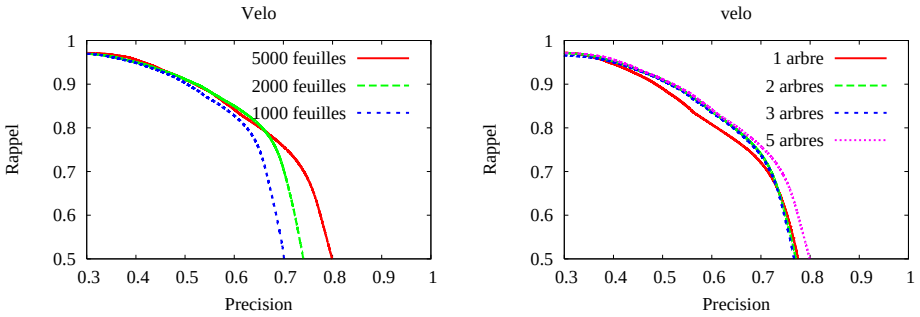
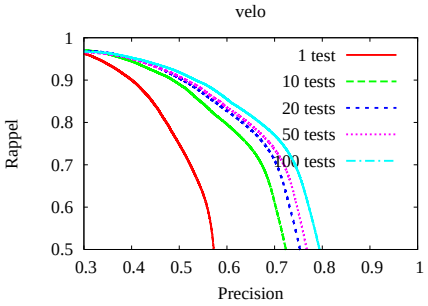


FIG. 6.7 – *Influence du nombre de feuilles par arbre (gauche) et du nombre d'arbres (droite) sur la précision des segmentations produites.*



Méthode	Nœuds	Temps (s)
Clustering		2507
Arbres	100 tests	1342
	50 tests	706
	20 tests	301
	10 tests	174
	1 test	34

FIG. 6.8 – *Gauche : étude de l'incidence du nombre de tests effectués en chaque nœud pour la construction de l'arbre sur le résultat final de segmentation. Droite : temps de calcul nécessaire à la construction de l'arbre ou du vocabulaire visuel.*

ment par rapport au « tout aléatoire », ensuite les performances augmentent encore mais de façon moins significative.

6.4.4 Résultats qualitatifs

Dans cette section, nous présentons des masques de segmentation, obtenus sur les bases TU-Graz02, MSRC et Pascal VOC 2006. Pour chaque classe, les images sont segmentées entre les différents objets d'intérêt et le fond. Pour la base Graz (les vélos, les voitures et les personnes) et pour la base MSRC (résultats en couleur), les modèles d'objets sont construits à partir d'images d'apprentissage segmentées. Dans le cas de la base Pascal 2006, les modèles de catégories d'objets sont construits sur les 50 premières images contenant l'objet, uniquement en utilisant les boîtes englobantes disponibles dans les annotations. Dans tous les cas, des masques de segmentation précis sont obtenus, malgré la large variation d'apparence inhérente aux catégories considérées. Il est à noter que pour la base Pascal VOC 2006 les segmentations sont considérées comme des problèmes binaires. Nous verrons dans la section 6.4.5 que, sur la base Pascal VOC 2007, les 20 classes d'objets en compétition pour une segmentation simultanée rendent le problème bien plus difficile. En ce qui concerne la base MSRC, même dans le cas d'une segmentation multiclasse, les images sont correctement segmentées. Cependant, les variations d'apparence que présentent les objets sont moindres par rapport à celles des bases Pascal, et le fond est plus facile à modéliser. Une classification indépendante des patchs donne déjà de très bons résultats sur ces images.

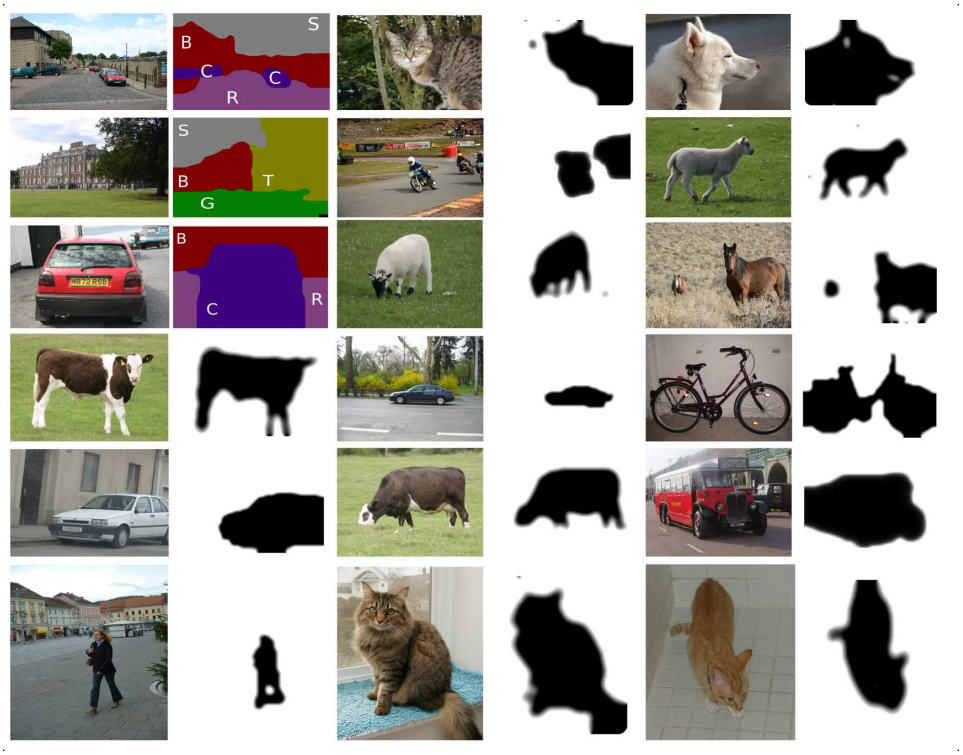


FIG. 6.9 – *Exemples de segmentations obtenues sur les images des bases TU-Graz02, Pascal VOC 2006 et MSRC. Pour cette dernière, l'encodage suivant est utilisé : G pour l'herbe, Sh pour les moutons, S pour le ciel, B pour les bâtiments, T pour les arbres, et C pour les voitures*

	Vache	Mouton	Avion	Visage	Voiture	Vélo	Panneau	Oiseau	Chaise	Chat	Chien	Corps	Personne
Textonboost [80]	58	50	60	74	63	**75**	35	19	15	**54**	19	**62**	7
MFAM [90]	73	**84**	**88**	70	**68**	74	33	19	**34**	46	49	54	**31**
notre méthode	**84**	81	66	**78**	50	62	**36**	**22**	16	43	**52**	30	9

TAB. 6.1 – *Résultats sur les 13 catégories d'objets de la base MSRC.*

6.4.5 Évaluation quantitative

Dans cette section, nous présentons d'autres résultats quantitatifs. Tout d'abord, nous présentons brièvement des résultats obtenus sur la base MSRC et nous présentons des expériences plus poussée sur la base Pascal VOC 2007.

Base MSRC

De part sa popularité, nous avons considéré la base MSRC (voir la description de la base dans la section 1.3.8). Notons que la tâche à résoudre est différente du problème que nous avons considéré dans notre approche. En effet, le fond est divisé en un certain nombre de classes (herbe, bâtiment, arbres, etc.) et le but n'est pas de segmenter les objets du fond, mais de segmenter complètement la scène. La table 6.1 présente les performances de notre algorithme sur les 13 classes d'objets et les compare aux résultats obtenus par la méthode Textonboost [80] et à ceux de la méthode *Markov Field Aspect Model* (MFAM) [90]. Notre méthode obtient des résultats comparables, bien qu'elle n'ait pas été conçue explicitement pour cette tâche.

base Pascal VOC 2007

La base Pascal VOC 2007 (voir la description dans la section 1.3.7) est utilisée pour des compétitions de catégorisation, de détection et, pour 2007, de segmentation d'images. Elle permet de se comparer aux dernières méthodes proposées.

La compétition de segmentation demande de générer des segmentations au niveau du pixel, le label de chaque pixel doit être prédit comme appartenant à une des classes considérées ou au fond, ce qui est exactement la tâche que nous considérons dans ce chapitre. Comme présenté dans [15], 20 classes d'objets et la classe de fond sont considérées. Nous disposons de plus de 5000 images d'apprentissage, dont seulement 422 sont segmentées avec précision, les autres ne possèdent qu'un marquage des objets à l'aide de boîtes englobantes.

Les expériences ont été réalisées selon le protocole de la compétition. La moyenne des précisions obtenues pour chaque classe d'objets et pour le fond est calculée. La précision de la segmentation d'une classe est définie comme le nombre de pixels de cette classe correctement labellisés divisé par le nombre réel de pixels de cette classe [15].

Les images sont représentées en utilisant un vocabulaire SIFT de 10000 mots (créé à l'aide de la méthode proposée dans le chapitre 2) et un vocabulaire couleur de 200 mots.

FIG. 6.10 – *Exemples d'annotations supplémentaires (masque de segmentation) produites automatiquement sur les images d'apprentissage non segmentées, en appliquant notre algorithme sur les boîtes englobantes.*

Le modèle a été entraîné en utilisant toutes les annotations disponibles, c'est-à-dire les masques de segmentations disponibles et les boîtes englobantes. L'apprentissage se fait en deux étapes afin de tirer parti des différentes formes d'annotations disponibles. Tout d'abord, un modèle initial des catégories est appris à partir des images segmentées. Ensuite, ce modèle des classes est utilisé dans notre modèle de segmentation pour segmenter les images d'apprentissages restantes. Durant cette estimation, les boîtes englobantes sont utilisées pour fixer le nombre, la position, et la classe des blobs (chaque boîte représente un blob). Ces paramètres étant fixés, on estime la meilleure affectation des patchs aux blobs et les modèles de couleur pour chaque image. Les segmentations ainsi produites fournissent toute une nouvelle série d'annotations plus précises que les boîtes, qui sont utilisées pour réestimer un modèle d'apparence des classes plus précis. Nous avons expérimentalement confirmé que les nouveaux masques ainsi produits sont fiables, des exemples de nouvelles annotations obtenues sont illustrés dans la figure 6.10.

Lors de la segmentation des images, le nombre de classes d'objets présentes dans une image n'est pas connu. Avec le grand nombre de classes possibles, nous avons observé (voir les résultats plus bas) que le fait d'initialiser notre algorithme avec des prédictions locales basées sur les patchs, comme cela a été fait jusqu'à présent, ne donne pas des résultats satisfaisants. Nous avons donc utilisé un détecteur d'objets basé sur une représentation globale de l'objet, et avons observé que cela améliore significativement la précision des segmentations produites. Plus précisément, nous avons utilisé le détecteur INRIA_PlusClass [15] pour initialiser le nombre, la position et le label des blobs. Ce détecteur est basé sur une approche par fenêtres glissantes utilisant un classifieur SVM linéaire, et des descripteurs d'images basés sur la représentation HOG [14], un descripteur composé d'histogrammes de gradients. Dans les résultats reportés dans la table 6.2, DI représente l'utilisation d'un détecteur pour l'initialisation, et NI l'initialisation naïve, basée sur la prédiction des patchs.

En parallèle de ces deux types d'initialisations, nous avons aussi évalué l'apport des segmentations produites sur les images d'apprentissage non segmentées sur les résultats de segmentation des images de test. Nous avons comparé notre méthode entraînée sur les 422 images d'apprentissage, et notée ST, avec notre méthode entraînée sur l'ensemble complet (constitué des plus de 5000 images d'apprentissage) et utilisant les masques de segmentation additionnels, générés par notre algorithme, notée FT.

	fond	avion	vélo	oiseau	bateau	bouteille	bus	voiture
FT+DI	49.36	20.5	70.36	23.50	16.53	28.72	22.69	58.38
ST+DI	57.23	13.63	35.10	19.60	10.60	23.75	16.78	56.82
FT+NI	14.97	17.68	9.42	1.56	15.85	4.76	10.2	25.10
ST+NI	20.97	11.67	10.02	3.57	15.45	8.65	10.67	17.39
Brookes	77.7	5.5	0	0.4	0.4	0	8.6	5.2
TKK	22.9	18.8	20.7	5.2	16.1	3.1	1.2	78.3

	chat	chaise	vache	table	chien	cheval	moto	personne
FT+DI	65.5	28.17	10.41	0.92	3.7	65.4	51.75	60.1
ST+DI	63.08	24.98	10.58	0.64	4.04	41.15	55.34	64.08
FT+NI	15.19	23.79	7.46	10.61	20.69	15.72	21.89	27.59
ST+NI	7.35	21.18	7.81	5.82	15.71	14.29	11.33	40.54
Brookes	9.6	1.4	1.7	10.6	0.3	5.9	6.1	28.8
TKK	1.1	2.5	0.8	23.4	69.4	44.4	42.1	0

	plante	mouton	sofa	train	écran	**moyenne**
FT+DI	22.02	23.71	27.93	65.20	65.46	**37.16**
ST+DI	14.37	17.83	24.13	46.21	59.72	**31.41**
FT+NI	38.01	8.88	4.24	4.94	17.46	**15.05**
ST+NI	3.42	8.52	8.66	3.93	18.09	**12.62**
Brooks	2.3	2.3	0.3	10.6	0.7	**8.5**
TKK	64.7	30.2	34.6	89.3	70.6	**30.4**

TAB. 6.2 – *Résultats obtenus sur la base Pascal VOC 2007. Les 4 premières lignes donnent les résultats obtenus avec notre méthode en utilisant l'initialisation naïve (NI) ou l'initialisation utilisant le détecteur (DI) et le petit ensemble d'apprentissage (ST) ou le grand ensemble d'apprentissage (FT). Les deux dernières colonnes donnent les meilleurs résultats parmi les méthodes de segmentation soumises et les méthodes de détection respectivement.*

De cette façon, nous avons 4 combinaisons possibles à évaluer. Les résultats obtenus sur les 20 classes de la base Pascal VOC 2007 sont donnés dans la table 6.2. Nous avons également reporté les meilleurs résultats de segmentation soumis à la compétition (notés Brooks dans la table), ainsi que les meilleurs résultats obtenus par les algorithmes de détection, où les résultats produits sont réduits à de simples boîtes englobantes (notés TKK).

Au vu de ce tableau, trois conclusions principales peuvent être tirées.

Premièrement, pour presque toutes les classes, une amélioration notable des résultats est obtenue lorsque l'ensemble des images d'apprentissage est utilisé.

Deuxièmement, les résultats démontrent l'importance d'une bonne initialisation, utilisant le détecteur d'objets. Avec celui-ci, une amélioration de près de 15 % est obtenue sur la précision moyenne. Cela peut être expliqué par le très grand nombre de classes d'objets considérées pour la tâche de segmentation. L'algorithme de détection propose des candidats pertinents qui sont ensuite validés et raffinés par l'algorithme de segmentation. Pour certaines classes, comme les *tables* ou les *chiens*, les résultats sont meilleurs avec l'initialisation naïve. Pour ces classes, le détecteur échoue souvent.

Troisièmement, nous surpassons les meilleures méthodes présentes dans cette compétition.

De façon à avoir une meilleure compréhension du rôle du détecteur, la figure 6.11 illustre le comportement du modèle sur certaines images. A partir des résultats du détecteur, la méthode de segmentation valide les hypothèses

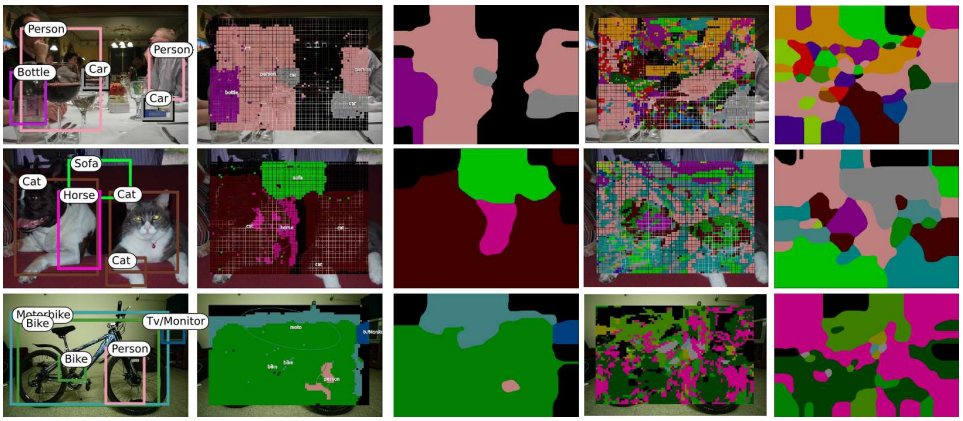


FIG. 6.11 – *Trois images exemples appartenant à la base Pascal VOC 2007. De gauche à droite : (i) l'image originale avec les résultats de la détection surimposés, (ii) l'affectation aux classes après quelques itérations, (iii) la segmentation finale produite à partir de cette initialisation, (iv) les labels des classes à partir de l'initialisation au niveau du patch, (v) le résultat final obtenu à partir de cette initialisation.*

d'objets et raffine leurs frontières. Pour la troisième image, on peut voir clairement que l'algorithme de détection s'est déclenché pour la classe *vélo* et la classe *moto*. Face à ces hypothèses en compétition, l'algorithme de segmentation a choisi le vélo. Nous pouvons aussi voir que des fausses détections évidentes, comme la personne de la troisième image, sont presque complètement supprimées.

Il est intéressant de mesurer la quantité d'images complètement segmentées nécessaire pour obtenir de bons résultats lors de la segmentation des images d'apprentissage additionnelles. Dans les expériences de la table 6.3, nous considérons un sous-ensemble de l'ensemble d'apprentissage réduit et annoté de 422 images, dont le nombre d'images peut être de 50, 166 ou 422. Ce sous-ensemble initial est ensuite utilisé pour apprendre un premier modèle d'apparence des objets (exactement comme précédemment) et ainsi segmenter des images d'apprentissage supplémentaires. On peut segmenter soit uniquement l'ensemble d'apprentissage *train* constitué de 2501 images, soit l'ensemble total d'apprentissage *train* + *val* qui constitue 5011 images. Lorsque l'ensemble de validation est considéré, nous avons comparé un premier cas, où le modèle d'apparence reste le modèle initial constitué du petit nombre d'images, et un deuxième cas, où le modèle d'apparence est réestimé après segmentation de toutes les images de l'ensemble *train*, et ce nouveau modèle « amélioré » est utilisé pour segmenter les images de l'ensemble *val*. Chaque mise-à-jour du modèle appliquée est notée dans le tableau en cochant de la case upd. De même, une croix dans la case T (resp. V) indique que l'ensemble d'apprentissage *train* (resp. *val*) est utilisé. Les résultats de cette table indiquent que l'augmentation du nombre d'images d'apprentissage améliore la qualité du modèle appris et augmente la précision obtenue. De même, le modèle appris sur un nombre plus élévé d'images (deux ensembles *train* et *val* considérés) est toujours meilleur que celui appris à partir de la moitié des images (seulement *train*). Cela indique que l'ajout d'exemples d'apparences possibles pour les catégories permet d'enrichir le modèle et de mieux

généraliser à de nouvelles instances. Enfin, remarquons que la mise-à-jour du modèle d'apparence après segmentation d'un nombre non négligeable de nouvelles images n'a pas eu d'influence significative sur les résultats finaux de classification.

Conclusion

La segmentation est souvent considérée comme un problème isolé. Sans aucune connaissance extérieure, l'image doit être segmentée de façon plausible. Ces méthodes sont bien évidemment limitées par leur manque d'interprétation du contenu. Notre méthode propose des modèles d'apparence des catégories d'objets relativement simples mais qui donnent l'information nécessaire pour guider le processus de segmentation.

La méthode par sac-de-mots utilisée à la base du modèle d'apparence permet d'être intrinsèquement robuste aux occultations, ce qui est impossible avec les modèles de forme rigides. La robustesse aux changements d'illumination est permise par les descripteurs de patchs employés [54, 88]. L'utilisation des blobs pour modéliser les différentes instances d'objets permet de s'adapter à toutes les positions et toutes les échelles. Nous avons volontairement choisi de ne pas utiliser d'a priori sur la position et l'échelle des objets pour chaque catégorie. Ainsi, le modèle est plus général et n'est pas mis en défaut par un positionnement inhabituel des objets dans la scène. Ce modèle flexible basé sur les blobs rend la reconnaissance robuste aux changements d'apparence liés aux variations de points de vue, et aux variations intra-classe. Il n'y a pas de géométrie rigide associée aux modèles de blobs, ce qui permet de segmenter des classes difficiles et non rigides comme les chats ou les personnes. En contrepartie, le modèle pourra plus facilement produire des faux positifs, qui pourraient être évités par un modèle d'objet rigide, ayant précisément appris la vue correspondante de l'objet.

Le modèle est capable de segmenter différentes instances de la même catégorie d'objets, en différents blobs. Cela peut paraître superflu lorsque le but final est de prédire la classe de chaque pixel, et non d'identifier les différentes instances de chaque catégorie. Cependant, il peut s'avérer bénéfique de modéliser séparément les instances puisque cela permet d'estimer plus précisément les modèles d'apparences spécifiques aux instances (la couleur dans notre cas). Ainsi, chaque objet peut être segmenté avec plus de précision, entraînant une meilleure segmentation de l'image.

Les modèles d'objets peuvent utiliser avantageusement des annotations précises à base de masques de segmentation. Lorsque seules des boîtes englobantes sont disponibles, les contours d'images et les discontinuités entre les régions homogènes sont utilisés pour compenser la faiblesse des annotations. Nous supposons également que les différentes images d'apprentissage présentent des objets suffisamment similaires entre eux et des fonds suffisamment variés pour apprendre les modèles d'objets.

Nous avons aussi étudié dans les expériences comment combiner les deux types d'annotations. La conclusion de ces expériences est qu'il est plus important d'avoir un très grand nombre d'images d'apprentissage, même si les annotations sont moins précises. Les modèles appris sont alors beaucoup plus robustes aux variations d'apparence.

Nombre d'images	T	V	upd	fond	avion	vélo	oiseau	bateau	bouteille	bus	voiture
FT+DI	x	x	x	49.36	20.5	70.36	23.50	16.53	28.72	22.69	58.38
422	x	x	-	50.13	20.44	69.78	24.25	16.48	28.74	20.85	58.38
422	x	x	x	49.95	20.44	69.51	22.49	16.13	29.17	21.54	57.96
422	x	-	-	50.23	20.29	69.98	21.86	16.50	27.56	21.25	58.37
166	x	x	-	50.03	20.35	69.62	21.87	16.12	28.88	21.44	57.97
166	x	x	x	50.01	20.21	69.45	22.10	16.14	28.85	21.61	57.98
166	x	-	-	50.25	20.32	70.01	21.68	13.16	27.56	21.74	58.28
50	x	x	-	49.94	20.16	69.61	21.67	16.36	28.80	21.20	57.98
50	x	x	x	50.00	20.35	69.67	22.02	16.23	28.82	21.34	57.83
50	x	-	-	50.26	20.25	69.98	22.09	16.05	27.55	21.37	58.39
Brookes				77.7	5.5	0	0.4	0.4	0	8.6	5.2
TKK				22.9	18.8	20.7	5.2	16.1	3.1	1.2	78.3

				chat	chaise	vache	table	chien	cheval	moto	personne
FT+DI	x	x	x	65.5	28.17	10.41	0.92	3.7	65.4	51.75	60.1
422	x	x	-	65.43	29.18	10.34	1.31	3.65	64.22	50.74	59.22
422	x	x	x	65.59	27.89	10.46	1.25	3.78	64.58	51.25	58.72
422	x	-	-	65.04	27.78	10.40	0.80	3.39	65.78	50.13	58.69
166	x	x	-	65.48	27.44	10.56	1.47	3.56	65.59	50.96	58.44
166	x	x	x	65.59	27.79	10.61	1.54	3.70	65.60	50.87	58.68
166	x	-	-	65.15	27.96	10.35	0.73	3.47	65.71	50.06	58.78
50	x	x	-	65.34	28.04	10.61	1.57	3.73	64.56	50.74	58.24
50	x	x	x	65.57	27.82	10.48	1.23	3.57	64.53	50.48	58.46
50	x	-	-	65.11	28.12	10.35	0.71	3.39	65.66	50.08	58.97
Brookes				9.6	1.4	1.7	10.6	0.3	5.9	6.1	28.8
TKK				1.1	2.5	0.8	23.4	69.4	44.4	42.1	0

				plante	mouton	sofa	train	écran	**moyenne**
FT+DI	x	x	x	22.02	23.71	27.93	65.20	65.46	**37.16**
422	x	x	-	22.54	23.31	28.50	64.65	66.30	**37.01**
422	x	x	x	22.12	23.46	28.22	64.27	65.92	**36.89**
422	x	-	-	22.26	23.30	26.70	64.31	65.94	**36.69**
166	x	x	-	22.90	23.42	28.29	64.03	66.13	**36.88**
166	x	x	x	22.96	23.50	28.46	64.38	66.08	**36.96**
166	x	-	-	22.26	23.52	26.90	65.01	65.64	**36.74**
50	x	x	-	22.96	23.73	28.24	64.59	65.95	**36.86**
50	x	x	x	22.73	23.61	28.30	64.64	66.06	**36.84**
50	x	-	-	22.29	23.39	26.92	64.76	65.67	**36.73**
Brooks				2.3	2.3	0.3	10.6	0.7	**8.5**
TKK				64.7	30.2	34.6	89.3	70.6	**30.4**

TAB. 6.3 – *Expérience complémentaire d'étude du rôle des segmentations initiales dans la génération des annotations supplémentaires.*

7 Détection d'objets appliquée à la recherche visuelle dans le cadre d'une application robotique

Sommaire

Les travaux présentés dans ce chapitre proposent une application du modèle présenté dans le chapitre précédent dans un contexte robotique. Le modèle de localisation des objets est utilisé pour produire des hypothèses sur la position d'un objet dans un environnement inconnu.


Copyright © 2002 Yutaka Izubuchi & AIST

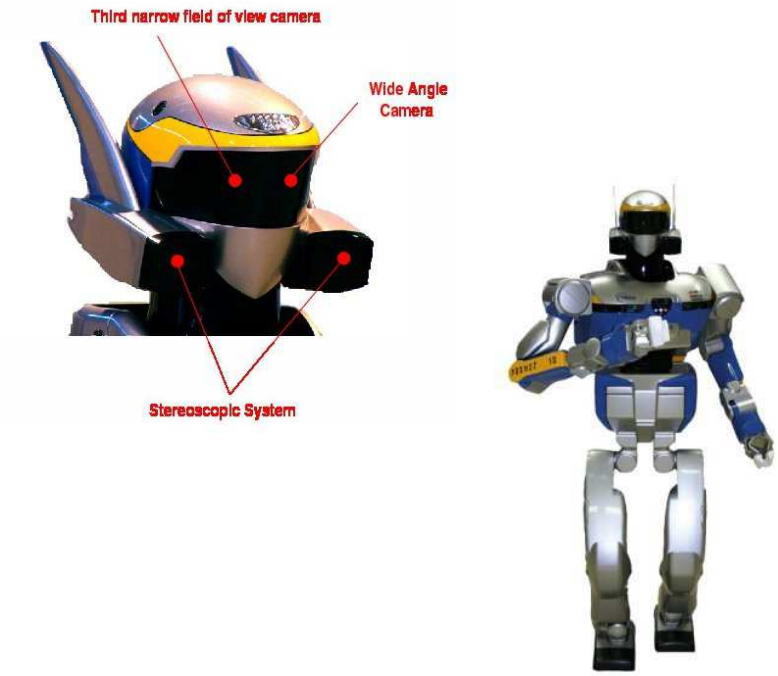


FIG. 7.1 – *Gauche : le système de vision du robot humanoïde HRP-2 (Droite).*

7.1 Introduction

Le travail présenté dans ce chapitre a été réalisé en collaboration avec le laboratoire franco-japonais, le *Joint Robotics Laboratory (CNRS, AIST)* de Tsukuba au Japon. Il correspond à l'application des travaux présentés dans le précédent chapitre à un problème robotique particulier, le problème de la chasse au trésor. Une plateforme spécifique a été considérée pour l'étude : le robot humanoïde HRP-2.

7.1.1 La plateforme HRP-2

Le robot humanoïde HRP-2 est la plateforme technologique centrale du JRL. Né au Japon en 2003, HRP-2[1] pèse 58 kg et mesure 1.54 m. Il doit son nom au projet *Humanoid Robotics Project*, un grand programme de recherche en robotique conduit à l'initiative du ministère japonais de l'économie, du commerce et de l'industrie (METI). Il existe actuellement 14 plateformes de robotique humanoïde de type HRP-2 dans le monde dont 13 sont au Japon.

HRP-2 est équipé de capteurs d'effort pour la gestion de son équilibre, la planification et le contrôle de ses actions mais aussi de caméras pour la vision.

Le système de vision du robot disponible au JRL a été modifié et possède 3 caméras avec un grand angle de vue (25 degrés) et une quatrième avec un large angle de vue (90 degrés). Cette dernière est utilisée pour le SLAM (*Simultaneous localization and mapping*) [83], tandis que les trois autres sont utilisées pour la reconnaissance d'objets.

[1] aussi appelé *Promet* par le dessinateur de manga Yutaka Izubuchi qui a réalisé son design

7.1.2 Le projet de la chasse au trésor

Ce travail fait partie d'un projet nommé *Treasure Hunting*, ou chasse au trésor. Le scénario auquel il s'applique est le suivant : un objet est montré à un robot pour que celui-ci en construise une représentation interne. Ensuite, l'objet est placé dans un environnement inconnu, et le robot doit le retrouver en complète autonomie.

Ce comportement est une question fondamentale des systèmes complexes de robots autonomes, afin de pouvoir aider les humains dans divers environnements de travail. Notre recherche s'intéresse à ce problème de façon intégrée, c'est à dire qu'il considère les capacités et les limitations du robot HRP-2.

L'implémentation complète de ce comportement nécessite de résoudre différents problèmes :

1. la recherche visuelle dans un environnement inconnu,
2. la génération des mouvements pour l'acquisition des images nécessaires à l'apprentissage du modèle d'objet,
3. la reconnaissance visuelle à une distance quelconque pour guider la recherche visuelle,
4. l'estimation de la pose, lorsque l'objet a été trouvé et est suffisamment proche.

Le comportement de recherche visuelle a été décrit en détail par Saidi *et al*[74]. Le robot y construit incrémentalement un modèle de son environnement, en supposant donné un module capable de reconnaître l'objet.

Une génération de postures dévouée au problème de l'acquisition de données d'apprentissage a été proposée. Son rôle est de fournir au module de reconnaissance les images nécessaires pour construire la représentation interne d'un objet donné. La génération de postures est optimisée pour prendre en compte les contraintes robotiques, tout en étant guidée par les exigences de la construction de représentation d'objets. Les détails de cette génération de mouvements sont proposés dans [84].

L'objet considéré est bien texturé et possède une géométrie complexe. Par conséquent, nous ne pouvons utiliser les méthodes 3D basées sur les arrêtes, ni une quelconque modélisation 3D manuelle de l'objet. Dans la suite de ce chapitre, nous nous intéresserons uniquement à la détection d'objets dans des conditions difficiles. Cela sera l'occasion d'appliquer le modèle présenté dans le chapitre précédent. Les autres composants sont décrits dans [84].

7.1.3 Quelques références

Si nous nous affranchissons de la recherche visuelle proprement dite, la question qui nous intéresse est extrêmement liée au problème de la détection d'objets. La détection d'objets a reçu énormément d'attention lors de ces dernières années, et le but n'est pas ici de faire une liste exhaustive des méthodes proposées. Notons cependant qu'un grand nombre de celles-ci utilisent des représentations globales à travers un descripteur de l'objet tout entier. Citons par exemple [93] ou encore [14]. Ces méthodes sont connues pour être très efficaces, mais nécessitent un grand nombre d'images d'apprentissage, et une très longue étape d'apprentissage. Elles supposent également que l'apparence de l'objet ne varie que faiblement, et la détection de plusieurs vues

simultanément nécessite en général d'entraîner plusieurs classifieurs, propres à chaque vue de l'objet. Alternativement des méthodes s'intéressent à des représentations locales des images, pouvant constituer des parties de l'objet. Dans [54], des points d'intérêts sont mis en correspondance. Ces méthodes fonctionnent bien lorsque suffisamment de mises en correspondance sont possibles entre des parties d'objet, et c'est en effet une méthode très similaire qui est utilisée pour estimer la pose à la fin de la recherche visuelle dans le projet de chasse au trésor [84]. Cependant, lorsque les objets sont trop petits, les points d'intérêt sont principalement sur le fond, et peu d'appariements apparaissent entre les objets. Ces méthodes semblent donc mal adaptées à notre problème. Certaines méthodes procèdent à une recherche exhaustive de l'image en position et en échelle [14, 93]. D'autres s'appuient sur les détections locales pour construire des hypothèses sur la position de l'objet [54, 51]. Nous avons vu que les hypothèses locales, en trop petit nombre, risquaient de manquer les objets.

La méthode proposée extrait des patchs densément dans l'image et ne dépend pas de points d'intérêt. De plus l'objet est texturé ce qui rend la classification à l'échelle du patch, représentant une partie d'objet, relativement fiable. Ce qui est encore renforcé par le fait que la classification demandée est à l'échelle de l'instance d'objet et non de la catégorie. Cette classification combinée aux contraintes locales et semi-globales ajoutées par notre modèle permet une bonne localisation de l'objet. Le choix de notre méthode semble donc adapté.

La section 7.2 propose une version étendue du modèle précédent afin d'utiliser les informations supplémentaires permises par la plateforme robotique. La section 7.3 présente une étude expérimentale du modèle proposée. Enfin, nous conclurons ce chapitre.

7.2 Modèle utilisé pour la détection

Le modèle proposé ici est une extension du modèle du chapitre 6 auquel des informations additionnelles ont été ajoutées, spécifiques à la plateforme expérimentale utilisée. En particulier, nous utilisons deux images de la scène à chaque fois, fournies par les caméras à faible angle de vue, situées à gauche et à droite de la tête du robot. Ces deux images sont ensuite utilisées pour calculer une carte de disparité, et l'estimation de la profondeur est considérée comme un composant du modèle.

Comme le modèle précédent, nous allons considérer deux composants :

- ▷ un modèle génératif basé sur les mots visuels, pour ses bonnes propriétés de localisation
- ▷ un champ de Markov (MRF) qui fournit un champ cohérent de labels qui suivent les frontières de l'objet, localisées grâce aux contours dans l'image 2D et aux discontinuités en profondeur.

7.2.1 Caractéristiques visuelles

Comme précédemment, les images sont représentées par un ensemble de n patchs qui se chevauchent, et par une mesure de connectivité entre deux patchs voisins (voir figure 7.2).

Les patchs, notés $\mathcal{P}_i, i \in \{1, \ldots, n\}$, sont extraits de façon régulière, et relativement dense, de sorte qu'ils se chevauchent fortement. Cinq caractéristiques différentes sont calculées à partir de chaque patch $\mathcal{P}_i$: les mêmes quatre caractéristiques que celles du chapitre 6, et une nouvelle cinquième. Pour rappel, les quatre caractéristiques identiques au chapitre précédent sont le mot visuel SIFT [54] noté w_i^{sift}, le mot visuel couleur [88] w_i^{color}, une valeur RGB obtenue à partir des pixels du centre du patch, ainsi que la coordonnée dans l'image $X_i = (x_i, y_i)$. La nouvelle caractéristique est l'estimation de la profondeur d_i, obtenue à partir de la carte de disparité.

La mesure de connectivité est calculée pour les paires de patchs voisins $\mathcal{P}_i$ et $\mathcal{P}_j$. Cette mesure est composée de deux parties. Tout d'abord, elle inclut les informations provenant de la carte de discontinuité, de la même façon que dans le chapitre précédent. Notée $\mathcal{G}$, cette carte donne en chaque pixel la probabilité de trouver un segment de l'image. L'extraction de ces contours peut se faire de différentes manière. Nous avons continué d'utiliser l'algorithme [57], mais un algorithme plus rapide pourrait être envisagé. À partir de la carte, la valeur $\Phi_{i,j}$ est calculée, comme la valeur maximale prise par $\mathcal{G}$ entre les centres de patchs $\mathcal{P}_i$ et $\mathcal{P}_j$. À cette valeur est ajoutée la discontinuité en terme de profondeur entre les deux patchs, $\Psi_{i,j} = |d_i - d_j|$. Au final la connectivité est définie par :

$$C_{i,j}(\mathcal{P}_i, \mathcal{P}_j) = \exp(-\beta(\Phi_{i,j} + \Psi_{i,j})), \tag{7.1}$$

où β est l'inverse de la moyenne des valeurs prises par $\Phi_{i,j} + \Psi_{i,j}$ dans l'image. Ainsi, la connectivité entre deux patchs voisins sera d'autant plus faible qu'un contour les sépare, ou qu'ils présentent des valeurs de profondeur très différentes.

7.2.2 Un modèle génératif de blobs

Exactement comme pour le chapitre précédent, nous considérons qu'une image est constituée de régions de formes elliptiques, les *blobs*, et que chaque blob est responsable de la génération des observations associées aux patchs de sa région selon un modèle qui lui est propre et qui dépend de sa catégorie.

Ainsi, la génération d'un patch nécessite de réaliser successivement les opérations suivantes :

- ▷ sélectionner un blob,
- ▷ générer un patch à partir du modèle spécifique à ce blob.

Encore une fois, la génération des blobs est supposée suivre un processus de Dirichlet, chaque patch nouvellement généré peut soit appartenir à un blob d'image existant B_k - avec une probabilité $\frac{N_k}{n-1+\alpha}$ où N_k est le nombre d'échantillons déjà générés par ce blob, et n est le nombre total d'échantillons - ou bien être échantillonné par un nouveau blob - avec une probabilité $\frac{\alpha}{n-1+\alpha}$, où α est le paramètre de concentration du processus de Dirichlet.

Chaque blob $B_k, 1 \leq k \leq K$ est caractérisé par le même ensemble de variables aléatoires que précédemment, plus une information d'échelle. $\Theta_k = \{\mu_k, \Sigma_k, C_k, l_k, S_k\}$. μ_k, Σ_k sont respectivement la moyenne et la matrice de covariance qui décrivent la forme du blob, C_k est un modèle de mixtures gaussiennes qui représente le modèle de couleur du blob dans l'espace RGB, l_k est le label du blob (la catégorie de l'objet qu'il représente), et S_k est

l'échelle du blob, qui est étroitement liée à la distance entre l'objet et la caméra.

Nous caractérisons chaque patch $\mathcal{P}_i$ par son ensemble de primitives $(w_i^{sift}, w_i^{color}, rgb_i, X_i, d_i)$, ainsi que par l'indice du blob qui l'a généré b_i. La probabilité de générer un patch, sachant qu'il a été généré par le blob B_k de paramètres Θ_k : $p(\mathcal{P}|b_i = k) = p(\mathcal{P}|\Theta_k)$ est composée de 5 parties distinctes, puisque le modèle suppose que la position et l'échelle des patchs, leur couleur et leur apparence sont indépendants pour un blob donné. Nous aboutissons ainsi à une formule identique à l'équation 6.1 pour la probabilité pour un blob B_k de générer un patch $\mathcal{P}_i$ qui est donc, hormis le terme en plus :

$$p(\mathcal{P}_i|\Theta_k) = p(w_i^{sift}|\Theta_k)p(w_i^{color}|\Theta_k)p(rgb_i|\Theta_k)p(X_i|\Theta_k)p(d_i|\Theta_k) \qquad (7.2)$$

Ces probabilités sont définies comme dans le chapitre précédent : la position X_i d'un patch est choisie selon une distribution normale de paramètres μ_k et Σ_k pour les blobs d'objets. Elle est uniforme pour les blobs de fond. Nous supposons que les blobs d'objets et de fond ont un modèle de couleur qui est une mixture de gaussiennes. La profondeur dépend de la taille du blob. Enfin, les probabilités des mots visuels SIFT et la couleur dépendent uniquement du label de classe. Elles sont apprises pendant la phase d'apprentissage selon la démarche décrite dans la section 7.2.5.

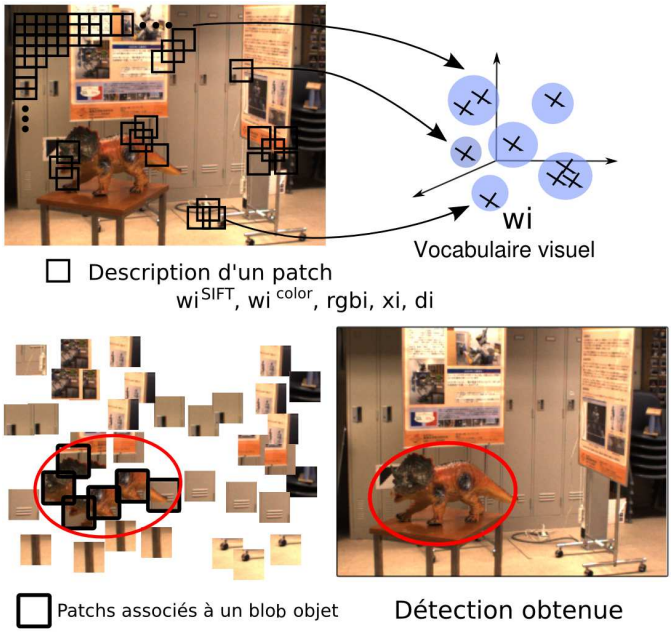


FIG. 7.2 – *Première ligne : les patchs sont extraits densément dans les images. Chaque patch est associé au mot visuel le plus proche pour ses descripteurs SIFT et couleur, et ensuite représenté par les index des mots w_i^{SIFT}, w_i^{color}, une valeur RGB rgb_i, une position x_i et une profondeur d_i, fournie par la carte de disparité. Deuxième ligne : le modèle calcule les meilleures affectations des patchs aux blobs d'objets et de fond, et estime la position et l'échelle des blobs.*

7.2.3 Une structure MRF d'affectations aux blobs

Tout comme dans le chapitre précédent, notre modèle intègre un deuxième composant, un champ de Markov qui s'intéresse aux affectations des patchs aux blobs. Rappelons que son but est de régulariser ces affectations entre patchs voisins de façon à aligner le découpage entre les objets et le fond (et entre les objets eux-mêmes) en s'appuyant sur les contrastes naturels de l'image et, ce qui est une extension du chapitre, sur les forts changements de profondeur. Ce champ est défini sur une grille de connectivité 8, dont les nœuds correspondent aux centres de patchs.

L'unification entre le modèle génératif et le champ de Markov se fait encore une fois en combinant l'a priori sur les champs de labels donnés par le modèle de Dirichlet avec un a priori provenant du MRF. Ainsi, la probabilité obtenue est la suivante :

$$p(\mathcal{P}, b|\Theta) \propto p_{dir}(b)p_{mrf}(b|\Theta)p(\mathcal{P}|b,\Theta). \tag{7.3}$$

qui est reformulée de façon énergétique dans l'équation que nous rappelons ici :

$$E(\mathcal{P}, b) = U(\mathcal{P}, b) + \gamma \sum_{i,j\in\mathcal{C}} V_{i,j}(b_i, b_j), \tag{7.4}$$

où $\mathcal{C}$ représente l'ensemble des voisins dans la grille de connectivité 8, γ est un paramètre qui équilibre les deux termes de la formule, et

$$U(\mathcal{P}, b) = -\log(p(\mathcal{P}|b,\Theta)p_{dir}(b)). \tag{7.5}$$

Le modèle du MRF p_{mrf} est représenté par le second terme de l'équation 7.4, et le potentiel sur les paires de voisins est défini à l'aide de la mesure de connectivité que nous avons introduite section 7.2.1, dans l'équation 7.1 par

$$V_{i,j}(b_i, b_j) = [l_{b_i} \neq l_{b_j}]C_{i,j}, \tag{7.6}$$

où [.] est la fonction indicatrice.

Le potentiel $V_{i,j}$ diffère légèrement de celui du chapitre précédent, puisqu'il prend en compte l'information de profondeur. Il agit toujours comme un terme de pénalité qui vaut 0 si deux patchs voisins ont le même label (*i.e.* sont générés par le même blob), et dont la valeur est d'autant plus grande que la connectivité est forte entre les patchs sinon. Il renforce ainsi la cohérence entre les labels de patchs voisins, et relâche les contraintes de cohérence lorsqu'un contour ou un changement de profondeur intervient.

7.2.4 Estimation du modèle

Maintenant que le modèle a été défini, il faut estimer ses paramètres pour chacune des images fournies par l'algorithme de recherche visuelle. C'est-à-dire estimer le nombre et la position des blobs, produire des labels objet/fond pour ces blobs (l_k), et estimer les affectations des patchs à ces blobs (b_i). L'estimation se déroule de la même manière que dans le chapitre 6, les détails de l'estimation peuvent être trouvés section 6.2.4

7.2.5 Apprentissage de l'apparence d'un objet

Pour apprendre l'apparence de l'objet à reconnaître, des exemples d'images contenant cet objet sont acquise par le robot, en utilisant la vision stéréoscopique. La carte de disparité dense qui en résulte fournit des informations locales qui sont utilisées pour créer des masques de segmentation sur les images positives. En effet, seules les parties texturées (majoritairement l'objet) sont mises en correspondance correctement. Les segmentations obtenues ne sont pas parfaites, mais le modèle appris est suffisamment robuste pour apprendre l'apparence de l'objet.

Nous avons également utilisé des images négatives (*i.e.* ne contenant pas l'objet) fournies par les caméras du robot lorsque celui-ci se déplace dans un environnement ne contenant pas l'objet. Ainsi un modèle du fond peut être appris.

Le reste du processus est identique au chapitre précédent. Les descripteurs SIFT et couleurs sont extraits de toutes ces images d'apprentissage, de la même façon que pour les images de test. Ces descripteurs sont utilisés tout d'abord pour créer les mots visuels à l'aide d'un processus de quantification, puis ils permettent de calculer la probabilité pour chaque mot visuel, d'être observé comme apparaissant dans l'objet ou dans le fond. Ces distributions de probabilité $p(w^{sift}|\Theta_k)$ et $p(w^{color}|\Theta_k)$ sont obtenues par normalisation des histogrammes d'occurrence.

7.3 Expériences

Afin d'évaluer notre méthode de détection dans des conditions réalistes, nous avons créé une base de test. Elle est composée de 118 images, prises avec les caméras du robot. La plupart d'entre elles contiennent l'objet, dans des conditions très diverses. Nous avons fait varier la distance à l'objet, le point de vue, l'illumination ; différents fonds et des occultations sont présents. Le reste des images ne contient pas l'objet.

7.3.1 Mesures de performance utilisées

Notre modèle fournit des régions de forme elliptique dans l'image, qui nous donnent approximativement la position de l'objet. Pour l'évaluation des performances de détection, ces blobs ont été transformés en boîtes englobantes décrivant des régions rectangulaires de l'image, susceptibles de contenir l'objet.

Ces images ont également été annotées à la main, rendant ainsi disponible une vérité terrain. Une détection est considérée comme correcte si l'aire de recouvrement entre la boîte englobante prédite B_p et la boîte englobante de la vérité terrain B_{gt} excède un certain seuil. Nous avons utilisé 50% par défaut. Le recouvrement est calculé par la formule suivante :

$$recouvrement = \frac{aire(B_p \cap B_{gt})}{aire(B_p \cup B_{gt})} \tag{7.7}$$

Les performances de détection sont évaluées en utilisant des valeurs de précision et de rappel. Si N_d désigne le nombre de détections obtenues par notre méthode, N_o le nombre d'objets effectivement présents dans l'ensemble

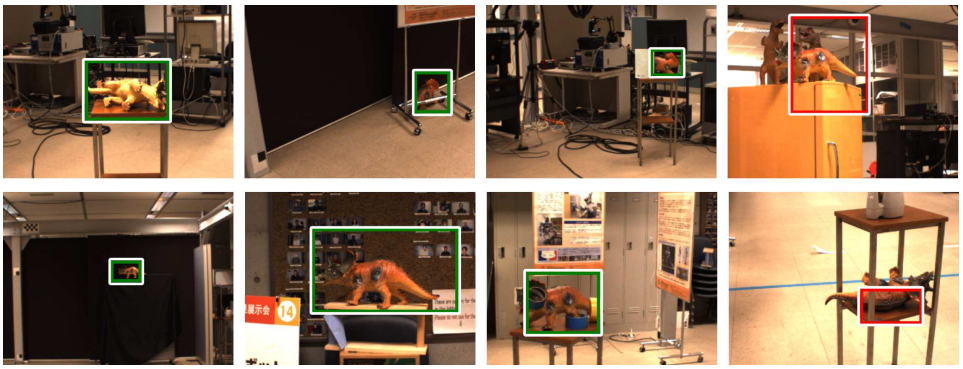


FIG. 7.3 – *Trois premières colonnes : exemples de détections correctes, dernière colonne : exemple de détections erronées.*

d'images de test, et N_c le nombre de détections correctes, alors nous pouvons définir la précision P et le rappel R à l'aide des formules suivantes : $P = \frac{N_c}{N_d}$ et $R = \frac{N_c}{N_o}$. Des détections multiples du même objet sont considérées comme fausses.

7.3.2 Évaluation quantitative de la détection

Dans cette partie, nous cherchons à évaluer le gain de performance permis par l'information de profondeur, à l'aide du score définit précédemment entre les boîtes englobantes prédites et réelles. Cette information est utilisée à deux reprises : tout d'abord en tant que caractéristique du patch, donc dans le modèle génératif, mais aussi dans le MRF où elle intervient dans les contraintes de voisinage.

La table 7.1 montre les valeurs de précision et de rappel pour différents seuils d'acceptation. Les colonnes présentent la précision et le rappel pour notre méthode (gauche) et pour la méthode originale du chapitre 6 (droite), qui n'utilise pas d'information de profondeur.

Tout d'abord, il est clair que les améliorations proposées pour la méthode augmentent les résultats de détection. La carte de disparité et la profondeur associée aident le modèle à estimer plus précisément les frontières de l'objet et donnent un bon indice sur la taille attendue pour l'objet.

À partir des valeurs de précision, nous pouvons voir que dans la plupart des cas (81%) l'objet détecté a au moins une partie en commun avec l'objet original. Ces erreurs de précision sur la localisation peuvent être corrigées grâce à la boucle réalisée par la recherche visuelle : en fonction de ces résultats, une nouvelle image à analyser est proposée.

Enfin, les valeurs de rappel indiquent que 64% des objets ont été détectés au moins partiellement par notre méthode.

La figure 7.3 montre des exemples de détections correctes (chevauchement supérieur à 50%) et de fausses détections (chevauchement inférieur à 50%).

7.3.3 Évaluation qualitative de la segmentation

Le modèle fournit également la liste des patchs qui appartiennent à une instance particulière d'objet. Les patchs correspondent à des ensembles de pixels qui appartiennent à leur support. En utilisant les informations de tous

	avec la profondeur (chap 7)		sans la profondeur (chap 6)	
seuil de tolérance	précision	rappel	précision	rappel
0.2	0.81	0.64	0.65	0.61
0.3	0.73	0.57	0.50	0.48
0.4	0.60	0.48	0.39	0.37
0.5	0.41	0.32	0.26	0.24
0.6	0.15	0.12	0.17	0.16

TAB. 7.1 – *Valeurs de précision et de rappel pour la méthode proposée et pour celle présentée dans le chapitre 6.*

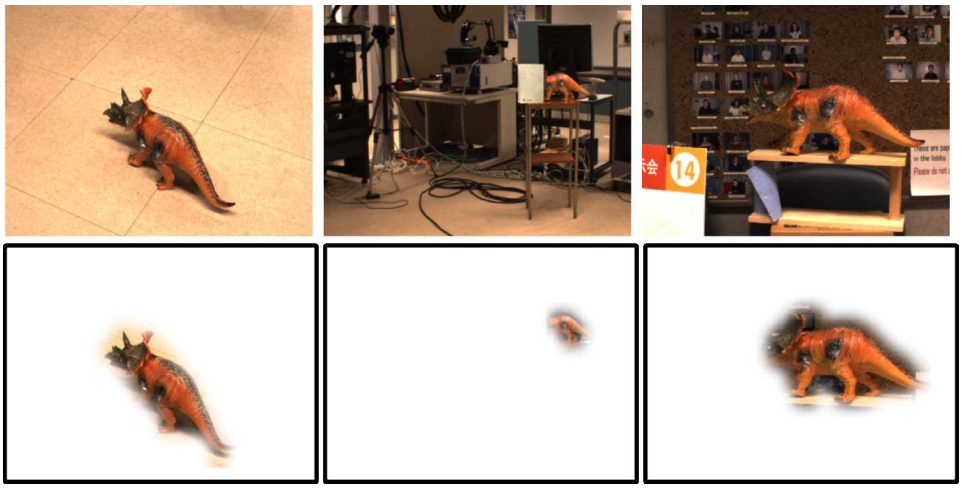


FIG. 7.4 – *Le modèle fournit une liste des patchs qui appartiennent à l'objet. Cela permet de produire des masques de segmentation.*

les patchs contenant un pixel donné, nous pouvons créer une segmentation des objets. Les détails de la création d'une segmentation à partir du modèle ont été donnés dans la section 6.4.2. La figure 7.4 présente des masques de segmentation en terme de cartes de probabilité sur la localisation des objets dans les images, pour les détections qui se sont montrées particulièrement précises.

Conclusion

Dans ce chapitre nous avons présenté une application du modèle exposé dans le chapitre précédent à un problème concret de robotique humanoïde. Nous avions fait l'hypothèse qu'il était possible de classifier localement les patchs de l'image, hypothèse que nous avons vérifiée dans les expériences. La reconnaissance locale au niveau du patch, associée au reste du modèle donne des résultats satisfaisants.

Le problème a été traité de façon intégrée, c'est à dire en tenant compte des spécificités de la plateforme d'application. Ces spécificités nous ont permis d'ajouter des améliorations au modèle, dont nous avons montré qu'elles donnent de meilleurs résultats. Il serait intéressant de réaliser une comparaison approfondie entre cette méthode et d'autres méthodes de détection.

Dans ce chapitre nous avons considéré un seul objet, afin de vérifier les contraintes posées par le problème de chasse au trésor : le modèle de

l'objet est appris en le montrant au robot, et une pose doit être estimée à la fin de la recherche visuelle. Or toutes les techniques employées dans notre méthode s'appliquent pour les catégories d'objets, et non pas uniquement pour une instance particulière. Si on considère uniquement le problème de recherche visuelle dans un environnement inconnu, et que l'on suppose un apprentissage différent, qui permet l'acquisition d'un modèle au niveau de la catégorie (par exemple à l'aide d'un ensemble d'images d'apprentissage, comme cela a été fait pour les autres chapitres de cette thèse), un problème plus complexe pour être résolu, celui de la « chasse à une catégorie ».

Conclusions et perspectives

8

Sommaire

Durant cette thèse, nous nous sommes intéressés au problème de la reconnaissance d'objets dans les images à travers deux tâches particulières : la catégorisation d'images et la segmentation de classe d'objets. Tout au long de notre travail, les images ont été considérées comme des collections de primitives locales, les descripteurs de patchs, représentées par des mots visuels.

8.1 Contributions

Nous rappelons ici les principales contributions de ce manuscrit. Elles sont organisées en deux axes : la création de vocabulaires visuels et l'utilisation de ces vocabulaires pour la localisation des objets dans les images.

8.1.1 Création de vocabulaires visuels

Nous avons proposé, étudié et comparé différentes approches permettant la création des mots visuels.

Quantification vectorielle de descripteurs denses Le chapitre 2 se place dans la situation où les patchs des images sont obtenus par un échantillonnage dense des images. En procédant ainsi, aucune information pertinente n'est ignorée, mais en contrepartie la quantité de données à traiter est beaucoup plus grande, et la densité des descripteurs de patchs dans leur espace de représentation présente de forts déséquilibres. Nous avons vu que les méthodes classiques sont mises en défaut par ces deux facteurs.

Nous avons proposé une méthode de quantification vectorielle, capable de traiter des millions de descripteurs visuels en entrée et de créer des vocabulaires de tailles conséquentes, avec une complexité plus faible que les méthodes de clustering classiques. Composée de deux phases, notre méthode alterne un échantillonnage biaisé des vecteurs (réduction de l'influence des clusters les plus larges), avec une étape de clustering qui ajoute un à un les nouveaux centres de clusters.

Une étude paramétrique complète permet d'évaluer l'importance des différents paramètres de la méthode. Nous avons également obtenu, lorsque nous avons mis cette méthode au point, les meilleurs résultats sur la compétition Pascal VOC 2005.

Production de vocabulaires discriminants Dans le chapitre 3, nous sommes partis de la constatation que l' objectif que nous recherchons avec les vocabulaires visuels n'est pas de représenter fidèlement les images mais de permettre des bonnes performances de classification. Aussi, nous avons cherché à créer des vocabulaires compacts où les mots sont définis de manière à améliorer la discrimination entre les classes.

Cela est rendu possible par l'estimation d'un modèle génératif où les images sont représentées comme des distributions sur des variables d'aspect (les topics) qui eux même sont décris par des distributions sur les mots visuels. L'estimation conjointe des topics et des mots améliore simultanément la précision des topics et la spécialisation des mots visuels.

Nous montrons au moyen de nombreuses expériences que de bonnes performances de classification peuvent être obtenues avec peu de mots visuels, au prix d'une phase d'apprentissage allongée.

Vocabulaires à base de forêts aléatoires Enfin nous avons tenté de concilier ces deux objectifs : prise en compte d'un grand nombre de descripteurs et utilisation des informations de classe disponibles, pour la création du vocabulaire visuel. C'est l'objet de la méthode du chapitre 4.

L'approche que nous avons mise au point est assez différente des méthodes antérieurement proposée pour créer des vocabulaires visuels. Elle s'attache à modéliser les frontières entre les mots plutôt que les mots eux-mêmes. Ces frontières sont définies par les classifieurs unitaires d'un arbre de décision aléatoire, lesquels sont construits en utilisant l'information sur les classes. Nous avons combiné plusieurs arbres pour réduire la variance, ce qui revient à utiliser plusieurs vocabulaires redondants simultanément. Les vecteurs de représentation d'images sont donc beaucoup plus larges que dans les méthodes traditionnelles (nombre total de feuilles sur tous les arbres), mais permettent une bonne généralisation, puisque des modèles d'objets fiables sont appris même avec peu d'images d'apprentissage par classe. Cette généralisation a été vérifiée dans les expériences. La structure arborescente permet un traitement rapide des images de test.

Nous avons vu également que ces arbres de décision sont capables de classifier individuellement les patchs, et avons utilisé cette propriété pour obtenir les segmentations présentées dans le chapitre 6.

8.1.2 Segmentation des objets dans les images.

Nous nous sommes aussi intéressés à la localisation des objets dans les images, ce qui constitue une prédiction plus riche que la seule présence/absence d'un objet dans une image.

Localiser pour mieux échantillonner. Dans le chapitre 4, nous avons défendu l'hypothèse selon laquelle les objets que l'on souhaite reconnaître peuvent être définis comme des régions saillantes des images. Les cartes de saillance construites dans ce chapitre viennent prédire la position et l'échelle des objets appartenant aux catégories d'intérêt. Nous avons utilisé cette information pour accélérer la construction des représentations d'images utilisées pour la classification. En effet, une fois un objet localisé, l'échantillonnage se concentre sur cette partie de l'image. Ainsi, plus d'information provient de zones contenant les objets d'intérêt et la quantité de bruit provenant des images est réduite : nous extrayons directement l'information importante des images.

Segmenter avec des variables d'aspect. Nous nous sommes ensuite intéressés à la localisation des catégories d'objets « au pixel près », ce que nous avons défini comme *la segmentation de classes d'objets*. Rappelons qu'il s'agit de déterminer, pour chaque pixel de l'image, son appartenance à l'une des catégories d'intérêt ou au fond. Pour cela nous nous sommes placés dans le contexte des modèles génératifs.

Un premier modèle (chapitre 5) a été proposé, lequel utilise des variables d'aspects pour décrire non plus les images, mais des régions semi-locales d'images (fenêtres rectangulaires qui recouvrent l'image tout en se chevauchant). Nous avons montré que la régularisation qui en résulte donne de meilleurs résultats que la description des aspects globalement sur l'image. Cependant ce modèle n'utilise ni de contraintes de voisinage entre les patchs voisins, ni de contraintes sur la cohérence de la distribution de couleurs de l'objet, et il n'impose pas que les frontières des objets épousent les contours de l'image.

Nous avons donc proposé un deuxième modèle, plus complet. Il utilise d'une part un modèle génératif où les régions semi-locales sont moins nombreuses et représentent chacune un objet distinct. Les instances d'objets peuvent être ainsi modélisées séparément. Le modèle utilise d'autre part un champ de Markov qui régularise les prédictions de labels au niveau des patchs et qui utilise les informations de contour dans l'image.

Nous avons étudié dans quelle mesure cette modélisation des différentes instances d'objet pouvait donner de meilleures segmentation. Nous avons vu qu'un modèle de couleur simple (chapitre 6) ou une information d'échelle (chapitre 7) améliorent significativement les résultats produits. Les résultats obtenus vont au-delà l'état de l'art.

Nous avons également montré que les méthodes de segmentation proposées peuvent fonctionner même si elles ne sont que faiblement supervisées.

8.2 Perspectives

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

8.2.1 Extension des méthodes proposées

Extension du modèle de segmentation. La méthode que nous avons proposée repose énormément sur la qualité de la classification individuelle des patchs extraits des images. Une bonne initialisation garantie une bonne convergence de l'algorithme. Il nous semble que ce point pourrait être approfondi et que des méthodes plus performantes pourraient être mise au point.

Il serait également opportun d'améliorer les modèles associés à chaque instance d'objet (blob). En plus de la mixture de couleur, l'enrichissement des modèles par des informations d'échelle, des modèles de co-occurence, ou des informations de géométrie entre les patchs (modèle de forme par exemple), semblent judicieux.

D'autre part, les interactions entre les instances des différents objets au sein d'une même image pourraient venir renforcer la reconnaissance par des informations de plus haut niveau. Nous pourrions ainsi tirer partie du contexte, en modélisant la structure de la scène.

Enfin, nous pouvons utiliser d'autres informations pour la régularisation entre patchs voisins, comme des contraintes dépendant de la classe des objets, qui pourraient être incorporées au champ de Markov.

Problème des instances trop proches. Nous avons vu dans le chapitre 6 que la modélisation des objets dans la méthode de segmentation était parfois mise en échec lorsque deux objets sont trop proches. Ceci limite son utilisation pour détecter des instances multiples des catégories dans les images, en particulier lorsque ces instances s'occultent. Notons qu'il s'agit d'un problème difficile, même pour les meilleures méthodes de détection de l'état de l'art.

Les méthodes basées sur des fenêtres glissantes suppriment les détections multiples par un procédé de suppression des non-maxima locaux. Ce mécanisme supprime également les objets trop proches d'un premier objet détecté.

Les modèles à géométrie faible, comme les collections de mots considérées dans ce manuscrit, possèdent les limitations que nous avons évoquées.

Enfin les modèles géométriques rigides (de type modèle de constellation) sont capables de différencier les instances d'objet grâce à leur forte géométrie. Cependant en cas d'occultation, ils seront eux-aussi mis en défaut.

Il serait possible d'améliorer le comportement de notre méthode dans ce cas de figure en incorporant une forme de géométrie associée à une instance, comme suggéré dans le paragraphe précédent. Cependant, cette géométrie doit rester suffisamment souple pour être toujours applicable aux catégories présentant de fortes variations d'apparence, comme par exemple les chats. Des modèles de co-occurences semblent un bon compromis.

8.2.2 Vers des méthodes complètement non-supervisée

Les remarques de cette section s'adressent aux méthodes proposées dans ce manuscrit mais aussi aux méthodes de l'état de l'art qui ont motivées nos travaux.

Des vocabulaires visuels universels. Nous avons évoqué dans l'introduction le problème de non-universalité du vocabulaire visuel. Les expériences que nous avons menées montrent que même si sa construction ne prend pas en compte les informations de classes, le vocabulaire dépend fortement des images d'apprentissage. Pour un descripteur donné, les statistiques d'apparence des descripteurs de patchs varient fortement selon l'échelle à laquelle sont extraits les patchs, le type d'image considéré (taille, niveau de bruit, condition d'acquisition) mais aussi la façon dont les objets apparaissent dans les images.

Il faudrait donc peut être pousser plus loin les pistes que nous avons ouvertes avec les forêts aléatoires. Ceux-ci résument l'image en un vecteur de grande dimension, avec de la redondance, et qui laisse le soin au classifieur d'apprendre les modèles de catégories. On peut envisager ainsi d'aller vers des vocabulaires universels.

Plus de classes. Les classifieurs sont dépendants des catégories considérées. A l'échelle de l'image, les classifieurs discriminants, comme les SVM, donnent de bons résultats. Mais les classifieurs doivent être ré-entrainés à chaque fois que l'on souhaite apprendre une nouvelle classe. Ce qui freine le passage à l'échelle. Lorsque nous nous intéresserons à plusieurs centaines

voire plusieurs milliers de catégories, cette organisation séquentielle des classifieurs sera mise en échec. Le même problème est rencontré lorsque l'on construit des classifieurs de patchs. A chaque nouvelle catégorie ajoutée, le classifieur doit être réappris.

La façon même dont sont organisées les catégories pose un problème pour le passage à l'échelle évoqué plus haut. Le fait qu'elles soient définies par des images d'apprentissage étiquetées (présence/absence) pour chaque catégorie d'objet considérée est contraignant et rigide. Tout d'abord, l'utilisation d'un degré d'appartenance peut être utile pour définir certaines catégories. De plus, si un grand nombre de catégories est considéré, nous avons besoin d'un nombre bien plus grand encore d'images d'apprentissage annotées. Le coût associé devient beaucoup trop élevé.

Nous pouvons imaginer d'autres façons d'apporter de la connaissance. Des premières structures, basées sur la similarité visuelle entre les images pourraient être apprises de façon non supervisée. Un utilisateur serait ensuite interrogé pour incorporer de la connaissance, par exemple, pour déterminer si deux groupes d'images créés de manière non-supervisé sont liés et de quelle façon. Des critères de similarité pourraient être appris (comme dans [66]) afin de transposer la connaissance apprises sur une classe à d'autres classes. Enfin, une organisation plus judicieuses (relation d'inclusion, d'exclusion, etc.) entre les classes pourrait être apprise et utilisée pour l'interprétation de scènes complexes.

Annexes

A

A.1 Attention visuelle du système de vision humain

FIG. A.1 – *Tableau d'Ilya Repine, intitulé en français « On ne l'attendait plus », 1988, huile sur canevas. Galerie Tretyakov, Moscou, Russie.*

Cette section illustre les travaux d'Alfred Yarbus [101] sur l'attention visuelle. Une de ces expériences les plus connues est celle sur le mouvement des yeux lors de la perception d'objets complexes, notamment lors de l'analyse du tableau de Repine (présenté figure A.1). Avant chaque analyse du tableau, les sujets sont priés d'effectuer l'une des sept tâches suivantes :

1. examiner librement le tableau
2. estimer la situation matérielle de la famille
3. évaluer l'âge des personnes
4. résumer ce que faisaient les membres de la famille avant l'arrivée du visiteur
5. se souvenir des vêtements portés par les membres de la famille
6. se souvenir de la position des personnes et des objets dans la pièce
7. estimer depuis combien de temps le visiteur est resté loin de la famille

Les chemins de fixations obtenus pour chacune des tâches sont donnés sur la figure A.2. On observe que les zones de l'image sur lesquelles s'est concentré l'examen de l'image diffèrent énormément entre les tâches.

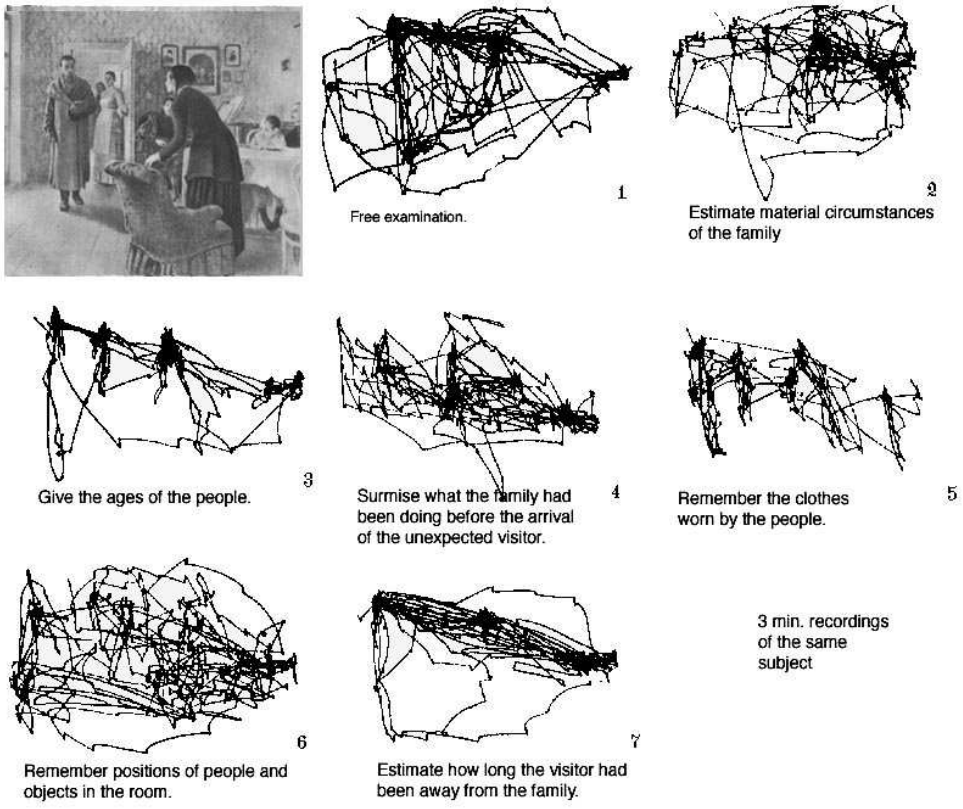


FIG. A.2 – *Résultats de l'analyse de la scène. Pour chaque tâche à effectuer, la figure montre le chemin entre les fixations de l'œil réalisées par un sujet.*

Bibliographie

[1] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 26(11) :1475–1490, November 2004. (Cité pages 7 et 25.)

[2] T. Avraham and M. Lindenbaum. Dynamic visual search using inner-scene similarity : Algorithms and inherent limitations. In *European Conference on Computer Vision*, 2004. (Cité pages 61 et 64.)

[3] C. Bishop. *Information Theory, Inference, and Learning Algorithms.* Springer, 2006. (Cité pages 12 et 13.)

[4] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Advances in Neural Information Processing Systems*, 2002. (Cité pages 13, 45, 80, 83 et 85.)

[5] J. Bonaiuto and L. Itti. Combining attention and recognition for rapid scene analysis. In *International workshop on attention and performance in computational vision*, 2005. (Cité page 64.)

[6] E. Borenstein and J. Malik. Shape guided object segmentation. In *IEEE Conference on Computer Vision & Pattern Recognition*, pages 969–976, 2006. (Cité page 83.)

[7] E. Borenstein, E. Sharon, and S. Ullman. Combining top-down and bottom-up segmentation. In *Proceedings of CVPR Workshops*, 2004. (Cité page 83.)

[8] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees.* Wadsworth and Brooks, Monterey, CA, 1984. (Cité page 66.)

[9] L. Cao and L. Fei-Fei. Spatially coherent latent topic model for concurrent object segmentation and classification. In *IEEE International Conference on Computer Vision*, 2007. (Cité page 80.)

[10] H. Cevikalp, D. Larlus, M. Douze, and F. Jurie. Local subspace classifiers : Linear and nonlinear approaches. In *IEEE Workshop on Machine Learning for Signal Processing*, Thessaloniki, Greece, aug 2007. (Cité page 19.)

[11] H. Cevikalp, D. Larlus, and F. Jurie. A supervised clustering algorithm for the initialization of rbf neural network classifiers. In *the 15th IEEE Signal Processing and Communication Applications Conference*, Eskisehir, Turkey, jun 2007. (Cité page 19.)

[12] N. Chawla, N. Japkowicz, and A. Kolcz, editors. *Proceedings of the ICML Workshop on Learning from Imbalanced Data Sets*, 2003. (Cité page 26.)

[13] G. Csurka, C. Dance, L. Fan, J. Williamowski, and C. Bray. Visual categorizationwith bags of keypoints. In *ECCV workshop on Statistical Learning in ComputerVision*, pages 59–74, 2004. (Cité pages 6, 25, 63 et 83.)

[14] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *European Conference on Computer Vision*, 2006. (Cité pages 112, 120 et 121.)

[15] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http ://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html, 2007. (Cité pages 17, 111 et 112.)

[16] M. Everingham, A. Zisserman, C. K. I. Williams, L. J. Van Gool, M. Allan, C. M. Bishop, O. Chapelle, N. Dalal, T. Deselaers, G. Dorkó, S. Duffner, J. Eichhorn, J. D. R. Farquhar, M. Fritz, C. Garcia, T. Griffiths, F. Jurie, D. Keysers, M. Koskela, J. Laaksonen, D. Larlus, B. Leibe, H. Meng, H. Ney, B. Schiele, C. Schmid, E. Seemann, J. Shawe-Taylor, A. J. Storkey, S. Szedmák, B. Triggs, I. Ulusoy, V. Viitaniemi, and J.Zhang. The 2005 pascal visual object classes challenge. In *MLCW*, pages 117–176, 2005. (Cité pages 16, 39, 40 et 54.)

[17] M. Everingham, A. Zisserman, C. K. I. Williams, and L. Van Gool. The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results. http ://www.pascal-network.org/challenges/VOC/voc2006/results.pdf, 2006. (Cité page 16.)

[18] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2005. (Cité page 47.)

[19] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *IEEE International Conference on Computer Vision*, volume 101, pages 5228–5235, 2005. (Cité pages 47, 63 et 80.)

[20] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *IEEE Conference on Computer Vision & Pattern Recognition*, pages 264–271, 2003. (Cité page 8.)

[21] G. Fritz, C. Seifert, L. Paletta, and H. Bischof. Entropy based saliency maps for object recognition. In *Proceeding of the Early Cognitive Vision Workshop*, 2004. (Cité page 63.)

[22] D. Gao and N. Vasconcelos. Discriminant saliency for visual recognition from cluttered scenes. In *Advances in Neural Information Processing Systems*, 2004. (Cité page 63.)

[23] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 6(6) :721–741, 1984. (Cité page 80.)

[24] B. Georgescu, I. Shimshoni, and P. Meer. Mean shift based clustering in high dimensions : A texture classification example. In *IEEE International Conference on Computer Vision*, pages 456–463, 2003. (Cité page 26.)

[25] P. Geurts. *Contributions to decision tree induction : bias/variance tradeoff and time series classification.* PhD thesis, University of Liège, 2002. (Cité page 66.)

[26] T. Griffiths and M. Steyvers. Finding scientific topics. In *Proceedings of the National Academy of Sciences*, 2004. (Cité pages 48 et 86.)

[27] D. Hall, B. Leibe, and B. Schiele. Saliency of interest points under scale changes. In *British Machine Vision Conference*, 2002. (Cité page 63.)

[28] R.M. Haralick and L.G. Shapiro. Image segmentation techniques. *CV-GIP*, 29 :100–132, 1985. (Cité page 79.)

[29] C. Harris. Acombined corner and edge detector. In *Alvey vision Conference*, Manchester UK, august 1988. (Cité page 63.)

[30] T. Hastie, R. Tibshirani, and J. Fridman. *The Elements of Statistical Learning : Data Mining, Inference, and Prediction.* Springer-Verlag, 2001. (Cité page 6.)

[31] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Mach. Learn.*, 42(1-2) :177–196, 2001. (Cité pages 13, 45 et 83.)

[32] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 20(11) :1254–1259, November 1998. (Cité pages 61 et 63.)

[33] T. Joachims. Text categorization with support vector machines : Learning with many relevant features. In *European Conference on Machine Learning*, pages 137–142, 1998. (Cité page 6.)

[34] B. Julesz. Texton gradients : The texton theory revisited. *Biological Cybernetics*, 54(4) :245–251, 1986. (Cité page 25.)

[35] F. Jurie and C. Schmid. Scale-invariant shape features for recognition of object categories. In *IEEE Conference on Computer Vision & Pattern Recognition*, pages II : 90–96, 2004. (Cité pages 26, 27 et 72.)

[36] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *IEEE International Conference on Computer Vision*, pages 604–610, 2005. (Cité pages 7, 26 et 46.)

[37] T. Kadir and M. Brady. Saliency, scale and image description. *IJCV*, 45(2), November 2001. (Cité page 63.)

[38] C. Koch and S. Ullman. Shifts in selective visual attention : Towards the underlying neural circuitry. *Human Neurobiology*, 4 :219–227, January 1985. (Cité page 63.)

[39] M. Kumar, P. Torr, and A. Zisserman. OBJ CUT. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2005. (Cité page 82.)

[40] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields : probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*, volume 18, pages 282–289, 2001. (Cité page 80.)

[41] D. Larlus and F. Jurie. Segmentation de catégories d'objets par combinaison d'un modèle d'apparence et d'un champ de markov. *Information - Interaction - Intelligence (i3).* Accepted for publication. (Cité page 19.)

[42] D. Larlus and F. Jurie. Latent mixture vocabularies for object categorization. In *British Machine Vision Conference*, 2006. (Cité page 19.)

[43] D. Larlus and F. Jurie. Category level object segmentation. In *International Conference on Computer Vision Theory and Applications*, mar 2007. (Cité page 19.)

[44] D. Larlus and F. Jurie. Combining appearance models and markov random fields for category level object segmentation. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2008. (Cité page 19.)

[45] D. Larlus and F. Jurie. Latent mixture vocabularies for object categorization and segmentation. *Journal of Image & Vision Computing*, 2008. accepted. (Cité page 19.)

[46] D. Larlus, E. Nowak, and F. Jurie. Segmentation de catégories d objets par combinaison d un modèle d apparence et d un champs de markov. In *Reconnaissance des Formes et Intelligence Artificielle*, 2008. (Cité page 19.)

[47] D. Larlus, J. Verbeek, and F. Jurie. Category level object segmentation by combining bag-of-words models and markov random fields. *International Journal of Computer Vision*. Submitted for publication. (Cité page 19.)

[48] S. Lazebnik, C. Schmid, and J. Ponce. Semi-local affine parts for object recognition. In *British Machine Vision Conference*, volume 2, pages 779–788, 2004. (Cité pages 15 et 54.)

[49] S. Lazebnik, C. Schmid, and J. Ponce. A maximum entropy framework for part-based texture and object recognition. In *IEEE International Conference on Computer Vision*, 2005. (Cité pages 14, 15 et 54.)

[50] B. Leibe, K. Mikolajczyk, and B. Schiele. Efficient clustering and matching for object class recognition. In *British Machine Vision Conference*, Sep 2006. (Cité page 26.)

[51] B. Leibe and B. Schiele. Interleaved object categorization and segmentation. In *British Machine Vision Conference*, 2003. (Cité pages 7, 10, 14, 25, 63, 82, 83 et 121.)

[52] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1) :29–44, June 2001. (Cité pages 25 et 63.)

[53] Y. Li, J. Sun, C.K. Tang, and H.Y. Shum. Lazy snapping. *ACM Trans. Graph.*, 23(3) :303–308, 2004. (Cité page 82.)

[54] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2) :91–110, 2004. (Cité pages 23, 25, 50, 63, 65, 83, 87, 98, 115, 121 et 122.)

[55] R. Marée, P. Geurts, J. Piater, and L. Wehenkel. Random subwindows for robust image classification. In *IEEE Conference on Computer Vision & Pattern Recognition*, volume 1, pages 34–40, 2005. (Cité pages 8, 62, 64, 65 et 71.)

[56] M. Marszalek and C. Schmid. Spatial weighting for bag-of-features. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2006. (Cité page 25.)

[57] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans.*

Pattern Anal. Mach. Intell., 26(5) :530–549, 2004. (Cité pages 9, 98 et 122.)

[58] R. R. Mettu and C. G. Plaxton. The online median problem. In *Annual Symposium on Foundations of Computer Science*, page 339, 2000. (Cité page 27.)

[59] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *European Conference on Computer Vision*, 2002. (Cité pages 25 et 63.)

[60] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *IEEE Conference on Computer Vision & Pattern Recognition*, pages 257–263, 2003. (Cité page 23.)

[61] F. Moosmann, D. Larlus, and F. Jurie. Learning saliency maps for object categorization. In *ECCV International Workshop on The Representation and Use of Prior Knowledge in Vision*. Springer, 2006. (Cité page 19.)

[62] F. Moosmann, E. Nowak, and F. Jurie. Randomized clustering forests for image classification. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2008. to appear. (Cité pages 41 et 108.)

[63] V. Navalpakkam and L. Itti. Sharing resources : Buy attention, get recognition. In *International workshop on attention and performance in computational vision*, 2003. (Cité page 64.)

[64] R. Neal. Markov chain sampling methods for dirichlet process mixture models. Technical Report 9815, Dept. of Statistics, University of Toronto, Sep 1998. (Cité page 99.)

[65] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *IEEE Conference on Computer Vision & Pattern Recognition*, volume 2, pages 2161–2168, 2006. (Cité page 26.)

[66] E. Nowak and F. Jurie. Learning visual similarity measures for comparing never seen objects. In *cvpr*, jun 2007. see also http ://lear.inrialpes.fr/people/nowak/. (Cité page 135.)

[67] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *European Conference on Computer Vision*. Springer, 2006. (Cité page 26.)

[68] A. Opelt and A. Pinz. *Object Localization with Boosting and Weak Supervision for Generic Object Recognition*, volume 3540. Springer, 2005. (Cité pages 8, 41, 70 et 72.)

[69] P. Orbanz and J. M. Buhmann. Smooth image segmentation by nonparametric bayesian inference. In *European Conference on Computer Vision*, 2006. (Cité page 81.)

[70] F. Perronnin, C. Dance, G. Csurka, and M. Bressan. Adapted vocabularies for generic visual categorization. In *European Conference on Computer Vision*, 2006. (Cité pages 8 et 45.)

[71] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007. (Cité page 26.)

[72] P. Quelhas, F. Monay, J. Odobez, D. Gatica-Perez, T. Tuytelaars, and L. Van Gool. Modeling scenes with local descriptors and latent aspects. In *IEEE International Conference on Computer Vision*, 2005. (Cité pages 45, 47 et 49.)

[73] C. Rother, V. Kolmogorov, and A. Blake. "grabcut" : interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3) :309–314, 2004. (Cité pages 82 et 100.)

[74] Francois Saidi, Olivier Stasse, and Kazuhito Yokoi. Active visual search by a humanoidrobot. In M. Sang Kim S. Lee, I. Hong Suh, editor, *Recent Progress in Robotics ; ViableRobotic Service to Human, selected papers from ICAR 2007*, page accepted. LNCIS Series, Springer-Verlag, 2007. (Cité page 120.)

[75] K. Sayre. Machine recognition of handwritten words : A project report. *Pattern Recognition*, 5(3) :213–228, 1973. (Cité page 9.)

[76] Cordelia Schmid and Roger Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 19(5) :530–534, 1997. (Cité page 5.)

[77] N. Sebe and M.S. Lew. Comparing salient point detectors. *Pattern Recognition Letters*, 24(1-3) :89–96, January 2003. (Cité page 63.)

[78] T. Serre, M. Riesenhuber, J. Louie, and T. Poggio. On the role of object-specific features for real world object recognition in biological vision. In *British Machine Vision Conference*, 2002. (Cité page 63.)

[79] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *IEEE International Conference on Computer Vision*, pages I :503–510, 2005. (Cité page 80.)

[80] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost : Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European Conference on Computer Vision*, pages I : 1–15, 2006. (Cité pages 10, 80, 81 et 111.)

[81] J. Sivic, B. Russell, A. Efros, A. Zisserman, and B. Freeman. Discovering objects and their location in images. In *IEEE International Conference on Computer Vision*, 2005. (Cité pages 45, 47 et 51.)

[82] J. Sivic and A. Zisserman. Video google : A text retrieval approach to object matching in videos. In *IEEE International Conference on Computer Vision*, 2003. (Cité page 25.)

[83] O. Stasse, A. Davison, R. Sellaouti, and K. Yokoi. Real-time 3d slam for humanoid robot considering pattern generator information. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, Beijing, China*, pages 348–355, October 9-15 2006. (Cité page 119.)

[84] O. Stasse, D. Larlus, B. Lagarde, A. Escande, F. Saidi, A. Kheddar, K. Yokoi, and F. Jurie. Towards autonomous object reconstruction for visual search by the humanoid robot hrp-2. In *IEEE RAS/RSJ Conference on Humanoids Robot*, 2007. (Cité pages 19, 120 et 121.)

[85] E. Stollnitz, T. DeRose, and D. Salesin. Wavelets for computer graphics : A primer, part 1. *IEEE Computer Graphics and Applications*, 15(3) :76–84, 1995. (Cité page 65.)

[86] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky. Describing visual scenes using transformed dirichlet processes. In *Advances in Neural Information Processing Systems*, 2005. (Cité page 80.)

[87] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky. Describing visual scenes using transformed objects and parts. *International Journal of Computer Vision*, 77(1-3) :291–330, 2008. (Cité page 99.)

[88] J. van de Weijer and C. Schmid. Coloring local feature extraction. In *European Conference on Computer Vision*, pages 334–348, 2006. (Cité pages 98, 115 et 122.)

[89] V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995. (Cité pages 24 et 31.)

[90] J. Verbeek and B. Triggs. Region classification with markov field aspect models. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2007. (Cité pages 81 et 111.)

[91] J. Verbeek and B. Triggs. Scene segmentation with crfs learned from partially labeled images. In *Advances in Neural Information Processing Systems*, 2008. (Cité page 80.)

[92] M. Vidal-Naquet and S. Ullman. Object recognition with informative features and linear classification. In *IEEE International Conference on Computer Vision*, pages 281–288, 2003. (Cité pages 26 et 63.)

[93] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, 2001. (Cité pages 120 et 121.)

[94] K.N. Walker, T.F. Cootes, and C.J. Taylor. Locating salient object features. In *British Machine Vision Conference*, 1998. (Cité page 63.)

[95] D. Walther, U. Rutishauser, C. Koch, and P. Perona. On the usefulness of attention for object recognition. *European Conference on Computer Vision*, 2004. (Cité page 63.)

[96] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *European Conference on Computer Vision*, pages I : 18–32, 2000. (Cité page 25.)

[97] G. M. Weiss. Mining with rarity : a unifying framework. *SIGKDD Explor. Newsl.*, 6(1) :7–19, 2004. (Cité page 27.)

[98] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *IEEE International Conference on Computer Vision*, 2005. (Cité pages 8, 26, 45, 46 et 47.)

[99] J. Winn and N. Jojic. Locus : Learning object classes with unsupervised segmentation. In *IEEE International Conference on Computer Vision*, pages 756–763, 2005. (Cité page 83.)

[100] J. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *IEEE Conference on Computer Vision & Pattern Recognition*, pages 37–44, 2006. (Cité page 82.)

[101] A. Yarbus. *Eye Movements and Vision*. New York : Plenum, 1967. (Cité pages 61, 63 et 137.)

[102] Y. Ye and J. Tsotsos. Where to look next in 3D object search. In *Proc. IEEE Int. Symp. Computer Vision*, 1995. (Cité page 69.)

[103] A. Zaharescu, A. Rothenstein, and J. Tsotsos. Towards a biologically plausible active visual search model. In *International workshop on attention and performance in computational vision*, pages 133–147, 2004. (Cité page 61.)

Ce document a été préparé à l'aide de l'éditeur de texte GNU Emacs et du logiciel de composition typographique LaTeX.