Construction et utilisation de la sémantique dans le cadre de l'annotation automatique d'images

Christophe Millet

Annotation automatique d'images : annotation cohérente et création automatique d'une base d'apprentissage

Christophe Millet

Thèse soutenue le 14 Janvier 2008

Pour obtenir le grade de docteur de
l'École Nationale Supérieure des Télécommunications
Spécialité Signal et Images.

Directrice de thèse :	Isabelle Bloch
Encadrants CEA :	Patrick Hède
	Pierre-Alain Moëllic
Président du jury :	Jean Charlet
Rapporteurs :	Patrick Gros
	Florence Sèdes
Examinateurs :	Allan Hanbury
	Beatriz Marcotegui

Remerciement

Je tiens à remercier les personnes suivantes :

Tout d'abord, toute l'équipe du LIC2M ! Je remercie plus spécialement ceux qui ont partagé mon bureau et supporté mes blagounettes : Patrick, Pierre-Alain et Adrian sans oublier Pixel et Syntagme ; Bertrand, l'expert en réseaux bayésiens ; Hervé grace à qui j'ai trouvé un poste après ma thèse ; Antoine et Aymen qui ont travaillé pour moi sans le savoir ; mes chefs de labo bien aimés : Christian Fluhr, Gregory Grefenstette puis Olivier Mesnard.

Isabelle, ma directrice de thèse, et tout ses doctorants. Isabelle a un emploi du temps très chargé mais a toujours trouvé un créneau pour s'occuper de moi quand j'en avais besoin et sait remonter le moral quand on a une petite baisse, et donner de nouvelles idées quand on en a plus (mais en général, ce n'est plus un problème à la fin de la thèse : on en a trop). Je reste aussi toujours très impressionné par la vitesse et la qualité de la relecture de mes articles et des chapitres de la thèse.

Les membres du jury pour avoir accepté de faire parti de mon jury, et pour leurs remarques, dans les rapports et lors de la soutenance.

Dont plus particulièrement Béatrice Marcotegui pour avoir partagé avec moi son algorithme de segmentation en cascades utilisant la morphologie mathématique à la fois rapide et efficace (quoique je n'ai pas eu la toute dernière version qui est encore meilleure semble-t-il !),

et aussi Allan Hanbury pour m'avoir accueilli pendant deux semaines au PRIP à Vienne dans le cadre du réseau d'excellence MUSCLE. C'était hélas trop court, mais très enrichissant (et Vienne est une très belle ville).

Ma famille, qui m'a toujours poussé à poursuivre mes études jusqu'au bout, et mes amis de différentes catégories : les doctorants qui comprennent la situation, les non doctorants qui aiment poser la question : "alors la thèse, ça avance ?", et ceux qui sont assez compréhensifs pour ne pas la poser et parler d'autre chose.

Enfin, je tiens à souligner qu'aucun animal n'a été maltraité au cours de nos expériences.

Résumé

L'annotation automatique d'images est un domaine du traitement d'images permettant d'associer automatiquement des mots-clés ou du texte à des images à partir de leur contenu afin de pouvoir ensuite rechercher des images par requête textuelle. L'annotation automatique d'images cherche à combler les lacunes des deux autres approches actuelles permettant la recherche d'images à partir de requête textuelle. La première consiste à annoter manuellement les images, ce qui n'est plus envisageable avec le nombre croissant d'images numériques, d'autant que différentes personnes annotent les images différemment. La seconde approche, adoptée par les moteurs de recherche d'images sur le web, est d'utiliser les mots de la page web contenant l'image comme annotation de cette image, avec l'inconvénient de ne pas prendre du tout en compte le contenu de l'image.

Quelques systèmes d'annotation automatique d'images commencent à émerger, avec certaines limites : le nombre d'objets reconnus reste de l'ordre de 10 à 100, alors que les humains sont capables de reconnaître de l'ordre de 10000 objets ; les mots-clés générés comme annotation pour une image sont parfois en contradiction entre eux, par exemple « éléphant » et « ours polaire » peuvent être détectés dans une même image ; la base de données pour l'apprentissage des objets est construite manuellement.

Les travaux effectués au cours de cette thèse visent à proposer des solutions à ces problèmes, d'une part en introduisant de la connaissance dans l'annotation automatique d'images, d'autre part en proposant un système complètement automatique, où notamment la base d'images pour l'apprentissage est construite automatiquement à partir des images du Web. Cette thèse est constituée de trois parties :

La première partie concerne la catégorisation d'une image en fonction de son type (photo, carte, peinture, clipart) puis pour les photographies, on s'intéresse à savoir quel est le contexte de la scène photographiée : est-ce une photographie d'intérieur ou d'extérieur, une photographie prise de nuit ou de jour, une photographie de nature ou de ville ? Y a-t-il des visages dans la photo ? Y a-t-il du ciel, de l'herbe, de l'eau, de la neige, une route, ... dans l'image ?

La deuxième partie étudie la possibilité de construire automatiquement une base d'images d'apprentissage pour n'importe quel objet donné. Ne connaissant que le nom du concept que l'on souhaite apprendre, nous déduisons automatiquement sa couleur et le milieu dans lequel il se trouve à partir du web. Nous utilisons ces connaissances pour filtrer des images récupérées également sur Internet, c'est-à-dire rejeter les images ne correspondant pas au concept recherché, et isoler la région correspondant à l'objet dans

l'image. Un séparateur à vaste marge peut ensuite apprendre à reconnaître ces objets dans de nouvelles images.

Enfin, la troisième partie concerne la désambiguïsation, c'est-à-dire la manière de choisir parmi plusieurs hypothèses de reconnaissance pour une région donnée celle qui permet une annotation globale de l'image sémantiquement cohérente. Deux sortes de désambiguïsation sont développées : la première utilise les relations spatiales, et s'assure que par exemple le ciel est toujours au-dessus de la mer. La deuxième tient compte du contexte de l'image, en utilisant la relation entre les objets et les milieux dans lesquels ils se trouvent : on a plus de chance de trouver un éléphant dans la savane, et un dauphin dans l'eau.

Abstract

Automatic image annotation is an image processing which allows to automatically associate keywords or text to images based on their content, so that the image can be retrieved via text-based searches later on. Automatic image annotation tries to overcome the problems of the two other approaches allowing to find images based on textual queries. The first one consists in manually annotation the images, which is not possible considering the steady increase in the number of digital images, and furthermore, each people annotate images differently. The second approach, used by the image search engines available on the Internet, is to use the text in the webpages containing images as an annotation of these images, with the drawback that it does not take into account the content of the image.

Recently, some automatic image annotation systems have been released, with some limits : the number of identified objects is typically between 10 and 100 whereas humans are able to recognize tens of thousands of objects ; generated keywords are sometimes contradicting each others, for example, "elephant" and "polar bear" can be detected in the same image ; the image database for learning the objects is manually constructed.

This thesis tries to propose some solutions to these problems, by introducing knowledge in automatic image annotation, and by proposing a totally automatic system, where in particular the learning image database is automatically constructed with images from the Internet. This thesis is divided into three parts :

The first part categorizes an image based on its type (photograph, map, painting, clipart) and, for photographs, determines the context of the depicted scene : is it an indoor or outdoor scene ? taken by night or by day ? taken in the nature or in a city environment ? Is there any face in the photograph ? Is there sky, grass, water, snow, road, etc. in the image ?

The second part studies the possibility to automatically build a learning image database for any given object. Starting only from the name of the object that we wish to learn, we automatically deduce from the Web its colour and its typical environment. We then use this knowledge to filter the images retrieved from the Internet by removing the images that do not correspond to the concept of interest, and by isolating the region in the image that corresponds to the object. A support vector machine can then learn to recognize these objects in unknown images.

Eventually, the third part is about disambiguation, i.e. how to choose, among several hypotheses of recognized objects in each region, the one which gives the most semanti-

cally consistent annotation of the image. Two kinds of disambiguation are developed : the first one ensures that spatial relationships are respected, for example that a sky region is always above a water region. The second one takes into account the context of the image by using the relation between the objects and their typical context : the probability is higher to find an elephant in the savannah and a dolphin in the water.

Table des matières

Chapitre 1

Introduction

一幅画抵得过一万字
yīfúhuà dǐdéguò yīwànzì
Une image vaut dix-mille mots.
(Confucius)

Avec le développement des appareils photographiques numériques, le nombre d'images numériques disponibles a subi une augmentation phénoménale ces dernières années. Voici quelques chiffres pour donner un ordre d'idée :Google recensait environ 2 milliard d'images présentes sur Internet le 9 août 2005[1]. En octobre 2006, il a été estimé que sur Flickr, le site web de partage de photographies le plus connu, environ 920 000 images sont soumises chaque jour[2]. Il n'est pas imaginable de consulter toutes ces images à la main pour retrouver celles que l'on recherche. Des solutions ont donc été développées pour permettre de faire des recherches dans des bases de données d'images de manière automatique. Ces solutions se répartissent en deux catégories :

- La recherche d'images par similarité (CBIR : *content-based image retrieval*) : l'utilisateur fournit en entrée du système une image, dite image question, ou requête, et souhaite trouver d'autres images qui lui sont similaires. La notion de similarité reste à définir mais, en général, c'est une similarité visuelle en couleur, texture et/ou forme qui est utilisée. Éventuellement, la fonction de similarité peut être affinée en fonction des besoins de l'utilisateur avec des techniques dites de « retour de pertinence », connues en anglais sous le nom de *relevance feedback*.
- La recherche d'images par mots clés : à partir d'un ou plusieurs mots, l'objectif est de retrouver des images qui sont en rapport avec ce mot. Les moteurs classiques de recherche d'images tels que Google et Yahoo ! permettent par exemple d'effectuer

[1] source : http ://www.webrankinfo.com/actualites/200508-index-google.htm
[2] source : http ://blog.forret.com/2006/10/a-picture-a-day-flickrs-storage-growth/

une recherche d'images par mots clés et les images retournées sont celles dont le nom ou la page Internet dont elles sont issues contiennent ce mot.

Il est bien évidemment possible de combiner ces deux formes de recherche, par exemple en commençant par une recherche par mot clé pour sélectionner un sous-ensemble d'une base, puis en effectuant une recherche par similarité sur ce sous-ensemble afin de réorganiser les images en fonction de l'intérêt de l'utilisateur. Nous décrivons par la suite plus en détail les avantages et les inconvénients de ces deux méthodes.

1.1 Recherche d'images par similarité

La recherche d'images par similarité se déroule en deux phases : la phase d'indexation puis la phase de recherche. La phase d'indexation consiste à calculer des descripteurs de bas niveau sur toutes les images d'une base déterminée. Ces descripteurs servent à représenter l'image sous une forme condensée, par exemple un histogramme à 64 composantes peut être choisi comme représentation d'une image initialement composée de quelques millions de pixels. On trouve parfois les termes de signature ou d'ADN de l'image pour illustrer ce concept. Cette signature présente deux avantages. D'une part, elle est petite : il est plus rapide de comparer 64 valeurs que de comparer les pixels de deux images un par un. D'autre part, elle s'attache à des caractéristiques particulières de l'image, couleur, texture ou forme, qui ne ressortiraient pas forcément dans un tableau de pixels.

La phase de recherche consiste à présenter au système une image, appartenant ou non à la base, pour laquelle les mêmes descripteurs que pour la phase d'indexation sont calculés afin de pouvoir la comparer aux images de la base indexée. Une fonction de distance ou mesure de similarité est alors à définir pour pouvoir retourner les images les plus proches au sens de cette distance. La thèse de Julien Fauqueur [40] de 2003 présente un état de l'art des mesures de similarités les plus utilisées.

Le problème majeur de la recherche d'images par similarité est le manque de sémantique dans les requêtes. Ainsi, une photographie d'un avion dans le ciel et d'un oiseau dans le ciel seront souvent visuellement très proches, notamment si le ciel occupe une bonne partie de l'image. En posant la photographie de l'avion comme requête, il est alors très probable d'obtenir la photographie de l'oiseau dans les résultats, alors qu'on s'attendrait plutôt a priori à obtenir d'abord d'autres photographies d'avions, éventuellement dans des milieux différents.

1.2 Recherche d'images par mots clés

Malgré tous les travaux de recherche menés pour trouver les meilleurs descripteurs de bas niveau pour les images (couleur, texture, forme), la performance des systèmes de recherche d'images par similarité est loin d'être satisfaisante à cause de ce que l'on appelle le fossé sémantique[3] [38]. Ce fossé sémantique représente la différence qui existe

[3]en anglais : *semantic gap*

entre les descripteurs de bas niveau d'une image – tels que les pixels, la couleur ou la texture – et la sémantique contenue dans l'image – « il y a un homme, jeune, il est joyeux,... ». En effet, des images représentant des concepts sémantiques (de haut niveau) différents peuvent avoir des descripteurs de bas niveau présentant de nombreux points communs alors que des images représentant un même objet peuvent être dispersés dans l'espace des descripteurs de bas niveau. Eakins [37] distingue trois niveaux de requêtes dans les systèmes de recherche d'images par le contenu :

- Niveau 1 : la recherche à partir d'attributs primitifs tels que la couleur, la texture, la forme ou les positions spatiales des éléments de l'image. Les requêtes de ce niveau sont typiquement les requêtes par l'exemple : étant donné une image, on cherche d'autres images qui lui ressemblent : « trouve des images comme celle-ci ».
- Niveau 2 : la recherche d'objets d'un type donné identifié par les descripteurs extraits de ces objets. Ce niveau peut faire appel à des inférences logiques. Par exemple : « trouve des images de fleur ».
- Niveau 3 : la recherche par attributs abstraits, impliquant un certain nombre de raisonnements de haut niveau afin de comprendre l'agencement des objets et de la scène, le sentiment qu'ils évoquent, etc. Par exemple : « trouve des images de fête d'anniversaire ».

Les niveaux 2 et 3 sont ce que l'on considère comme des recherches sémantiques d'images. L'écart entre les requêtes de niveaux 1 et 2 constitue le fossé sémantique. Le niveau 1 n'est pas toujours facilement utilisable dans la pratique, car il nécessite d'avoir à disposition une image exemple de ce que nous cherchons, ou de procurer au système un schéma dessiné à la main. Cela n'est pas toujours possible, et les niveaux 2 et 3 sont plus utilisables en pratique car ils permettent la recherche d'images par mots clés.

En comparant cette classification avec une étude menée en 1997 sur les requêtes des utilisateurs de plusieurs banques d'images [4], il ressort qu'en pratique, on trouve en effet très peu de requêtes de niveau 1. La plupart des requêtes sont de niveau 2, mais le niveau 3 en constitue également une part significative, notamment pour les banques d'images d'art.

Afin de pouvoir accéder à ces deux niveaux sémantiques (2 et 3) par des requêtes textuelles, il faut d'abord avoir annoté les images, c'est-à-dire leur avoir associé des mots clés (qui peuvent être considérés comme étant de haut niveau sémantique par rapport aux descripteurs souvent qualifiés de bas niveau) décrivant le contenu de l'image. Il devient alors possible de comparer et de retrouver les images selon leur sémantique par l'intermédiaire de ces mots clés.

1.3 Contexte applicatif de cette thèse

Le but visé par nos travaux est de pouvoir annoter automatiquement et de façon cohérente des images issues de collections personnelles et que l'on retrouve typiquement sur Internet lorsque l'on fait une recherche dans Google Image Search, ou lorsque l'on consulte le site FlickR. Cette annotation devrait ensuite permettre de retrouver les images par requêtes textuelles.

Ces images présentent des inconvénients lors de la recherche par mot clés, par rapport à des collections professionnelles qui ont les moyens de faire appel à des documentalistes :

- une annotation manuelle n'est pas toujours disponible pour ces images, par exemple sur le site de FlickR, seules 10% des images sont annotées,
- l'annotation, lorsqu'elle est disponible, est faite sans règles, et deux images similaires peuvent avoir un ensemble de mots clés complètement différents.

1.4 Différences entre annotation manuelle et automatique

1.4.1 Annotation manuelle

Bien souvent, les annotations d'images sont effectuées manuellement, ce qui permet d'obtenir le niveau d'abstraction désiré. Ces méthodes remontent aux années 1970. Elles font appel à des documentalistes experts en annotation manuelle d'images, appelés iconographes. Cette annotation est utilisée actuellement par les musées, les agences de presse, ou les fournisseurs de contenus visuels. GraphicObsession[4], par exemple, propose 500 000 images, chacune étant annotée manuellement avec typiquement de 10 à 50 mots clés par image.

Cependant, écrire les annotations à la main est un processus très long et coûteux en comparaison avec une méthode qui serait automatique. De plus, les annotations manuelles sont très subjectives : des personnes différentes adoptent des points de vue différents pour une même image, utilisent un niveau de détail plus ou moins grand (animal, félin, puma) ou emploient des mots clés différents pour décrire le même objet (puma, couguar, tigre rouge). Il est d'autant plus difficile de maîtriser la cohérence des annotations procurées par différentes personnes que la base de données est grande. Ainsi, chez GraphicObsession, deux images de gros plans d'une tête de jaguar très similaires contiennent les mots clés suivants :

1. animaux de safari ; couleur ; danger ; espèces en danger ; faune ; faune sauvage ; félin ; grand félin ; gros plan ; jaguar ; jour ; marron ; moustaches ; plein cadre ; prise de vue en extérieur ; sans personnage ; tête d'un animal ; thème des animaux ; un seul animal ; vertical ; vie sauvage ; vue de face.
2. animal ; carnivore ; face ; gaze ; jaguar ; nature ; vertical

On constate d'une part que l'une est annotée en français, l'autre en anglais, et d'autre part que même s'il s'agissait de la même langue, seuls les mots « jaguar » et « vertical » (orientation de la photo) sont communs aux deux annotations.

J. Sunderland [129] a mené une étude sur les variations et les différents niveaux de détails que l'on peut obtenir en annotation manuelle d'images en fonction de la personne qui annote. Il a demandé à un enfant de 12 ans, à un profane et à un historien d'art de décrire la même image : une peinture de John Everett Millais *Le Christ dans la maison de ses parents* (figure 1.1).

L'expérience est relatée ainsi par K. Vezina [148] :

[4]http ://www.graphicobsession.fr

FIG. 1.1 – Peinture de John Everett Millais *Le Christ dans la maison de ses parents* 1850. Sunderland [129] a utilisé cette peinture pour montrer que trois personnes d'horizons différents produisent des annotations d'images très variées.

- l'historien d'art a identifié l'artiste, le titre de l'œuvre, sa date d'exécution, l'endroit où se trouvait le tableau, le style de l'œuvre (préraphaélite) ainsi que le sujet (moment prophétique pour Jésus lorsqu'il se blesse avec un clou) et il continue son interprétation en expliquant la symbolique de l'œuvre ;
- le profane, quant à lui, indique que c'est une image religieuse où est représentée la Famille Sainte dans l'atelier de Joseph. Il mentionne aussi le fait qu'il est évident que la famille est heureuse et que l'on voit la relation d'amour et de bonheur qui existe entre le père, la mère et l'enfant ;
- l'enfant de 12 ans a plutôt décrit les éléments se trouvant dans l'image : une femme à genou tenant un enfant. L'enfant a un trou dans sa main. Un homme tient la main de l'enfant et un clou ou quelque chose. Il y a des copeaux de bois sur le plancher, un jeune garçon avec un bol dans ses mains, etc.

Les annotations ainsi générées auront donc des niveaux de détails très différents selon la qualification de la personne qui effectue l'annotation. Cela aura une incidence ensuite pour la recherche de telles images si ces annotations sont utilisées pour retrouver l'image. De plus, de même que pour l'annotation, les requêtes varieront en fonction de la personne qui interroge, et on peut imaginer également les trois mêmes niveaux de requêtes : « une image avec un enfant », « une image représentant la Famille Sainte » ou bien « une peinture de l'artiste Millais ».

Furnas et al. [53] ont étudié la difficulté pour les nouveaux utilisateurs d'un système d'utiliser les bons mots clés afin d'obtenir les résultats qu'ils souhaitent ; c'est ce qu'ils

ont appelé le problème de vocabulaire. Leur étude montre que si une personne assigne un unique nom à un objet, un novice au système n'arrivera à y accéder du premier coup que dans 10% à 20% des cas, car il utilisera un synonyme, ou un terme plus précis (hyponyme) ou moins précis (hypernyme).

Afin de résoudre ce problème, deux solutions sont généralement envisagées. La première est l'utilisation d'une base de données lexicale permettant de connaître les relations entre les concepts : les synonymes (automobile est un synonyme de voiture), les hypernymes (véhicule est un hypernyme de voiture), les hyponymes (quatre-quatre est un hyponyme de voiture), les méronymes (roue est un méronyme de voiture), les holonymes (voiture est un holonyme de roue), ... Ces information sont ensuite utilisées pour reformuler la requête. Ainsi, si par exemple la requête est « voiture » et que la base de données contient une image annotée « quatre-quatre », il sera possible d'afficher cette image dans la liste des réponses. Une base de données lexicale telle que WordNet [97] contient ce type d'information et peut être utilisée dans ce but, comme l'ont proposé notamment Hollink et al. [63].

Une autre solution à ce problème est de faire annoter chaque image par le plus grand nombre de personnes possible afin d'atténuer les problèmes dus à la subjectivité d'une personne en particulier, et de permettre d'avoir en quelque sorte une « opinion publique » sur l'image. Pour que cela soit réalisable, il faut notamment un très grand nombre de personnes disponibles. Cela devient désormais possible grâce à Internet et au développement des sites collaboratifs.Le principe est le suivant : chaque individu annote manuellement avec a priori peu de mots clés, ce qui ne lui prend que très peu de temps, mais le nombre de participants devrait permettre d'obtenir finalement beaucoup de mots clés par image parmi lesquels certains seront souvent cités, et d'autres, plus subjectifs, ne seront cités que par une seule personne. Différents points de vues seront ainsi exprimés dans les annotations ce qui rend l'annotation plus objective et l'accès à l'image devient possible par un plus grand nombre de personnes.

Citons comme exemple le site Flickr[5] qui permet à des internautes de déposer leurs photographies et de leur attribuer des tags. N'importe quel autre utilisateur peut également rajouter des commentaires à ces photographies, qui pourront servir alors comme annotation manuelle. Un autre exemple est le site *ESP Game* [147] [6] dont l'ambition est d'annoter les images du Web. Le principe est le suivant : deux personnes, qui ne se connaissent a priori pas, observent en même temps une même image et proposent différents mots clés pour cette image. Si l'un des mots proposés est commun aux deux personnes, alors ce mot est retenu et ajouté aux annotations de l'image. Les images sont proposées plusieurs fois à plusieurs personnes et, afin de s'assurer d'obtenir de nouveaux mots clés, ceux qui ont été précédemment affectés sont interdits. Le fait que deux personnes doivent se mettre d'accord sur un même mot clé diminue la subjectivité dans le choix de ceux-ci et procure donc une annotation de meilleure qualité d'après ce que nous avons dit ci-dessus. Cette technologie a été rachetée récemment par Google qui l'utilise déjà dans son moteur de recherche d'image. Je n'ai hélas pas pu obtenir le nombre

[5] http ://www.flickr.com
[6] http ://www.espgame.org/

d'images qui ont été ainsi annotées, et ne peux pas juger du succès de cette méthode d'annotation.

1.4.2 Annotation automatique

L'annotation automatique est une alternative à l'annotation manuelle qui utilise le contenu de l'image. Elle est plus rapide et (donc) moins coûteuse, et peut également être plus homogène et systématique. Les travaux pionniers dans ce domaine sont la recherche d'images par similarité qui s'est développée dans les années 1980. Pour ces recherches d'images, des descripteurs de bas niveau (couleur, texture, forme) sont calculés à partir du contenu de l'image, et sont utilisés pour comparer deux images. En utilisant des méthodes d'apprentissage, il est possible d'utiliser ces descripteurs pour apprendre des concepts et générer des mots clés automatiquement. Un état de l'art est présenté en section 2.3. En revanche, le niveau d'abstraction obtenu par les méthodes d'annotation automatique est encore très faible et ne permet donc pas des requêtes de niveau 3 (selon la définition donnée en partieřefintro-requetes) D'autre part, l'annotation automatique est un problème difficile car il faut franchir l'écart, appelé fossé sémantique, qu'il y a entre savoir calculer des descripteurs bas niveau et pouvoir identifier les objets présents dans l'image.

Liu et al. [82], dans leur état de l'art sur les recherches d'images de haut niveau sémantique par le contenu, ont groupé les méthodes pour réduire le fossé sémantique en cinq catégories :

1. utiliser une ontologie d'objets pour définir des concepts de haut niveau,
2. utiliser des techniques d'apprentissage automatique (*machine learning*) pour associer des descripteurs de bas niveau avec les concepts de la requête,
3. introduire le retour de pertinence (*relevance feedback*) dans la boucle de recherche pour un apprentissage continu de l'intention de l'utilisateur,
4. générer un modèle sémantique (*semantic template*) qui gère la recherche d'images de haut niveau (ces modèles sont construits à partir d'un retour de pertinence),
5. utiliser à la fois le contenu visuel de l'image et les informations textuelles obtenues sur les pages Internet pour la recherche d'images sur Internet.

Par rapport à l'expérience de Sunderland [129] décrite précédemment, les informations données par le profane et l'historien d'art ne peuvent pas ou difficilement être extraites automatiquement en s'appuyant uniquement sur le contenu de l'image. Par exemple, l'auteur et la date de l'œuvre ne peuvent être connus que si le système a une base de données de peintures dont celle-ci fait partie, et a reconnu la peinture dont il s'agit. Ce que cherchent à obtenir les systèmes actuels d'annotation automatique d'images est plutôt quelque chose de similaire à la description de l'enfant de 12 ans. C'est-à-dire que l'on souhaiterait être capable déjà de pouvoir décrire les objets de l'image « il y a une femme, un enfant, un bol » et les relations simples entre les objets « la femme est à genou et serre l'enfant dans ses bras, un jeune garçon tient un bol dans ses mains », sans interprétation de ce que cela signifie « c'est la Famille Sainte, ... ».

Même l'obtention de ces descriptions reste aujourd'hui un problème très difficile pour les systèmes d'annotation automatique d'images.

1.4.3 Approche proposée dans cette thèse

Nous souhaitons dans cette thèse proposer des méthodes introduisant la sémantique des objets lors de l'annotation automatique d'images afin de réduire le fossé sémantique. Nous souhaitons également nous orienter vers une annotation complètement automatique de l'image, avec un temps d'annotation permettant de le faire à la volée (ou en ligne), c'est-à-dire de l'ordre de quelques secondes.

Si l'on se reporte aux cinq manières énoncées par Liu et al. [82] pour réduire ce fossé et énumérées ci-dessus, les points 3 et 4 concernent le retour de pertinence qui nécessite qu'un utilisateur valide les réponses données par le système. Nous excluons ces deux points de notre étude comme ne permettant pas un système « complètement automatique ». En revanche, les trois autres points seront étudiés au cours de cette thèse, dans différentes parties.

L'approche proposée dans cette thèse est illustrée sur la figure 1.2. Nous pouvons la voir comme constituée de deux grandes parties :

– l'apprentissage de concepts tels que les scènes, les fonds et les objets,
– l'annotation d'une nouvelle image, en fonction des concepts qui ont été appris.

Pour des raisons d'organisation, notamment dû au fait que ces deux parties ne sont pas indépendantes mais au contraire très liées, la thèse n'est pas divisée tout à fait suivant cette structure.

Nous présentons tout d'abord dans le chapitre 2 un état de l'art des différents travaux sur lesquels s'appuient notre travail, et/ou qui concernent l'annotation automatique d'images. Cet état de l'art décrit d'abord sommairement les principaux algorithmes permettant de faire de la classification multiclasse qui ont été appliqués avec succès en traitement d'images (2.1). Ensuite sont présentés le calcul et l'utilisation des relations spatiales pour la compréhension d'images (section 2.2), que j'utiliserai par la suite. Enfin, la majeure partie de cet état de l'art est consacrée à la description des différentes orientations proposées pour faire de l'annotation automatique d'images (section 2.3).

Les différents descripteurs auxquels nous avons recours pour extraire des informations de bas niveau liées à la couleur, la texture et la forme sont détaillés dans le chapitre 3. Ces descripteurs nous permettent d'effectuer d'une part de la classification de scènes, chapitre 4, et d'autre part de la reconnaissance d'objets, chapitre 6.

Nous proposons une approche de bas en haut pour l'annotation automatique d'images. Initialement, nous supposons qu'une image inconnue est soumise au système qui doit la décrire en n'utilisant que son contenu, c'est-à-dire les informations de bas niveau issues des descripteurs. La première étape consiste à avoir une idée de quel type d'image il s'agit : est-ce une photo, une peinture, un dessin ? Si c'est une photo, est-elle prise à l'extérieur ou à l'intérieur, en ville ou en campagne, de jour ou de nuit ? Afin d'obtenir ces informations, nous avons mis en place une classification hiérarchique représentant les principaux types de scènes que nous avons observés dans les bases d'images, et notamment sur Internet. Par rapport à ce qui a été fait dans la littérature, nous proposons

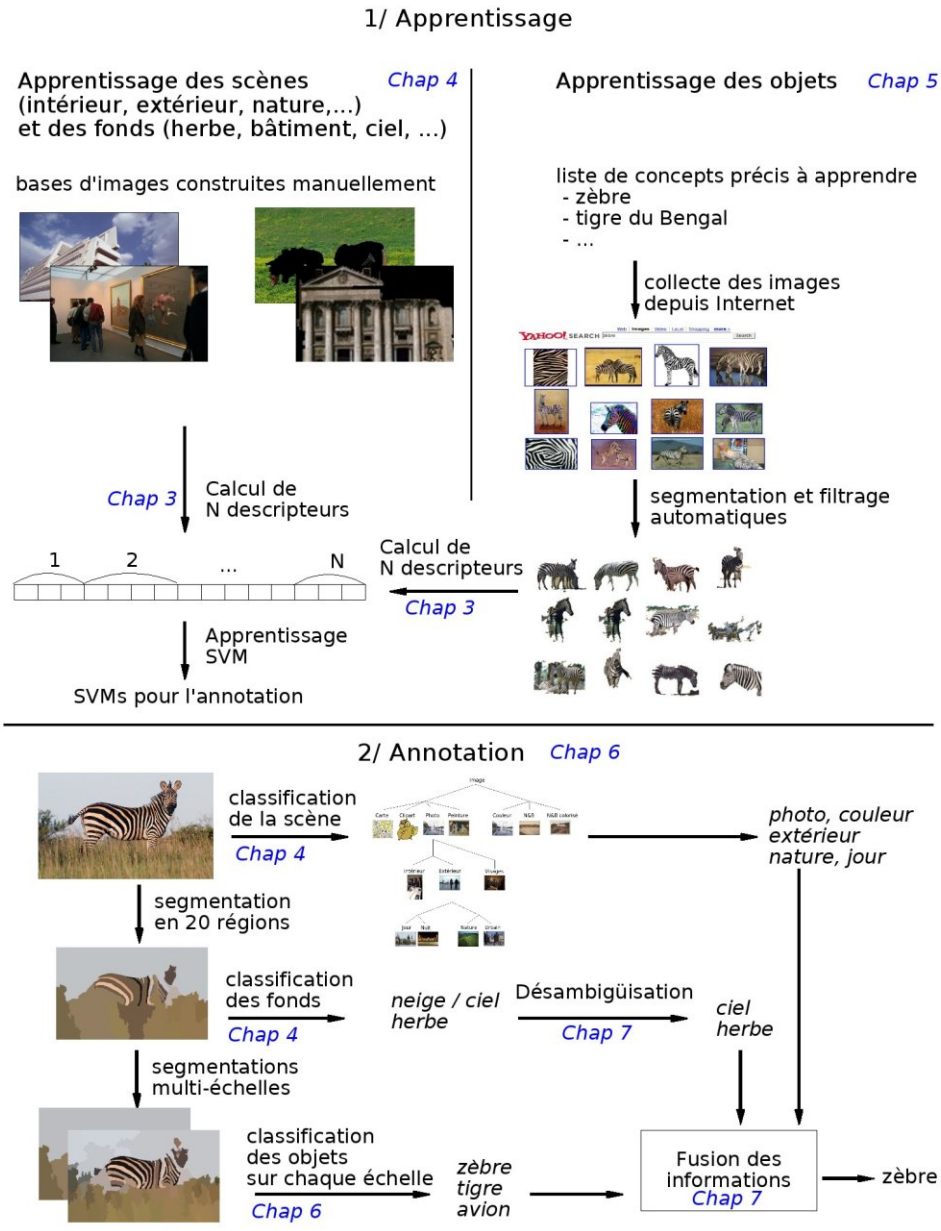


FIG. 1.2 – Schéma présentant l'approche proposée dans cette thèse.

plus de classes afin de traiter une plus grande variété de types d'images. Nous avons également évalué pour la première fois une telle classification dans le cadre d'une campagne d'évaluation. En plus de la classification de scènes, nous proposons d'identifier les fonds les plus courants dans les images reconnues comme des photographies d'extérieur, tels que le ciel, les arbres, l'eau, etc. Ces classifications fournissant une bonne connaissance de l'environnement de la scène sont présentées dans le chapitre 4.

Nous nous attaquons ensuite au difficile problème de la reconnaissance des objets de l'image. En amont de ce problème se situe le manque actuel de bases de données annotées manuellement qui pourraient permettre de reconnaître n'importe quel objet. Nous étudions chapitre 5 la possibilité et la viabilité de considérer Internet comme source d'images pour construire des bases d'apprentissage. Le principal obstacle à cela est que les images retournées par les moteurs de recherche d'images sur Internet ne sont pas toutes pertinentes. Notre contribution dans ce domaine porte principalement sur le filtrage automatique de ces images pour enlever les intrus, et sur la segmentation automatique de ces images, les deux problèmes étant liés dans notre approche. Cela nous permet de créer des bases de données d'apprentissage d'une manière complètement automatique.

La détection d'objets est introduite au chapitre 6 autour de deux expériences. La première décrit la reconnaissance d'animaux utilisant pour l'apprentissage des images segmentées manuellement, et pour la reconnaissance une segmentation automatique multi-échelle. Deux choses sont donc à évaluer : d'une part la qualité des descripteurs extraits et de l'algorithme d'apprentissage, d'autre part la qualité de la segmentation. La deuxième expérience, plus originale, vise à évaluer la qualité des bases d'images construites et segmentées automatiquement à partir d'Internet en comparaison avec celles qui sont construites et segmentées manuellement.

Enfin, le chapitre 7 s'intéresse à deux types de désambiguïsation permettant d'améliorer la reconnaissance d'objets en faisant des post-traitements après l'algorithme de classification : la désambiguïsation par relations spatiales et par le contexte. La désambiguïsation par relations spatiales concerne la classification de fonds. Parmi les fonds classés, certains sont souvent confondus car ils sont visuellement très proches, comme par exemple l'eau, le ciel et la neige, ou causent des fausses détections sur des régions qui concernent par exemple un objet. Nous montrons comment les relations spatiales peuvent être combinées avec la connaissance des positions attendues de ces régions pour résoudre certains cas de fonds reconnus pour un autre ou de fausses détections. La désambiguïsation par le contexte est appliquée à la reconnaissance d'animaux mais pourrait être appliquée à n'importe quel objet. Nous étudions comment la connaissance des co-occurrences des objets et des scènes peut servir pour améliorer la qualité de la reconnaissance des objets. Nous comparons deux méthodes d'extraction automatique de ces co-occurrences : à partir d'une base d'images annotées et à partir d'un corpus textuel.

Nous concluons, chapitre 8, en dégageant quelques pistes de recherche intéressantes autour de ces travaux que je n'ai pas eu le temps d'aborder en ces trois années de thèse.

Chapitre 2

État de l'art

Il m'arrive d'avoir certaines idées avant les autres. Mais ce n'est pas moi qui suis en avance, ce sont eux qui sont en retard.
Alfred Capus, Les Pensées

Nous proposons un état de l'art constitué de trois parties où sont présentés
- les principaux algorithmes de classification multiclasse utilisés dans le domaine de l'annotation automatique d'images,
- le calcul et l'utilisation des relations spatiales,
- une tentative de répartition en plusieurs groupes des différentes méthodes proposées dans la littérature pour aborder le problème de l'annotation automatique d'images.

La troisième partie sur les différentes approches existantes occupe la majeure partie de cet état de l'art. La partie sur la présentation des algorithmes de classification est utile, d'une part, pour comprendre cette troisième partie et, d'autre part, servira de base pour discuter nos choix algorithmiques. Les relations spatiales ont leur place dans cette état de l'art car nous les utilisons pour introduire de la sémantique dans l'annotation d'images. Nous détaillons cet aspect au chapitre 7.

2.1 Algorithmes de classification multiclasse pour l'annotation automatique d'images

Dans cette section sont présentées des techniques qui permettent une classification multiclasse, soit intrinsèquement (k plus proches voisins,...) soit par extension d'algorithmes de classification binaire (séparateur à vaste marge,...). Le choix des techniques

décrites ici a été principalement fait en considérant celles qui ont été le plus souvent appliquées au domaine de l'annotation automatique.

Le problème de la classification multiclasse peut être décrit par le formalisme suivant : on souhaite effectuer une classification en K classes, et on dispose de N observations d'apprentissage dont la classe est connue. On peut écrire ces données formellement sous la forme $S = \{(x_1, y_1), (x_2, y_2), ...(x_N, y_N)\}$ où les x_i appartiennent à un espace $X \subset \mathbb{R}^m$ et où les $y_i \in Y = \{1...K\}$ sont les classes. Dans le cas, par exemple, où l'on s'intéresserait à la classification d'images d'animaux, x est un vecteur de caractéristiques (texture, couleur, forme) résultant de la description d'une image et y est la classe de l'image (cheval, chien, éléphant, etc.).

Un classifieur multiclasse est une fonction $f : X \rightarrow Y$ qui à un élément x de X associe une classe de Y. La tâche est de trouver une définition pour la fonction inconnue f, étant donné l'ensemble d'apprentissage. Bien que beaucoup de problèmes dans le monde réel soient des problèmes multiclasse ($K > 2$), beaucoup d'algorithmes de classifications sont conçus initialement pour traiter un problème binaire. Il existe heureusement plusieurs techniques pour créer un classifieur multiclasse à partir de plusieurs classifieurs binaires.

2.1.1 k plus proches voisins

Cette méthode est l'une des plus simples pour la classification multiclasse. Dans l'ensemble de données X, il faut établir une fonction distance d entre deux éléments. Un nouvel élément x est alors classifié dans la classe la plus représentée parmi ses k plus proches voisins, où la notion de proximité est donnée par la distance d. Il est bien sûr possible de complexifier cette méthode, par exemple, en donnant un poids plus important aux votes des éléments les plus proches.

Cette méthode ne donne en général pas les meilleurs résultats parmi les méthodes proposées dans cette section, mais elle a l'avantage d'être facile à analyser : on peut visualiser les plus proches voisins et en déduire quelles sont les images qui posent problème dans la classification. Un inconvénient majeur de cette technique de classification est le fait de devoir choisir une distance. Il existe notamment des dizaines de distances différentes couramment utilisées dans le domaine de la recherche d'images par similarité (euclidienne, Mahalanobis, *Earth Mover's Distance*, ...) et il est rarement possible de décider quelle est la meilleure théoriquement ou la mieux adaptée aux données étudiées.

2.1.2 Classification bayésienne

La méthode de classification bayésienne s'appuie sur la formule de Bayes pour deux variables aléatoires x et y :

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

Dans la cas de N classes y_1, ..., y_N, la formule de Bayes donne la probabilité que x appartienne à la classe y_k :

$$P(y_k|x) = \frac{P(x|y_k)P(y_k)}{P(x)}$$

Il faut ensuite modéliser la fonction de probabilité $P(x)$, par exemple par une fonction gaussienne, dont on estime les paramètres à partir de l'ensemble d'apprentissage. Cette estimation se fait souvent à l'aide de l'algorithme d'espérance-maximisation [29].

Les classifieurs bayésiens ont notamment été utilisés par Vailaya et al. [142, 140, 139, 141] pour faire de la classification de scène. Leurs travaux seront détaillés par la suite.

2.1.3 Arbres de décision

Un arbre de décision est un ensemble de tests appliqués les uns après les autres où le résultat d'un test est utilisé pour déterminer le test qui sera ensuite appliqué. Chaque test essaie de séparer différentes classes, et le dernier test donne idéalement la classe de l'élément. La suite de tests à appliquer est apprise automatiquement à partir de la base d'apprentissage, à l'aide d'algorithmes divers et variés.

La thèse de Raphaël Marée [90] donne un état de l'art des techniques actuellement les plus utilisées en arbres de décisions, et s'intéresse à la classification multiclasse automatique d'images par arbres de décision. Ses tests sur la base de données Corel 1000 (10 classes de 100 images) donnent 15,9% d'erreur de classification, ce qui égale le meilleur taux présenté dans [30] qui compare 9 algorithmes de la littérature sur cette même base.

2.1.4 Séparateurs à vaste marge

Les séparateurs à vaste marge, ou machines à vecteur support (*Support Vector Machines, SVM*) sont un algorithme de classification binaire développé par Vapnik [143].

Le principe des séparateurs à vaste marge est de trouver l'hyperplan séparateur d'un ensemble de données correspondant à deux classes qui maximise la marge entre ces deux classes. Le principe d'hyperplan à marge maximale est illustré sur la figure 2.1 pour un espace à deux dimensions.

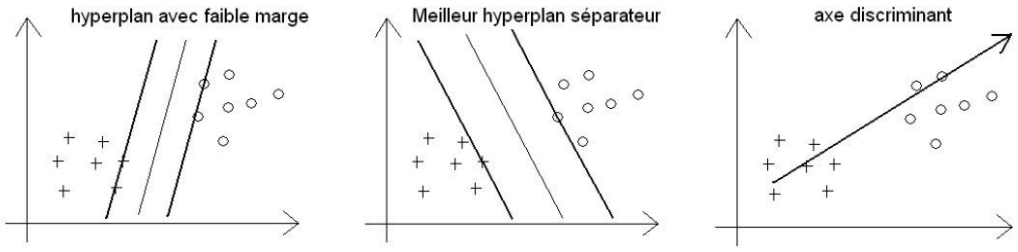


FIG. 2.1 – Illustration de la recherche de l'hyperplan à marge maximale dans un espace à deux dimensions.

En pratique, il est possible d'obtenir de meilleurs résultats en changeant la dimension de l'espace des données grâce à une fonction $\phi : X = \mathbb{R}^m \longrightarrow \mathbb{R}^p$ et en cherchant le meilleur hyperplan séparateur dans ce nouvel espace qui a souvent une dimension p plus grande que la dimension m de l'espace initial. Il n'est pas commode de chercher à définir directement la fonction ϕ. Dans le cas des séparateurs à vaste marge, heureusement, seul

le produit scalaire entre les éléments intervient pour trouver l'hyperplan. C'est ce qu'on appelle l'astuce du noyau (*kernel trick*) qui consiste à ne définir que le noyau :

$$k(x_{i,j}) = \phi(x_i).\phi(x_j)$$

Les noyaux gaussiens sont par exemple fréquemment utilisés :

$$k(x_{i,j}) = exp(-\gamma \|x_i - x_j\|^2)$$

La classification d'un élément x se fait ensuite simplement par une fonction f :

$$f(x) = signe(\sum_{i=1}^{N} y_i \alpha_i k(x, x_i))$$

où les x_i sont les vecteurs d'apprentissage, de classe y_i. Les coefficients α_i sont appris pour maximiser la classification sur les données d'apprentissage. Les vecteurs x_i dont les coefficients α_i sont non nuls sont appelés les vecteurs supports.

À partir des classifieurs binaires que nous pouvons ainsi entraîner, nous souhaiterions désormais faire de la classification multiclasse. Parmi les méthodes existantes pour transformer un problème multiclasse en plusieurs problèmes binaires plus simples, on distingue trois principaux groupes : un contre tous (*one-vs-all, OVA*), un contre un (*one-vs-one, OVO*) et le code correcteur d'erreur (*error correcting output code, ECOC*).

Un contre tous. Cette méthode a été proposée indépendamment par plusieurs auteurs [24] [3]. Elle consiste à construire K classifieurs binaires : pour chaque classe i, un classifieur binaire f_i est appris en utilisant les exemples de la classe i comme exemples positifs, et les exemples de toutes les autres classes comme exemples négatifs. Ensuite, pour combiner ces classifieurs, on garde simplement la classe qui a été prédite avec le plus grand pourcentage de confiance. Le classifieur multiclasse f est alors défini par :

$$f(x) = argmax_{j \in \{1,\dots,K\}} f_j(x)$$

Un contre un. Cette méthode fut proposée d'abord par Knerr et al. [70] puis améliorée par Hastie et al. [61]. $K(K-1)/2$ classifieurs binaires sont construits de la manière suivante : pour toute paire $i \neq j$, un classifieur f_{ij} est appris en utilisant comme exemples positifs ceux de la classe i, et comme négatifs ceux de la classe j, sans utiliser les autres classes. Il y a différentes méthodes ensuite pour combiner les classifieurs obtenus, la plus commune étant un vote [52] : quand on classifie un nouvel élément, chaque classifieur binaire vote pour une des deux classes qu'il a apprise. Une autre approche pour combiner ces classifieurs est le graphe acyclique orienté de décision (*Decision Directed Acyclic Graph, DDAG*) [113]. Cette méthode construit un graphe binaire acyclique enraciné (*rooted binary acyclic graph*) à partir des classifieurs binaires. Les nœuds sont arrangés en un triangle avec le nœud racine au sommet, deux nœuds au deuxième étage, quatre nœuds au troisième étage, etc. Pour évaluer un DDAG sur un élément x, on part du nœud racine, on évalue la fonction binaire, puis on fait de même sur l'un ou l'autre des deux nœuds fils suivant le résultat. La réponse finale est la classe donnée par le nœud feuille, à l'étape finale.

Code correcteur d'erreur. Cette méthode a été développée par Dietterich et Bakiri [31]. Elle utilise une matrice M de taille $K * F$ qui prend ses valeurs dans $\{-1, 1\}$ où F est le nombre de classifieurs binaires et K le nombre de classes. La $j$ème colonne de la matrice, correspondant au $j$ème classifieur, induit une partition des K classes en deux métaclasses : les classes positives et les classes négatives. Une classe i est positive pour le $j$ème classifieur binaire f_j si $M_{ij} = 1$, négative sinon. Pour tout élément x d'une classe i, la fonction de décision f que les auteurs ont définie, utilisant la distance de Hamming entre chaque ligne de la matrice M et la sortie des F classifieurs, est donnée par :

$$f(x) = argmin_{i \in 1,\ldots,K} \sum_{j=1}^{F} \left(\frac{1 - sign(M_{ij} f_j(x)}{2} \right)$$

Cette fonction retourne donc la classe d'un élément x selon les valeurs retournées par les classifieurs binaires $f_j(x)$.

D'autres approches utilisent d'autres mesures de distance entre les sorties des classifieurs et les lignes des matrices, ou des méthodes plus sophistiquées [150]. Les résultats de cette approche dépendent fortement de l'indépendance des classifieurs, comme le font remarquer Kong et al. [71], sans laquelle l'approche de correction d'erreur échouerait.

Hsu et al. [65] ont mené une comparaison des principales méthodes et ont conclu que la méthode « un contre un » est parmi les meilleures en termes de temps d'apprentissage et de performance de la classification. En effet, même si plus de SVM sont construits dans « un contre un », le temps total d'apprentissage est moins long que pour « un contre tous » (3 fois moins long en moyenne lors de nos tests), car les ensembles de données utilisés pour chaque apprentissages sont plus petits.

Il est également possible de combiner ces méthodes. Ainsi, García-Pedrajas et Ortiz-Boyer [54] montrent une amélioration des résultats en combinant les stratégies « un contre un » et « un contre tous ».

2.1.5 Apprentissage d'instances multiples

Le *Multiple Instance Learning* [91] est une technique visant à apprendre à partir de données dont les classes ne sont pas connues pour chaque élément individuellement, mais plutôt pour des groupes d'éléments, pour lesquels il est connu si ce groupe possède ou non des éléments d'une classe donnée. Les données d'apprentissage sont réparties en plusieurs groupes, et deux types de groupes sont distingués :

- les groupes dont au moins un élément est positif, i.e. appartient à la classe à reconnaître,
- les groupes dont tous les éléments sont négatifs.

Dans le cas de la reconnaissance d'objets dans les images par exemple, les éléments sont les régions d'une image, et un groupe d'éléments est l'image elle-même. Cet apprentissage est utilisable dans le cas où l'on a l'information de la classe de l'image sans avoir l'information de la classe de chacune des régions séparément, ce qui est bien souvent le cas.

2.1.6 Boosting

Cet algorithme permet de choisir, parmi un ensemble souvent très grand de classifieurs donnant des résultats médiocres, un sous-ensemble restreint, ainsi qu'une combinaison linéaire de ces classifieurs permettant de construire un classifieur ayant de bonnes performances. Les premiers algorithmes de boosting répondant à ce problème ont été ceux développés par Freund [51] et Schapire [120] en 1990.

Viola et Jones [144] ont eu l'idée d'appliquer cette technique pour permettre la détection de n'importe quel objet dans une image et ont démontré son efficacité notamment pour la détection de visages. Ils utilisent une cascade de classifieurs afin d'optimiser les résultats et le temps de calcul.

Lienhart et al.[79] ont étudié les influences des paramètres dans l'adaboost en cascade proposé par Viola et Jones. Ils proposent aussi quelques améliorations, dont l'utilisation de descripteurs tournés à 45 degrés. Les résultats sont plus rapides et meilleurs que ceux de Viola Jones, pour une fenêtre optimale de 20x20.

Le boosting a également été utilisé par Ferencz et al. [46] pour essayer d'apprendre à reconnaître une classe d'objets à partir d'un seul représentant de cette classe.

2.1.7 Hiérarchie de classifieurs

Le problème ici est le suivant : on dispose d'un ensemble de classifieurs, et on aimerait savoir dans quel ordre les utiliser pour obtenir une classification optimale. Notamment, le choix du classifieur suivant pourra dépendre de la réponse donnée par le classifieur précédent. Cela revient à construire un arbre de décision, où chaque nœud est un classifieur différent. Imaginons par exemple le cas de 5 classes A, B, C, D, E. Le premier classifieur C_1 appliqué permet de faire la différence entre les ensembles $A \cup B$, $C \cup D$ et E. S'il répond $A \cup B$, nous utiliserons un classifieur C_2 capable de différencier A et B. S'il répond $C \cup D$, nous ferons appel à un autre classifieur C_3 permettant de séparer C et D. S'il répond E, il n'est pas nécessaire de continuer la classification.

Divers méthodes existent pour déterminer une bonne hiérarchie de classifieurs à utiliser, mais il est souvent très difficile de trouver la hiérarchie optimale (pour une base de données déterminée). J.M. Martínez-Otzeta et al. [92] proposent de recourir aux algorithmes génétiques pour trouver une bonne hiérarchie de classifieurs et montrent qu'ils obtiennent des résultats similaires ou meilleurs que ceux obtenus par *bagging* d'arbres de décision ou par *boosting*.

2.1.8 Indexation sémantique latente

L'indexation sémantique latente (*latent semantic indexing*, LSI) a été introduite par Dumais et al [34, 33] en 1988 pour le domaine de la recherche textuelle. Sa capacité d'induire et de représenter les sens des mots et les documents suivant leur utilisation a rendu la technique populaire et fourni un outil puissant pour l'indexation et la recherche textuelle automatique. Elle permet une recherche selon le contenu conceptuel d'un document au lieu d'une recherche au niveau des termes, ce qui permet une recherche plus

sémantique.

La justification de cette technique était de proposer une solution au problème de la synonymie et de la polysémie. Par exemple, si la requête contient les mots *ordinateur* et *humain*, on souhaite pouvoir retrouver un document qui contiendrait *interface utilisateur* même s'il ne contient pas les deux mots ci-dessus. La technique proposée jusqu'alors était de construire une liste de synonymes pour *ordinateur* et *humain*, et de poser autant de requêtes qu'il y a de synonymes, ce qui n'est pas efficace, à la fois en temps de calcul, mais également en qualité des résultats [34]. L'indexation sémantique latente construit d'abord une matrice de co-occurrence, puis utilise une décomposition en valeurs singulières pour réduire cet espace et obtenir un espace sémantique, plus petit que l'ensemble des mots, dans lequel les mots synonymes correspondent au même élément. Cette méthode permettra par exemple de deviner que les mots *voiture* et *automobile* sont synonymes, même s'ils ne sont pas présents dans les mêmes textes, mais parce qu'ils ont le même contexte : ils sont voisins des mêmes mots. À l'inverse, un mot polysémique, qui a par conséquent plusieurs contextes d'utilisation différents, pourra correspondre à plusieurs éléments dans l'espace sémantique. Lors de l'indexation ou de la requête, chaque mot est traduit dans ce niveau sémantique, ce qui donne de meilleurs résultats que lorsque la recherche est effectuée au niveau des mots. L'un des défauts de cette technique, cependant, est qu'elle ne fait pas la différence entre synonymes et antonymes.

2.2 Relations spatiales

Nous utilisons dans nos travaux les relations spatiales comme outil permettant de désambiguïser les fonds (ciel, eau, etc.) que nous reconnaissons dans les images. Pour cette raison, nous faisons ici un bref état de l'art de l'utilisation des relations spatiales en traitement d'image.

Les relations spatiales relatives ont été étudiées principalement dans le domaine de l'intelligence artificielle. Dans le domaine du traitement d'images, elles sont très peu appliquées. Les principales applications que l'on peut trouver sont pour la reconnaissance de structures à partir de modèles, notamment dans des images médicales, ou alors pour faire de la description linguistique d'images.

Dès 1975, J. Freeman avait dénombré treize relations spatiales permettant de décrire la position relative d'une région par rapport à une autre, en deux dimensions. Ces relations peuvent se classer en trois groupes : les relations topologiques : *à l'intérieur de*, *à l'extérieur de*, *touche*, *entre* ; les relations liées à la distance : *loin de*, *près de* ; et les relations directionnelles : *au-dessus de*, *en-dessous de*, *à gauche de* et *à droite de*.

Nous nous intéressons particulièrement aux relations topologiques et directionnelles pour lesquelles nous souhaitons avoir un pourcentage de confiance pour chaque relation : dans quelle mesure un objet est-il à l'intérieur ou à gauche d'un autre ?

2.2.1 Calcul des relations spatiales

Beaucoup de techniques ont été proposées afin de calculer des relations spatiales entre régions, dont les histogrammes d'angles, les histogrammes de force, et des méthodes utilisant la morphologie mathématique.

La méthode construisant un **histogramme des angles** a été présentée par Miyajima et al. [100]. Elle permet de calculer les relations du type *au-dessus de*, *en-dessous de*, *à gauche de* et *à droite de*. Un histogramme d'angle est calculé entre deux régions en considérant toutes les paires de points possibles. Pour chaque paire, l'angle entre le segment et l'axe horizontal (figure 2.2) est reporté dans l'histogramme.

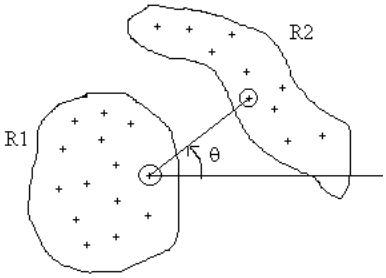


FIG. 2.2 – Relation spatiale entre deux points de deux régions.

Cet histogramme est ensuite normalisé et multiplié par une fonction floue qui est une fonction cosinus carré centrée en 0 radian (respectivement $\pi/2$, π, et $3\pi/2$) pour obtenir le taux (entre 0 et 1) avec lequel la relation *à droite de* (respectivement *au-dessus de*, *à gauche de*, et *en-dessous de*) est vérifiée. La somme de ces taux pour les quatre relations spatiales vaut 1. Cette fonction est représentée sur la figure 2.3.

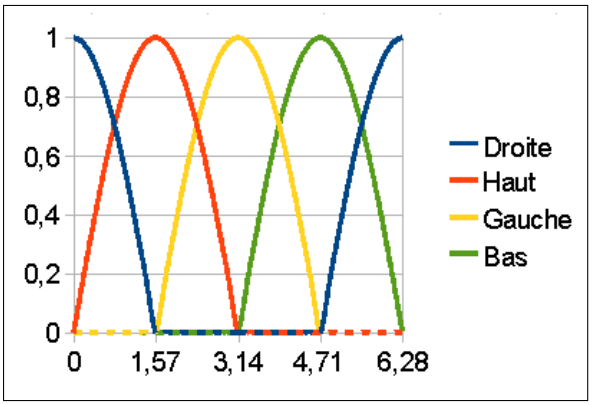


FIG. 2.3 – Fonctions floues utilisées pour obtenir le taux de vérification des relations spatiales pour les quatre directions spatiales.

Par exemple, soit h l'histogramme des angles, la relation « R_2 est à droite de R_1 » est vérifiée avec le taux $\Re_{droite}$ défini par :

$$\Re_{droite} = \sum_{\theta=-\pi/2}^{\pi/2} h(\theta) * cos^2(\theta)$$

La relation obtenue est alors celle de R_2 par rapport à R_1 : $\Re_{droite} = 1$ signifie que R_2 est à droite de R_1, et que R_1 est à gauche de R_2.

Les **histogrammes de forces** ont été définis par Matsakis dans sa thèse de doctorat [93]. Ils sont illustré sur la figure 2.4.

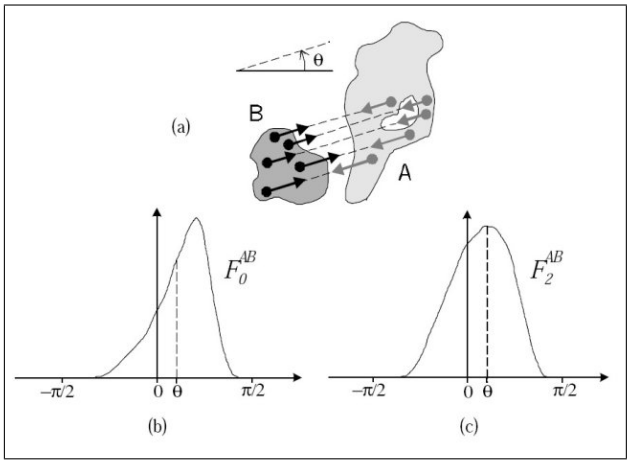


FIG. 2.4 – Schéma illustrant les histogrammes de force, issu de [94].

Soient A et B deux objets dont on cherche à évaluer les relations spatiales relatives directionnelles. L'histogramme calcule, pour chaque angle, une valeur $F^{AB}(\theta)$ qui additionne des contributions élémentaires provenant de chaque couple constitué d'un point de A et un point de B faisant un angle θ avec l'axe horizontal. La contribution de ce couple décroît avec la distance d entre les deux points considérés et est proportionnelle à $\frac{1}{d^r}$ où r est un entier qui varie, permettant de construire un histogramme de force pour chaque valeur de r. Les histogrammes correspondant à $r = 0$ et à $r = 2$ sont représentés sur la figure 2.4. $r = 0$ revient à considérer que tous les points ont la même importance, quelle que soit leur distance et correspond à la méthode développée par Miyajima et al. [100] décrite ci-dessus. $r = 2$ correspond au modèle d'attraction gravitationnelle et donne plus d'importance aux couples qui sont les plus rapprochés.

Il est possible de calculer d'autres relations plus complexes telles que « est à l'intérieur de », « est à l'extérieur de » et « est entre » [94], mais nous n'utiliserons pas ces relations.

Bloch et al. [13] offrent une bonne comparaison des différentes techniques de calcul de relations spatiales directionnelles floues utilisées dans la littérature. Les méthodes comparées ici sont l'histogramme des angles, la méthode des centroïdes, les histogrammes de force, une méthode fondée sur la projection, et une approche utilisant la morphologie mathématique.

2.2.2 Relations spatiales pour l'indexation et la classification

Les méthodes décrites ci-dessus permettent de calculer des relations spatiales relatives entre deux ou plusieurs régions. Il n'est cependant pas trivial d'intégrer ces relations spatiales dans l'indexation d'images afin de permettre de faire de l'apprentissage automatique.

Omhover et al. [109] intègrent les relations spatiales dans la recherche d'images par similarité. L'utilisateur sélectionne plusieurs régions d'une image requête et l'algorithme recherche dans une base d'images déterminée une image contenant des régions similaires qui ont également un agencement spatial proche de celui de l'image requête. Trois axes de relations spatiales ont été utilisés : la connexité, la relation « gauche / droite » et la relation « au-dessus / en-dessous ». Cela donne un histogramme qu'ils peuvent comparer entre deux couples de régions pour calculer leur similarité au niveau des relations spatiales. Leur mesure finale de distance entre deux images considère tous les couples possibles de l'image question et des images de la base pour prendre en compte en même temps la similarité spatiale et la similarité visuelle.

Datta et al. [27] proposent un modèle dénommé *Structure-Composition* fonctionnant de la manière suivante :

- l'image est d'abord quantifiée en un nombre fixé N_c de couleurs en appliquant l'algorithme des K-moyennes sur l'histogramme des couleurs dans l'espace LUV sur toutes les images d'apprentissages. La même quantification est appliquée à toute les images, et le nombre de régions alors obtenues dans l'image est variable ;
- ensuite, pour chaque couple de couleurs (C_i, C_j), une fonction $\Delta(C_i, C_j)$ représente la longueur de bords communs entre une région de couleur C_i et une région de couleur C_j ;
- $\Delta(C_i, C_j)$ est normalisé par la somme des périmètres $\Theta(C_i)$ des régions de couleur C_i. Le rapport $f(i,j) = \Delta(C_i, C_j)/\Theta(C_i)$, compris entre 0 et 1, n'est donc pas symétrique. Ce rapport vaut 1 si les régions de couleur C_i sont entourées par des régions de couleur C_j. Il vaut 0 si les régions ne se touchent pas.

Cela fournit un descripteur de taille fixe $(N_c^2 - N_c)$ qui peut être utilisé pour l'apprentissage. Datta et al. choisissent de l'apprendre en le modélisant par une loi bêta définie avec les paramètres (α, β) :

$$f(x; \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

où la fonction gamma est définie par

$$\Gamma(z) = \int_0^{+\infty} t^{z-1}\, e^{-t}\, dt$$

Bosch et al. [14] se servent des relations spatiales pour améliorer la reconnaissance de régions (par exemple : ciel, herbe, route, végétation, terre) dans des scènes d'extérieurs. Ces relations spatiales sont l'adjacence entre régions et la position absolue d'un pixel dans l'image. À partir de l'apparence en texture et en couleur (les détails ne sont pas donnés

sur les descripteurs qu'ils utilisent) ainsi que la position absolue des pixels, une densité de probabilité de forme gaussienne est estimée afin de pouvoir ensuite classer séparément chaque pixel de manière probabiliste en l'une de ces cinq classes. La position absolue d'un objet est exprimée selon la surface des parties de cette objet qui sont présentes en haut, au milieu ou en bas de l'image. Ensuite, les classes assignées à chaque pixel sont affinées en prenant en compte l'adjacence entre les régions ainsi obtenues : si une région inconnue est adjacente à une région identifiée, alors ces deux régions sont comparées en termes de couleur et de texture. Si elles sont suffisamment similaires, elles sont fusionnées et annotées comme la région reconnue. L'approche de Bosch et al., schématisée sur la figure 2.5, leur permet de traiter en même temps les problèmes de la segmentation de l'image, et de la classification de chaque région comme un objet connu ou inconnu. La classification n'utilisant que l'apparence du pixel donne 54,21% de bonnes classifications, alors qu'avec l'inclusion des relations spatiales, ils atteignent 89,87%.

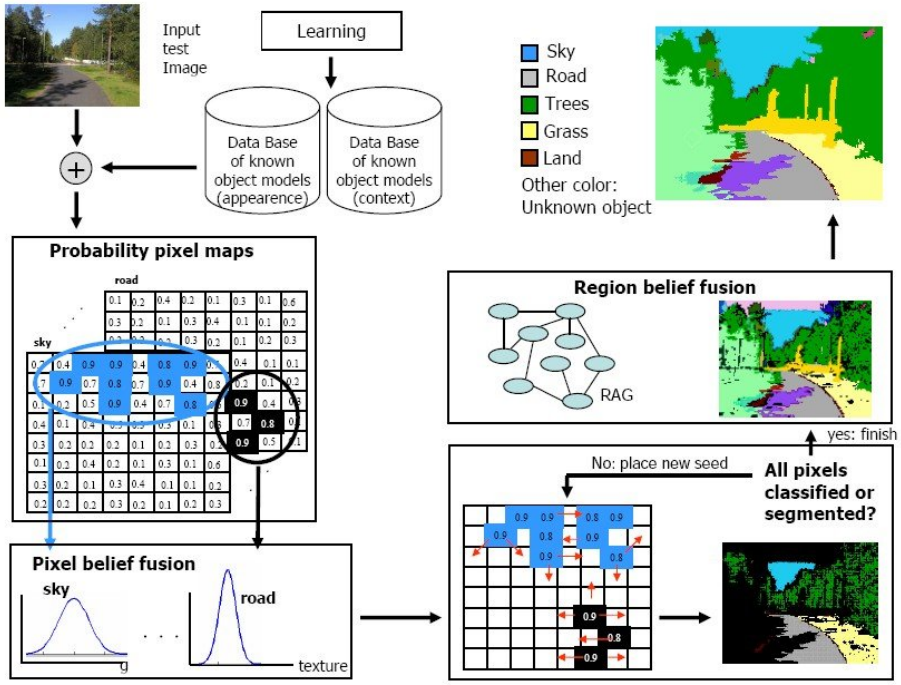


FIG. 2.5 – Approche proposée par Bosch et al. [14] pour améliorer la classification de régions en incluant des relations spatiales absolues et relatives.

2.3 Annotation automatique d'images

Après avoir établi un état de l'art plutôt au niveau algorithmique, d'une part sur les différents algorithmes de classification multiclasse (section 2.1) et d'autre part sur

les relations spatiales (section 2.2), nous passons maintenant à la dernière et principale partie de notre état de l'art concernant les travaux sur l'annotation d'images. Nous chercherons notamment à dégager les types de mots-clés que ces travaux ont tenté d'extraire des images et les méthodes proposées pour y parvenir. Nous verrons que la plupart des méthodes décrites ici font appel à au moins un des algorithmes de classification présentés précédemment. Nous avons choisi d'ordonner les travaux d'après leurs méthodes plutôt que d'après les caractéristiques sémantiques qu'ils extraient, car même si les deux classements sont possibles, une méthode donnée n'est souvent pas spécifique aux mots-clés qu'elle permet de reconnaître et peut donc s'étendre parfois sans effort à d'autres mots-clés. Dès lors, il devient plus difficile de justifier un classement par mots-clés extraits.

Les premières publications sur l'annotation automatique d'images sont apparues récemment, vers 1999, et depuis ce domaine a suscité beaucoup d'intérêt dans la communauté du traitement d'images. Dans sa thèse, S. Tollari [135] (p96) a dressé un tableau comparatif d'une vingtaine de modèles d'annotation automatique proposés entre 1999 et 2006. Nous incluons de nombreux autres travaux dans cette section.

Nous distinguons deux grandes familles d'approches pour traiter le problème de l'annotation automatique. Une première manière de procéder est une approche orientée scène (*scene-oriented approach*). L'idée est de classer la scène dans son ensemble, sans chercher à segmenter l'image. Une seconde méthode consiste à utiliser un algorithme de segmentation d'images pour diviser l'image en un certain nombre de régions de formes plus ou moins irrégulières, et d'essayer d'annoter ces régions.

2.3.1 Approches orientées scène

Je reprends dans cette partie les grandes lignes du classement des travaux proposé dans l'état de l'art « *Which is the best way to organize/classify images by content* » par Anna Bosch, Xavier Muñoz et Robert Marti [16], qui proposent notamment d'organiser les différents travaux selon la technique utilisée pour la classification plutôt que selon les classes considérées, qui est le choix que nous adoptons également comme expliqué ci-dessus.

Les approches orientées scène cherchent à reconnaître le sujet principal d'une image, et/ou sa catégorie sémantique en la considérant comme un tout, sans chercher à reconnaître les objets de l'image individuellement. Des exemples de catégories sémantiques sont montrés sur la figure 2.6.

La reconnaissance de scène peut être vue comme un moyen d'organiser rapidement une grande base de données d'images en plusieurs ensembles plus petits. Elle peut aussi être considérée comme une première approche d'analyse de l'image permettant de guider la reconnaissance d'objets [136].

Ce problème n'est pas aussi ambitieux que le problème plus général de compréhension d'images qui essaye de reconnaître tous les objets d'une image. Les scènes peuvent souvent être classées sans connaître exactement les objets présents dans l'image. Dans certains cas, l'utilisation de critères de bas niveau tels que la couleur et la texture peuvent être suffisants pour faire la différence entre plusieurs scènes. Dans d'autres cas plus complexes, même si la reconnaissance d'objets peut être utile à la classification de scène, il

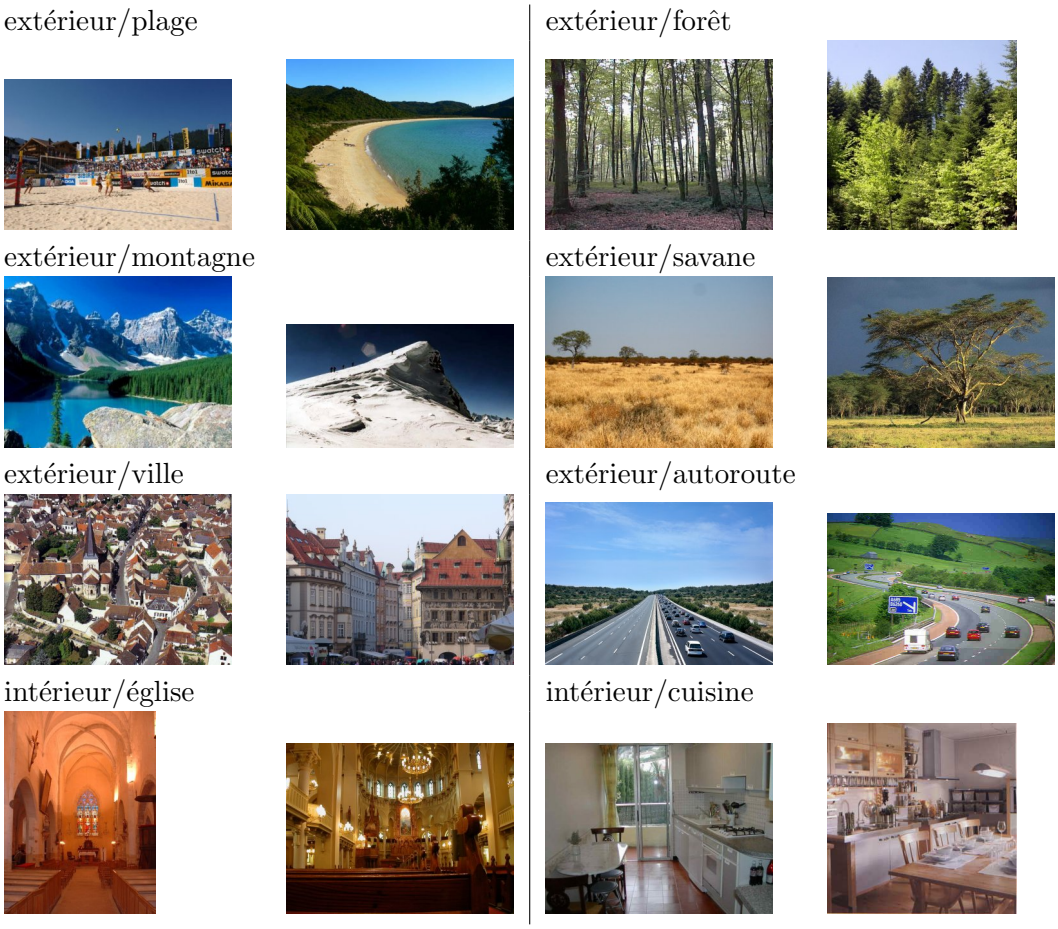


FIG. 2.6 – Exemples de catégories sémantiques d'images typiquement utilisées en classification de scènes.

est probable qu'une reconnaissance imprécise d'un ou deux objets dans l'image suffise sans que l'on ait besoin de reconnaître tous les objets. Par exemple, si une scène contient un arbre en haut de l'image, et de l'herbe en bas, il est possible de faire l'hypothèse que l'image est une scène de forêt, même sans distinguer tous les détails dans l'image [17]. Bosch et al. [16] ont regroupé les techniques de reconnaissance de scènes utilisées dans la littérature en cinq catégories, présentées sur la figure 2.7. Ils proposent également à la fin de leur article un tableau récapitulatif des différentes approches.

La reconnaissance de scènes pour la classification en utilisant des caractéristiques de bas niveau est étudiée en recherche d'images et de vidéos depuis plusieurs années [126]. Les premiers travaux dans ce domaine utilisaient des descripteurs de couleur, de texture et de forme appliqués directement sur l'image entière, avec des méthodes d'apprentissage supervisées pour classifier par exemple les scène d'intérieur, d'extérieur, de ville, de paysage, de coucher de soleil, de forêt, etc. Il y a cependant un fossé sémantique entre les descripteurs de bas niveau et la sémantique de l'image. Des travaux plus récents

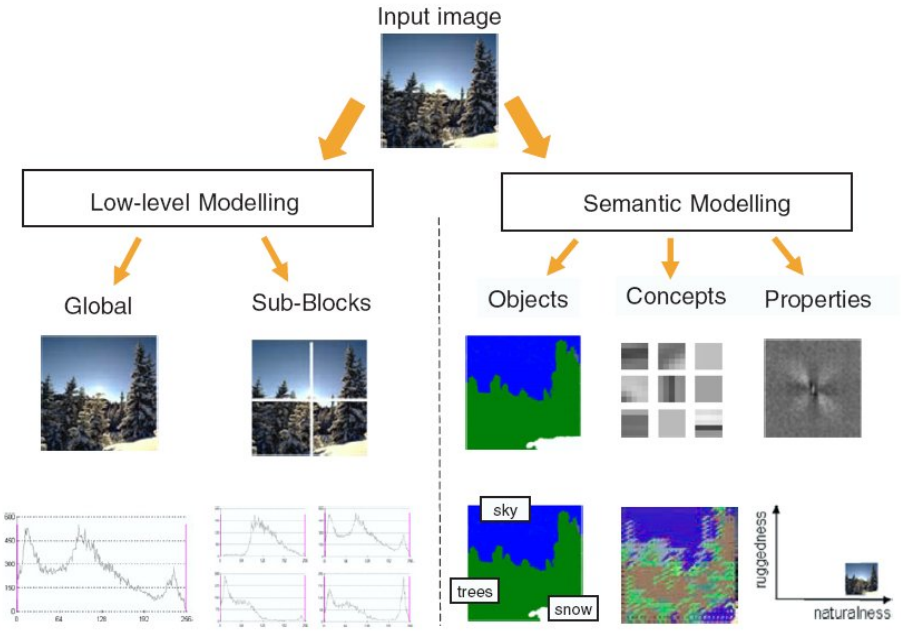


FIG. 2.7 – Présentation des différentes méthodes de reconnaissance de scènes dans la littérature selon Bosch et al. [16].

ont donc proposé d'utiliser une représentation sémantique intermédiaire pour réduire ce fossé, en utilisant la connaissance que nous avons des scènes. Par exemple, une forêt est principalement constituée d'arbres et d'herbe ; une scène de plage contient du sable et de l'eau. Il faut alors chercher à reconnaître les arbres, l'herbe, le sable et l'eau, ce qui est souvent un problème plus simple que de chercher à classifier directement la scène.

La question de savoir si les caractéristiques de bas niveau sont suffisantes pour la classification de scènes est toujours actuellement débattue. Thorpe et al. [134] ont montré que les humains sont capables, pour des images qu'ils n'ont vu que pendant 20ms, de les catégoriser comme contenant ou non un animal ou un véhicule en 150ms avec un très bon taux de réussite. Fei-fei et al. [43] ont ultérieurement remarqué que nous avons besoin de peu ou pas d'attention pour effectuer une catégorisation de scène naturelle aussi rapidement. Ces études sont venues contredire l'opinion qui était alors la plus répandue et qui soutenait qu'il fallait d'abord reconnaître chaque objet pour en déduire la catégorie de la scène, théorie développée notamment par Treisman [138] dans les années 1980.

Apprentissage directement à partir des critères de bas niveau

Ces méthodes considèrent que le type de scène peut être décrit en utilisant uniquement les propriétés de couleur et de texture de l'image. Ce n'est pas sans fondement : par exemple une scène de forêt contient des régions vertes fortement texturées (les arbres), une montagne contient des quantités importantes de bleu (le ciel) et de blanc (la neige)

et une scène urbaine présente des bords verticaux et horizontaux assez marqués.

Parmi les articles qui se sont intéressés à ce sujet, il est possible de distinguer deux tendances :

- approche globale : les caractéristiques sont calculées sur l'image toute entière,
- approche par sous-blocs : l'image est divisée en plusieurs sous-blocs, et les caractéristiques sont extraites de chaque sous-bloc séparément.

Gorkani et Picard [55] sont parmi les premiers à avoir proposé la classification de scènes. Ils ne considéraient alors que deux catégories : paysages de villes et de nature. La classification est faite à partir d'un histogramme des orientations des contours de l'image, en considérant qu'une image avec des orientations verticales prédominantes aura tendance à être une scène urbaine, et les autres plutôt des scènes naturelles. Ce simple critère leur permet d'obtenir 92,9% de bonnes classifications sur une centaine d'images, montrant qu'il est possible de classer des scènes avec des critères globaux.

En 1996, Elaine Yiu [154] a mis en œuvre une classification pour distinguer les scènes d'intérieur des scènes d'extérieur en utilisant des histogrammes de couleur et des orientations de textures. Elle a obtenu une précision de 92% sur une base d'environ 500 images, en combinant par vote majoritaire des k-plus proches voisins et des séparateurs à vaste marge.

Quelques années plus tard, Vailaya et al. [142, 140, 139, 141] proposent une classification des images typiques de vacances toujours par une approche globale, et montrent que des critères de bas niveau permettent de discriminer efficacement plusieurs types de scènes en utilisant une structure hiérarchique (figure 2.8).

Pour apprendre les concepts sémantiques à partir des informations de bas niveau, ces auteurs utilisent un classifieur bayésien binaire, en imposant que l'image doit appartenir à une et une seule des classes qu'ils ont définies. Les images sont d'abord classées entre images d'intérieur et images d'extérieur ; les images d'extérieurs sont divisées entre les images représentant une scène urbaine et les images de paysage. Les paysages sont ensuite répartis dans l'une des trois classes suivantes : coucher de soleil, forêt et montagne, en deux étapes, d'abord en séparant coucher de soleil de forêt-montagne, puis forêt et montagne. Ces classes ont été déterminées en demandant à plusieurs personnes de ranger 171 images en plusieurs classes qu'ils devaient eux-mêmes définir [142]. L'organisation qui en a découlé est représentée sur la figure 2.9.

Différentes caractéristiques sont utilisées à chaque étape suivant la nature de la classification : la distinction intérieur / extérieur utilise des moments de couleur spatiaux ; la distinction ville / paysage est faite avec un vecteur de cohérence des directions des contours ; la séparation coucher de soleil / forêt / montagne utilise des vecteurs de cohérences et des histogrammes de couleurs dans l'espace TSVal.

Sur 6931 photographies, les taux de bonnes reconnaissances obtenus pour chaque classifieur binaire sont de 90,8% pour la classification intérieur / extérieur, 95,3% pour ville / paysage, 94,9% pour coucher de soleil / forêt-montagne et 93,6% pour forêt / montagne. Ils comparent leur classifieur bayésien à un k-plus proche voisin, comme ce qui est utilisé notamment par Szummer et al. [132] pour la classification intérieur / extérieur, et soulignent que le classifieur bayésien permet de gagner beaucoup au niveau

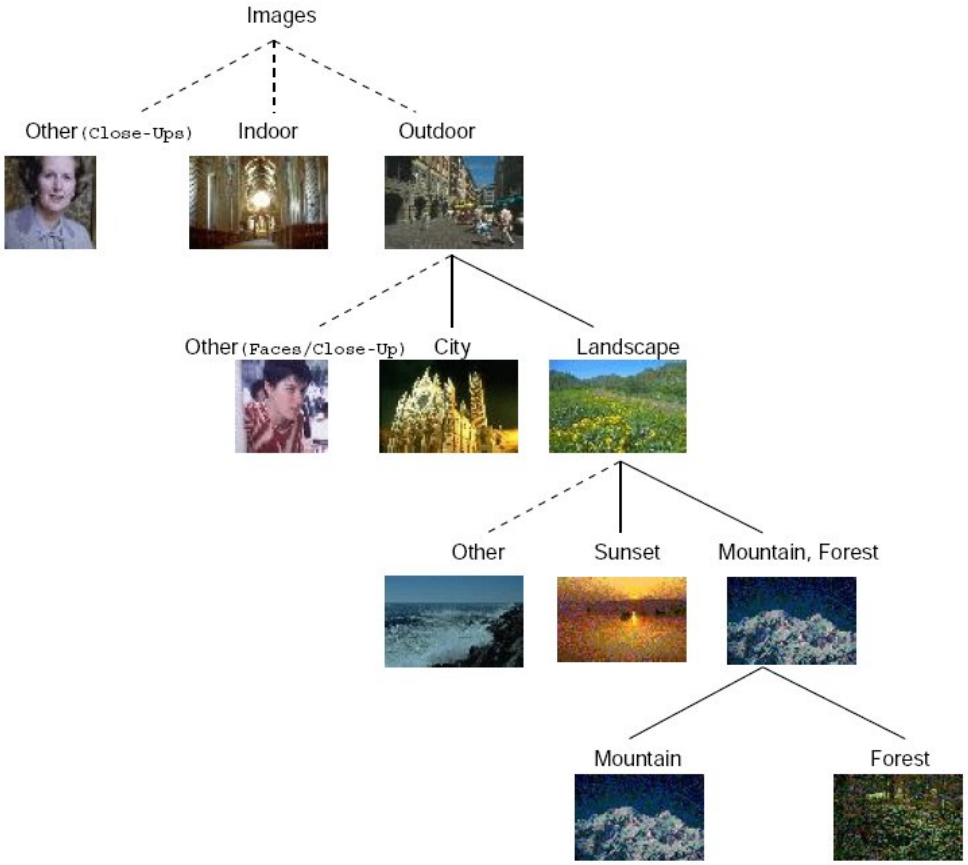


FIG. 2.8 – Classification hiérarchique proposée par Vailaya et al. [139]. Ils se sont intéressés aux classifications représentées par les traits pleins.

du temps de classification, sans perdre de précision, puisqu'ils obtiennent 90,8% de bonne classification pour environ 7000 images, un taux comparable à Szummer et al.. Nous pouvons cependant reprocher à ce type d'approche hiérarchique qu'une erreur lors de la première classification ne pourra pas être rattrapée lors des classifications suivantes. D'autre part, les classes qu'ils ont considérées ne s'excluent pas mutuellement : il est possible par exemple d'observer un coucher de soleil à la montagne.

Chang et al. [22] font également une classification de scènes à partir de caractéristiques globales sur l'image, en assignant un pourcentage de confiance de chaque classe pour chaque image. Parmi 25000 images pour 116 catégories, ils définissent un ensemble d'images d'apprentissages et entraînent un *Bayes Point Machine, BPM* [62] pour chaque classe. Ils montrent qu'ils obtiennent de meilleurs résultats avec un BPM qu'avec un SVM. Les caractéristiques utilisées sont un histogramme de couleurs (12 couleurs sont définies) ainsi que la moyenne, la variance, l'élongation et l'étendue de chaque couleur. Les caractéristiques de textures sont calculées avec des transformées en ondelettes

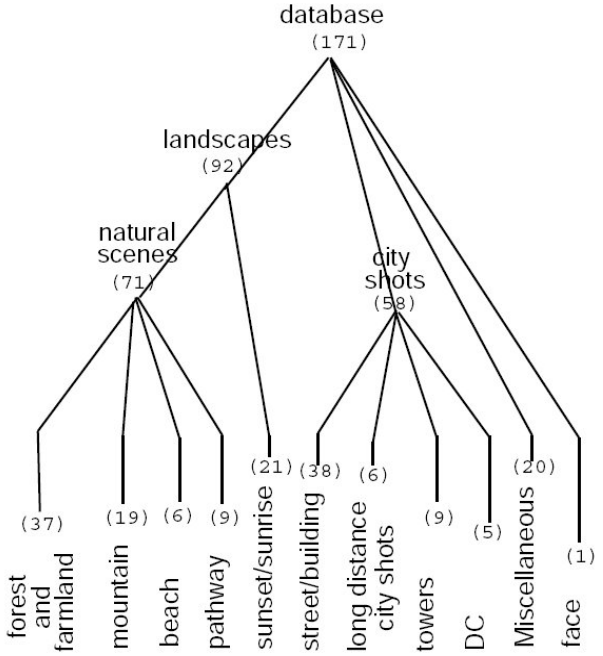


FIG. 2.9 – Classification déterminée spontanément par des personnes auxquelles 171 images de vacance sont présentées [142].

discrètes dans trois directions : horizontale, verticale et diagonale et dans trois résolutions différentes. Pour chacune, ils retiennent la moyenne, la variance, l'élongation et l'étendue de l'énergie. Lors du test, les images sont classées par tous les classifieurs qui lui attribuent chacun un score de confiance, ce qui permet notamment de gérer les cas où les images appartiennent à plusieurs classes en même temps. Des exemples de résultats sont montrés sur la figure 2.10.

En utilisant la moitié des images pour l'apprentissage, ils obtiennent en moyenne 63,1% de bonne classification sur chaque classe.

Remarquons également les travaux de Raimondo Schettini, Carla Brambilla, Claudio Cusano et Gianluigi Ciocca [121] qui classifient cette fois en trois classes : intérieur, extérieur et plan rapproché[1]. Les caractéristiques utilisées sont liées à la couleur, à la texture et aux contours. Pour la couleur, ils utilisent la moyenne , l'écart type et la dissymétrie des moments d'ordre 0, 1 et 2 des canaux de l'espace TSVal.

Pour les contours, ils utilisent divers critères mesurant leur répartition dans l'image. Les caractéristiques de texture sont notamment les énergies et les écarts types d'une décomposition en ondelettes, ainsi que diverses mesures de contraste, complexité, etc. La classification est faite avec une forêt de décision (un ensemble d'arbres de décision). Parmi leur base de 9000 images (4500 pour l'apprentissage, et 4500 pour le test), environ 89% des images sont correctement classées dans les trois classes.

[1]traduction pour l'anglais « close-up »

FIG. 2.10 – Exemples de classifications proposées par Chang et al. [22]. Première ligne : *ville* est la classe reconnue avec la plus grande probabilité, et aucune autre classe n'a une probabilité élevée. Deuxième et troisième lignes : *ville* est la classe reconnue avec la plus grande probabilité, et une seconde classe a une probabilité non négligeable. Cette classe est juste pour les images de la seconde ligne et fausse pour celles de la troisième.

Yavlinsky et al. [153] proposent de résoudre le problème en faisant une estimation de densité non-paramétrique, et en introduisant la distance EMD (*Earth Mover Distance*) dans cette estimation. Les critères de bas niveau utilisés sont relativement simples. Pour chaque pixel, ils calculent les trois valeurs du pixel dans l'espace couleur CIE-Lab, et un sous-ensemble de trois caractéristiques de texture parmi celles proposées par Tamura [133] et adaptées pour le traitement d'images par Howarth et al. [64] : la rugosité, le contraste et l'orientation. Ensuite, pour chacune de ces six valeurs, les moments d'ordre un (la moyenne), deux (l'écart type), trois et quatre sont calculés sur toute l'image, résultant en un vecteur à 24 composantes. Ils comparent leur approche pour l'annotation automatique d'images avec les travaux de Lavrenko et al. [72], Metzler et al. [95] et Feng et al. [45][2] en utilisant la même base de 5000 images de Corel correspondant à un vocabulaire de 371 mots, décomposée en 4500 images d'apprentissage et 500 images pour l'évaluation. Leurs résultats (16% de précision et 19% de rappel) sont comparables à ceux de [72], et légèrement moins bons que [95] et [45]. Nous voyons par ces résultats les limites actuelles des algorithmes d'annotation automatique d'images lorsque l'on chercher à reconnaître quelques centaines de classes.

Détaillons maintenant les approches cherchant à d'abord classifier des sous-blocs d'une image pour en déduire la catégorie de l'image. Les premiers travaux de ce type remontent en 1998 quand Szummer et Picard [132] ont proposé de classifier indépendamment

[2] ces travaux sont décrits plus loin, dans la partie 2.3.2

chaque sous-bloc et d'en déduire la classe de l'image par un vote majoritaire. Ils ont appliqué cette technique à la classification d'images d'intérieur et d'extérieur. L'image est d'abord partitionnée en 16 sous-blocs, et pour chaque bloc sont extraits un histogramme de couleur dans l'espace de Otha, les paramètres d'un modèle auto-régressif en multirésolution (*multiresolution simultaneous autoregressive*, MSAR) [88] qui tient compte de la texture, et les coefficients de la transformée en cosinus discrète. La classification est ensuite effectuée avec un algorithme de k-plus proches voisins, en utilisant une intersection d'histogrammes :

$$dist(X,Y) = \sum_{i=1} N(X(i) - min(X(i), Y(i)))$$

La classe de l'image est déterminée par un vote majoritaire. Ils obtiennent 90,3% de bonnes classifications sur une base d'un peu plus de 1300 images, et montrent que cette méthode est plus efficace que de classifier directement l'image sans la diviser en sous-blocs.

Des résultats similaires ont été obtenus par Paek and Chang [110]. Ils ont de plus expérimenté la combinaison de plusieurs classifieurs probabilistes en utilisant un réseau de croyances : ils ont entraîné des classifieurs pour « intérieur/extérieur », « ciel/pas de ciel », « végétation/pas de végétation » en utilisant un histogramme dans l'espace TSVal et un histogramme de directions des contours. Les résultats de chaque classifieur sont alors utilisés en entrée du réseau de croyances qui retourne en sortie une décision sur la classification « intérieur/extérieur », leur permettant de passer de 83,1% de bonnes détections à 86,3%.

Navid Serrano, Andreas E.Savakis et Jiebo Luo [122, 123] proposent une amélioration de [132], cette fois en utilisant un SVM à deux étapes : une première étape pour la classification de chaque sous-bloc, et une seconde pour la décision finale en fonction des classifications des sous-blocs. Ils utilisent également une transformée en ondelettes pour la texture à la place de MSAR. Leur base d'images est la même que celle utilisée dans [132], à laquelle ils ont enlevé les images qui étaient presque le doublon d'une autre image, ainsi que celles pour lesquelles la classification « intérieur/extérieur » était ambiguë. Sur ce sous-ensemble, la méthode de Szummer et Picard obtient 85% de précision, alors que leur nouvelle méthode atteint 90,2%. Ils ont également essayé d'inclure des niveaux sémantiques intermédiaire dans la classification, comme la détection d'herbe et de ciel, en plus des caractéristiques de bas niveau (couleur et texture) [84, 123], mais dans leur cas, cela ne leur permet de passer que de 89% à 90,7%.

Apprentissage en utilisant un niveau intermédiaire

Certains travaux ont démontré que le contenu de la scène, comme par exemple la présence de personnes, de ciel ou d'herbe, peut être utilisé comme une information supplémentaire pour améliorer les performances des classifieurs entraînés uniquement à partir de caractéristiques de bas niveau tels que ceux que nous avons décrits ci-dessus. Cela permet de réduire le fossé sémantique entre les caractéristiques de bas niveau et la sémantique de la scène, en utilisant ce qui peut être considéré comme un niveau

sémantique intermédiaire. Ces techniques sont regroupées sous le nom de « modélisation sémantique » (*semantic modelling*). Utiliser un niveau sémantique intermédiaire rend le problème de classification de scènes plus difficile à mettre en œuvre que lorsque l'apprentissage se fait directement à partir des caractéristiques de bas niveau. En effet, avoir recours à un niveau sémantique intermédiaire implique souvent une étape de reconnaissance d'objets, et fait donc appel à des techniques d'analyse locale de l'image, notamment par régions ou par points d'intérêt. Toutefois, c'est un moyen d'acquérir des informations supplémentaires par rapport aux caractéristiques de bas niveau, et on peut espérer alors obtenir une meilleure performance pour la classification de scènes.

Une approche classique consiste à décrire les différents types de scènes que l'on cherche à reconnaître en fonction des objets qui doivent apparaître dans la scène. Ces travaux impliquent que l'on soit déjà capable de reconnaître les objets en question, ce qui n'est pas forcément plus simple que de reconnaître la scène elle-même, et s'oppose aux méthodes où la reconnaissance de la scène est utilisée pour guider la reconnaissance d'objets.

Quelques travaux récents explorent une deuxième piste en tentant d'éviter l'étape de la segmentation et de la détection d'objets. Ils se servent pour cela de représentations sémantiques intermédiaires détectées à l'aide de descripteurs locaux. Ils identifient d'abord un vocabulaire de mots visuels ou de concepts sémantiques locaux, et effectuent un apprentissage de ces mots visuels pour chaque catégorie de scène. Les concepts sémantiques locaux définissent la sémantique d'une image à partir d'informations locales. Ces concepts locaux correspondent en général à des objets tels que *ciel bleu*, *ciel gris*, *eau avec des vagues*, *montagne enneigée*, *montagne sans neige*.

Une troisième tendance utilise des qualités globales et locales liées à la structure d'une scène, tels que la rugosité, l'expansivité, etc. et est issue des travaux pionniers de Oliva et Torralba [107]. Nous pouvons donc classer les travaux en trois approches pour la reconnaissance de scènes :

- en faisant d'abord une reconnaissance des objets dans l'image par analyse de régions, notamment en segmentant l'image au préalable pour en déduire une description de la scène ;
- en extrayant des descripteurs locaux autour de points d'intérêt afin de faire de la classification. C'est dans cette catégorie notamment que se situent les méthodes récentes de classification par sacs de mots ;
- en utilisant des propriétés sémantiques de l'image : Y a-t-il des personnes dans l'image ? L'image est-elle plutôt naturelle ou urbaine ? Est-ce un espace dégagé ou fermé ?

Dans la catégorie des approches commençant d'abord par une **reconnaissance d'objets par régions** se trouvent par exemple les travaux de Fan et al. [39] qui s'intéressent à la classification d'images naturelles. Pour cela, ils apprennent et reconnaissent des objets appelés *concept-sensitive salient objects* au niveau de la région à l'aide d'un SVM. Un modèle de mélange fini est alors optimisé pour déduire de ces classifications au niveau de la région des annotations sémantiques au niveau de l'image.

Luo et al. [85] ont proposé une approche hybride : des caractéristiques de bas niveau

– un histogramme de couleurs dans l'espace Otha et une analyse MSAR (*multiresolution simultaneous autoregressive*) de texture comme dans Szummer et al. [132]) – et des reconnaissances d'objets dans la scène – détection de ciel, d'herbe, etc. – sont combinées pour entraîner un réseau bayésien permettant de classifier la scène. Les auteurs proposent trois applications de ce système :

- détecter les sujets principaux dans une image [86],
- trouver l'image la plus pertinente pour un évènement,
- classifier les images entre images d'intérieur et images d'extérieur, comme ils l'avaient fait dans [84].

Les auteurs montrent que les performances obtenues avec l'utilisation conjointe de caractéristiques de bas niveau et de caractéristiques sémantiques (les objets détectés) sont bien meilleures qu'en n'utilisant que les caractéristiques de bas niveau, améliorant notamment le taux de bonnes classifications « intérieur/extérieur », qui passe de 82,3% en utilisant seulement les caractéristiques texture et couleur à 90,1% en rajoutant la composante sémantique.

Aksoy et al. [2] ont également utilisé un réseau bayésien, qu'ils ont associé à une grammaire visuelle. La représentation de scènes est faite en décomposant l'image en régions prototypes, et en modélisant les interactions entre ces régions en termes de relations spatiales. Les images sont d'abord segmentées, puis des groupes de régions significatifs pour la discrimination des scènes présentes dans la base sont créés automatiquement par apprentissage afin de construire un modèle de grammaire visuelle.

Mojsilovic et al. [103] recherchent explicitement des objets dans des images pour en déduire la catégorie sémantique de la scène. Après une première étape de segmentation de l'image suivant des critères de textures et de couleurs, des caractéristiques locales sur la texture, la couleur et la forme des régions sont extraites ainsi que des caractéristiques globales comme par exemple le nombre de régions, la complexité de l'image ou sa symétrie. Ces caractéristiques sont utilisées pour reconnaître certaines régions telles que *peau*, *ciel*, *eau*, *montagne*, *objet artificiel*. Ces objets sont ensuite utilisés pour attribuer une catégorie sémantique à la scène elle-même, dont notamment : *portrait*, *personnes*, *extérieur*, *paysage*, *ville*.

Enfin, Voget et Schiele [145, 146] font une segmentation simple de l'image en utilisant une grille régulière 10x10, puis extraient des caractéristiques couleur et texture pour classifier chaque bloc avec un algorithme de k-plus proches voisins ou un SVM. Une image est alors représentée par un vecteur de co-occurrence (COV) mesurant la fréquence des différents objets dans une image. La moyenne des COV des N_c images d'une catégorie c définit ce que les auteurs nomment le prototype de catégorie P_c :

$$P_c = \frac{1}{N_c} \sum_{j=1}^{N_c} COV(j)$$

Ce prototype permet alors de faire la classification de scène.

Plus récemment, des travaux ont montré que la technique dite de sacs de mots (section 3.4) pouvait donner de très bons résultats pour la classification de scènes [15, 42, 73, 117]. Bosch et al. [15], notamment, calculent des sacs de mots à partir

du descripteur local SIFT [83] et les utilisent pour l'apprentissage de catégories avec l'algorithme pLSA *probabilistic latent semantic analysis*. Les descripteurs locaux sont calculés à partir d'une grille régulière. Ils classifient 13 catégories avec un taux moyen de bonnes reconnaissances de 73,4%.

Quelhas et al. [117] avaient proposé une approche similaire, ne portant que sur 3 catégories, où les descripteurs SIFT étaient calculés autour de points obtenus à partir d'un détecteur de points d'intérêt. Toutefois, il a été démontré [42] que dans le cas de la classification de scènes, et notamment avec des images de la nature, les descripteurs denses (par grille régulière) donnent de meilleurs résultats que les descripteurs diffus (par points d'intérêt).

Fei-fei et Perona [42] utilisent également des sacs de mots avec le descripteur local SIFT et effectuent l'apprentissage en proposant une variante de l'algorithme LDA (*Latent Dirichlet Allocation*) [12] pour classifier les scènes en 13 catégories : chambre, cuisine, salon, bureau, autoroute, intérieurs de villes, grand bâtiments, rues, banlieue, forêt, côte, montagne et paysages ouverts. Un schéma de leur approche est montré sur la figure 2.11. Les régions sont d'abord classées localement en différents thèmes intermédiaires (les concepts sémantiques locaux) puis en catégories. L'apprentissage des catégories en fonction des concepts sémantiques locaux est automatique, le seul travail manuel à effectuer au préalable étant l'attribution d'une catégorie aux images de la base d'apprentissage.

Parmi les travaux utilisant des attributs sémantiques pour la classification de scènes, remarquons notamment l'approche de Oliva et Torralba [107, 108, 137] qui utilisent plusieurs propriétés perceptuelles qu'ils définissent : le naturel par opposition à ce qui est fait par l'homme, l'ouverture définie par la présence d'une ligne d'horizon, la rugosité calculée comme correspondant à la complexité fractale de l'image, l'expansion correspondant à la profondeur de perspective de la scène, et la rudesse qui représente la déviation des lignes d'une scène par rapport à sa ligne d'horizon : une image avec une rudesse élevée a des bords obliques et une ligne d'horizon masquée. Chacune de ces propriétés correspond ensuite à une dimension de ce qu'ils appellent « l'enveloppe spatiale » d'une image. Ils estiment ces propriétés à l'aide d'une représentation spectrale de l'image.

Néanmoins, les travaux sur la classification de scènes partent de l'hypothèse que les scènes sont clairement séparables, ce qui n'est pas le cas dans toutes les applications considérées dans cette section : une image peut représenter à la fois une scène de forêt et de montagne, une scène de plage et de coucher de soleil, et ville et nature ne s'opposent pas forcément, comme illustré sur la figure 2.12.

L'approche de Chang et al. [22] décrite précédemment est la seule parmi celles présentées dans cet état de l'art à proposer de permettre à une image d'appartenir éventuellement à deux classes différentes. L'autre possibilité est, comme nous le verrons dans la suite, d'annoter une image par rapport à ses régions et les objets qu'elle contient plutôt que de l'annoter par rapport à la scène qu'elle représente.

2.3.2 Approches par régions

Les travaux présentés jusqu'ici dans cette section de l'état de l'art sur l'annotation automatique d'images ont été qualifiés d'« approches orientées scène ». Nous entendons

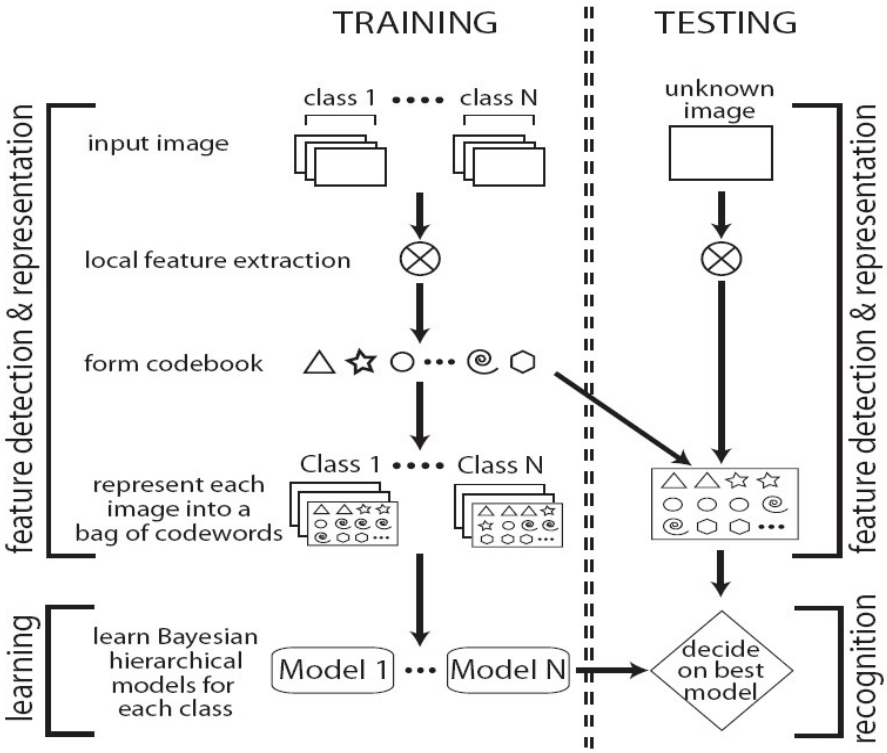


FIG. 2.11 – Schéma de l'approche proposée par Fei-fei et Perona [42] pour la classification de scènes.

par là que ces approches cherchaient à attribuer un ou plusieurs mots-clés à la scène, c'est-à-dire à l'image considérée comme un tout. Au contraire, les travaux présentés dans ce qui suit ont pour but d'attribuer des mots-clés aux régions de l'image et seront donc appelés « approches par régions ». La distinction entre ces deux types d'approches est dans certains cas difficile à faire car comme nous l'avons vu, certaines « approches orientées scène » se fondent sur une reconnaissance des objets de la scène et pourraient donc tout aussi bien être classées comme « approches par régions ». De même une classification de la scène représentée par l'image peut servir d'opération préalable à l'annotation au niveau des régions. Nous sommes donc conscient que la séparation que nous avons faite, suivant ce qui est au final classé (les régions ou la scène), est discutable et n'est pas dans la réalité une séparation stricte.

Classification des régions à partir de régions annotées

Ces approches consistent à annoter les régions d'une image à partir d'une base d'apprentissage constituée de régions qui ont été manuellement sélectionnées dans des images et étiquetées. La stratégie pour l'annotation d'une nouvelle image est alors fondamen-

FIG. 2.12 – Exemple d'images appartenant à plusieurs catégories de scène. De gauche à droite : montagne/forêt, plage/coucher de soleil, ville/nature.

talement la suivante : l'image est segmentée en plusieurs régions, puis des descripteurs sont extraits de chaque région et comparés aux descripteurs extraits des régions de la base d'apprentissage pour classifier la région. À l'étape de segmentation près, cela est donc très proche de l'apprentissage directement à partir de critères de bas niveau des approches orientées scène présentées précédemment (section 2.3.1).

Citons par exemple les travaux d'Aghbari et al. [1] qui effectuent une classification des régions d'images d'extérieurs avec des séparateurs à vaste marge. Les 13 fonds suivants sont détectés : objets artificiels, nuit, forêt verte, forêt rouge, ciel bleu, océan bleu, sable, caillou, coucher de soleil dans le ciel, coucher de soleil sur l'eau, nuage, neige, chute d'eau. Ils sont représentés et regroupés sous forme d'un arbre hiérarchique afin d'optimiser la vitesse de classification. L'image est d'abord segmentée avec l'algorithme de segmentation hill-climbing. Ensuite, pour chaque région, les descripteurs calculés pour la classification sont un histogramme de couleurs dans l'espace TSVal, un histogramme de directions des contours, une auto-corrélation des vecteurs de contours, et une transformée en ondelettes en arbre dual complexe (*dual-tree complex wavelet transform*).

Classification des régions à partir d'images annotées globalement

La classification des régions à partir d'images annotées globalement concerne la possibilité d'attribuer des mots aux régions d'une image, en n'ayant comme connaissance qu'une annotation au niveau de l'image : l'image est annotée avec certains mots-clés, mais on ne sait pas où se situent les objets correspondant à ces mots dans l'image. Cette approche est jugée plus intéressante que la précédente par beaucoup de travaux dans le sens où elle permet de s'affranchir de la constitution manuelle d'une base de régions étiquetées, et notamment de l'étape nécessaire de la segmentation manuelle qui est assez fastidieuse.

Une des premières tentatives pour résoudre ce problème a été proposée par Mori et al. en 1999 [104]. Ils séparent les images en plusieurs régions à l'aide d'une grille rectangulaires et extraient un histogramme RVB à 64 composantes ainsi que des histogrammes de directions des contours de Sobel. Une technique de regroupement est appliquée à ces primitives de bas niveau d'une part et les mots d'une image sont attribués à toutes ses régions d'autre part. Un modèle de co-occurrence des mots m_i et des groupes c_j obtenus

est alors entraîné en estimant la probabilité conditionnelle $P(m_i|c_j)$ afin d'associer les mots aux groupes.

Une autre étude qui a servi de base à de nombreux autres travaux fut celle de Duygulu et al. [35, 36]. Ils ont créé un vocabulaire discret de classes formées par regroupement (*clustering*) de régions extraites d'une base de données d'images. Ces images sont par ailleurs accompagnées d'une annotation textuelle au niveau global. Ils proposent d'appliquer le modèle de *machine translation* inspiré du domaine de la traduction automatique pour déduire automatiquement la correspondance entre ce vocabulaire de régions et les mots clés. Cela permet de passer de manière automatique d'une annotation au niveau global pour l'image à une annotation au niveau de ses régions. Ils ont poursuivi ces études et les ont comparées avec d'autres modèles dans Barnard et al. [6].

Le principe du modèle de *machine translation* est d'introduire une variable cachée l pour faire de l'apprentissage non supervisé. Soit F l'espace des caractéristiques visuelles (*features*) et M l'ensemble des mots ; ce modèle s'écrit sous la forme :

$$P_{F,M}(f,m) = \sum_{l \in L} P_{F,M|L}(f,m|l)P_L(l)$$

Pour un état donné, les probabilités des caractéristiques visuelles et des mots sont considérées indépendantes :

$$P_{F,M|L}(f,m|l) = P_{F|L}(f|l)P_{M|L}(m|l)$$

permettant de réécrire $P_{F,M}(f,m)$ ainsi :

$$P_{F,M}(f,m) = \sum_{l \in L} P_{F|L}(f|l)P_{M|L}(m|l)P_L(l)$$

Les probabilités $P_{F|L}(f|l)$ et $P_{M|L}(m|l)$ sont ensuite estimées par l'algorithme EM (Espérance maximisation [3]). L'annotation d'une nouvelle image est obtenue en appliquant la formule de Bayes :

$$P_{M|F}(m|f) = \frac{P_{F,M}(f,m)}{P_f(f)}$$

Jeon et al. [67] ont reformulé l'annotation d'images et la recherche d'images par similarité en un problème de recherche d'information crosslingue : les régions sont regroupées pour définir un vocabulaire réduit de même que dans [35], mais un modèle de pertinence cross-media (*cross-media relevance model, CMRM*) est utilisé au lieu de l'algorithme EM pour estimer la probabilité conjointe de la présence d'un mot et d'une région dans une image donnée. Cela peut-être utilisé de deux façons : dans le premier cas, les régions sont utilisées pour générer des mots avec une certaine probabilité, ce qui permet d'annoter les images. Le deuxième cas consiste à utiliser au contraire les mots-clés posés en question pour sélectionner certaines régions issues du vocabulaire visuel et leur associer une certaine probabilité afin de pouvoir ensuite comparer ces régions avec celles contenues dans

[3] En anglais : *Expectation-maximisation algorithm*

chaque image en utilisant la distance de Kullback-Liebler. Jeon et al. ont obtenu des résultats avec une précision moyenne deux fois meilleure que celle obtenue par Duygulu et al. [35] en recherche d'images.

Lavrenko et al. [72] ont adapté le modèle de Jeon et al. [67] pour utiliser une fonction de densité de probabilité continue (*Continuous Relevance Model, CRM*) afin de décrire le processus de génération des caractéristiques à partir des régions, espérant éviter la perte d'information due à la quantification. Ils fixent également la taille de l'espace des variables latentes en imposant que chaque image de la base d'apprentissage soit un état de cette variable. Ils ont amélioré sensiblement les résultats par comparaison avec le modèle CMRM sur le même ensemble d'images.

Metzler et Manmatha [95] utilisent les annotations de l'ensemble d'apprentissage, et les connectent pour construire un réseau d'inférence. Une nouvelle image est alors annotée en propageant l'information extraite de ses régions dans ce réseau pour obtenir des mots. Ce réseau d'inférence permet également de développer un langage de requêtes évolué rendant possible par exemple de combiner image et texte en question du système de recherche d'images. Les résultats sont comparés au modèle CRM utilisé dans [72], montrant une amélioration à la fois du rappel et de la précision.

Feng et al. [45] remplacent les régions irrégulières par des blocs rectangulaires et modélisent l'annotation d'image par un modèle de distribution de Bernoulli multiple (*multiple Bernoulli distribution*), ce qui leur donne de meilleurs résultats que Lavrenko et al. [72] et Metzler et al. [95].

Les méthodes décrites ci-dessus utilisent des techniques de regroupement (*clustering*) et dépendent donc de la qualité de ce regroupement pour l'apprentissage de régions. Yang et al. [151] proposent d'utiliser le *Multiple-Instance Learning* et notamment l'algorithme *point-wise diverse density, PWDD* [91] comme variante. *Multiple-Instance Learning* est un apprentissage supervisé où les données d'apprentissage sont divisées en plusieurs ensembles d'instances. Un ensemble d'instances est positif pour l'apprentissage si au moins une instance de cet ensemble est positive. Il est négatif si tous les éléments qu'il contient sont négatifs. Cela correspond au problème d'apprentissage d'objets dans les images lorsqu'on ne dispose que d'annotations au niveau image. Les instances sont alors les régions, et les ensembles d'instances sont les images, positives pour l'apprentissage si elle contiennent l'objet, et négative dans le cas contraire. L'algorithme *PWDD* permet d'apprendre les meilleures descripteurs ainsi que les meilleures régions permettant de discriminer l'objet. Ces régions sont ensuite utilisées pour entraîner un classifieur bayésien permettant l'annotation de nouvelles images. Sur la base de 5000 images utilisée par Duygulu et al. [35], Yang et al. [151] obtiennent 31% de précision et 46% de rappel pour l'annotation des 49 mots clés les plus fréquents, contre 20% de précision et 34% de rappel pour [35].

Dans les apprentissages non supervisés considérés ci-dessus, les probabilités des mots et des caractéristiques visuelles étant donné une variable cachée sont considérées comme indépendantes même si ce n'est pas le cas dans la réalité. Carneiro et al. [20] souhaitent définir explicitement la dépendance entre les mots et les caractéristiques en cherchant à

estimer la distribution $P_{F|M}(f|m)$ qu'ils utilisent pour calculer $P_{M|F}(m|f)$:

$$P_{M|F}(m|f) = \frac{P_{F|M}(f|m)P_M(m)}{P_F(f)}$$

Ce modèle permet également de se libérer de l'estimation de la distribution de la classe négative, qui est nécessaire dans un apprentissage multiclasse de type « un contre tous ». Les caractéristiques qu'ils extraient des images sont des régions de 8×8 pixels sur toute l'image, puis pour chaque région, ils obtiennent un vecteur à 192 composantes en calculant la transformée en cosinus discrète sur chaque canal de l'espace YBR. Ils utilisent ensuite un algorithme de *Multiple-Instance Learning* pour l'apprentissage. Pour une précision égale, ils obtiennent 16% de mieux en rappel par rapport au meilleur parmi [35], [45], [72] et [104] sur les 5000 images extraites de la base Corel. Pour la catégorisation d'images, ils améliorent les résultats de 10% (passant de 26% à 36%) par rapport à [74].

Fergus et al. [48] ainsi que Dorko et al. [32] classifient les objets à partir de descripteurs calculés sur des régions entourant des points d'intérêt. Leur principal apport est la détermination automatique de la taille de la région à utiliser pour chaque point en fonction d'une mesure de l'entropie afin que le descripteur soit invariant par changement d'échelle.

Un apprentissage en deux phases est expérimenté par Li et al. [76] : d'abord une phase générative, puis une phase discriminante. Les descripteurs utilisés sont de trois sortes : les valeurs L,a,b de chaque région issue d'une quantification des couleurs dans l'espace CIELab ; les coefficients de textures de Gabor calculés sur les régions issues de la segmentation par couleur précédente ; des descripteurs de structure, rendant compte du nombre de lignes dans une région, de leurs orientations, de leurs intersections et des couleurs de part et d'autre de chaque ligne. Étant donné que le nombre de régions par image est variable, un algorithme EM est utilisé dans la phase générative pour modéliser les données sous forme d'un modèle de mélange gaussien utilisé ensuite pour produire un descripteur de taille fixe. Ensuite, la phase discriminative consiste à entraîner un algorithme d'apprentissage sur ces descripteurs, et en particulier dans l'article, un perceptron à trois couches. Les résultats montrent une amélioration par rapport à Duygulu et al. [35], ALIP [74], Fergus et al. [48] et Dorko et al. [32].

Une autre approche intéressante est celle de Blei et Jordan [11] proposant une extension au modèle d'allocation de Dirichlet latent (*Latent Dirichlet Allocation, LDA*) [12] qui fait l'hypothèse qu'un mélange de facteurs latents *mixture of latent factors* est utilisé pour générer à la fois les mots et les descripteurs issus des régions. Les mots et descripteurs sont donc supposés indépendants entre eux, mais dépendants des mêmes facteurs latents. Les auteurs montrent alors comment utiliser ce modèle pour assigner des mots à chaque région. L'algorithme EM est là encore utilisé.

D'autres travaux de recherche se rapprochent de l'annotation automatique, notamment ceux de Lim et al. [80], où ils proposent de faire de la découverte de régions sémantiques récurrentes dans les images. En partant d'une base d'images classifiée, les régions qui sont communes à plusieurs images d'une classe, et celles qui permettent de

différencier les classes sont découvertes et apprises afin de pouvoir classifier de nouvelles images. Il n'y a pas d'utilisation d'une segmentation : les régions apprises sont rectangulaires mais différentes échelles sont prises en compte.

2.3.3 Utilisation de la sémantique des objets

Nous donnons dans cette partie quelques travaux faisant explicitement usage de la sémantique des objets afin de raffiner une annotation automatique qui aurait été obtenue par l'une des méthodes décrites précédemment ou de produire une annotation sémantiquement cohérente d'une image. Certaines des méthodes précédemment présentées utilisaient déjà la sémantique des objets. Nous décrivons ici les travaux qui nous semblent importants et qui sont plutôt axés sur l'inclusion de la sémantique dans le processus d'annotation alors que les autres travaux étaient axés vers la manière d'annoter, directement ou indirectement, à partir des critères de bas niveau.

Quattoni et al. [116] proposent d'utiliser des relations sémantiques entre objets afin d'améliorer les résultats en annotation automatique, et notamment la connaissance de quels types d'objets sont présents dans quels types de scènes. Les types de scènes sont par exemple *forêt*, *rue*, *bureau*, *chambre*, et les objets sont *arbre*, *voiture*, *soleil*, *lit*. Ce qu'ils proposent, c'est d'annoter chaque région ainsi que la scène dans sa globalité en même temps. Dans cette étude, les relations sémantiques sont supposées fixées et ne peuvent pas être remises en cause par l'algorithme. Ensuite, l'algorithme calcule une probabilité pour chaque région et pour la scène en utilisant un Champ Conditionnel Aléatoire (Conditional Random Field, CRF) [115]. Une fonction appelée *compatibilité* doit alors être maximisée pour obtenir le meilleur ensemble régions/scène tout en restant cohérent avec les relations sémantiques définies. Ils ont montré que les résultats sont meilleurs lorsque les relations sémantiques sont prises en compte, le taux d'erreurs passant de 15% à 9%.

Un travail proche est celui de Carbonetto et al. [19] où ils proposent d'apprendre les co-occurrences entre les différents objets en analysant, dans une base d'images annotées, les couples de mots-clés qui apparaissent le plus souvent ensemble dans la description d'une même image. Ils modélisent alors les données par un champ aléatoire de Markov qui prend en compte les régions voisines de la région à classifier. Dans cette approche, la présence d'un avion par exemple peut être utilisée pour faire la différence entre l'eau et le ciel.

D'autres travaux utilisent la base de données lexicales WordNet [97] pour déduire les similarités entre concepts. Datta et al. [27] notamment utilisent la mesure de congruence définie par Leacock et Chowdrow [44] afin d'éliminer les mots-clés qui ne seraient pas pertinents par rapport aux autres, parmi une liste de mots générée par annotation automatique. Cette mesure utilise les relations d'hypernymie « est un » en cherchant le nombre minimum de nœuds intermédiaires pour aller d'un concept à un autre. La similarité S entre deux concepts c_1 et c_2 est calculée de la manière suivante :

$$S(c_1, c_2) = -log(PlusCourtChemin(c_1, c_2)/(2 * P))$$

où P est la plus grande profondeur de l'arbre considéré.

Jin et al. [68] présentent un état de l'art des mesures de similarité entre concepts sémantiques. Pour éliminer des mots-clés contradictoires issus d'un modèle de *machine translation* tel que celui de Duygulu [35], Jin et al. combinent trois de ces mesures grâce à la théorie de Dempster-Shafer afin de pallier les inconvénients de chacune. Les résultats en précision sont de 20,0% pour le modèle de *machine translation* seul, 22,3% au mieux quand on utilise une seule mesure, et 30,2% quand les trois mesures sont combinées.

2.3.4 Utilisation d'ontologies visuelles

Une bonne définition de ce qu'est une ontologie est donnée par Billen et al. [10] : « *En intelligence artificielle et en informatique, le terme ontologie réfère à un vocabulaire ou un système de classification qui décrit les concepts opérant dans un domaine particulier à travers des définitions suffisamment détaillées pour saisir la sémantique du domaine. À la notion d'ontologie on associe également les algorithmes permettant la traduction des concepts dans les modèles de données ainsi que les modes de représentation de ces concepts.* »

Le terme ontologie en informatique n'a pas encore de sens bien défini : certains chercheurs utilisent ce terme dans des emplois que d'autres considèrent comme abusifs. En général, ce que les chercheurs dans le domaine de la reconnaissance d'objets désignent comme ontologie, ce sont des hiérarchies de concepts de niveau intermédiaire qui doivent permettre à un humain de donner une description d'un objet en le décomposant suivant ces concepts. Nous designons pas la suite par ontologie ce que les auteurs eux-mêmes ont appellé ontologie.

Ce niveau intermédiaire pourra décrire par exemple les objets en termes de couleur, de position, de taille et de forme. Ainsi, l'objet *ciel* peut être défini comme une région de couleur *bleu ciel*, de texture uniforme, et située en haut de l'image. L'idée sous-jacente est que d'une part, il est plus facile de faire un lien entre les descripteurs du niveau intermédiaire et les descripteurs de bas niveau que d'apprendre directement à reconnaître les objets de haut niveau sémantique à partir de leurs caractéristiques visuelles de bas niveau. D'autre part, il est plus facile pour l'utilisateur de manipuler le niveau intermédiaire que de manipuler les descripteurs de bas niveau.

Mezaris et al. [96] ont construit une telle ontologie (fig. 2.13) pour retrouver des objets à partir de requêtes par mots clés, tels que « aigle », « voiture rouge » ou « tigre ». Chaque objet est décrit par l'utilisateur suivant sa couleur, sa position, sa taille et sa forme. La couleur est définie par trois attributs : la luminance, le rapport de vert par rapport au bleu, et le taux de jaune par rapport au bleu. Ces trois informations sont obtenues grâce à une quantification de l'espace couleur TSL. La position est définie par la position sur l'axe horizontal et celle sur l'axe vertical par rapport au centre de l'image, chaque axe étant quantifié en trois valeurs. La taille peut être petite, moyenne ou grande, en quantifiant le nombre de pixels de la région. Les attributs de forme décrivent l'allongement de l'objet, et sont obtenus à partir de l'excentricité de la région.

Dans sa thèse [87], N. Maillot défend l'idée qu'une catégorisation d'objets peut être découpée en trois phases indépendantes, chacune des phases ayant une ou plusieurs ontologies propres. Cette indépendance a pour conséquence la possibilité de réutiliser

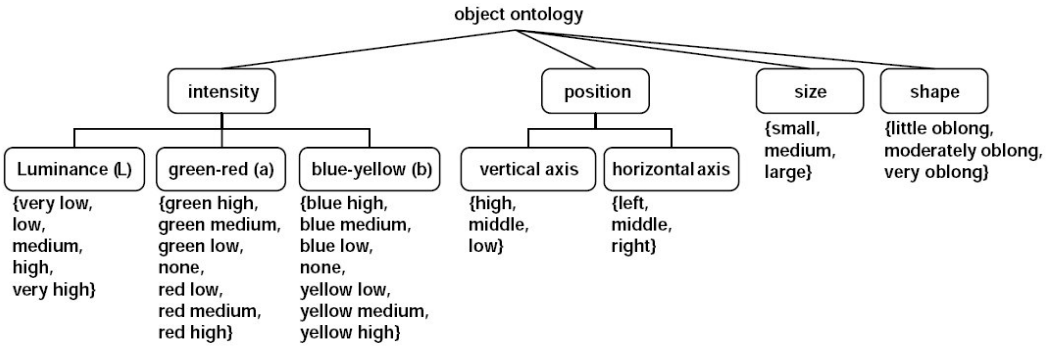


FIG. 2.13 – Ontologie proposée par Mezaris et al. [96].

facilement le système pour une autre application : il est possible de changer l'une des phases sans affecter les deux autres.

Les trois phases sont :

- la phase d'acquisition des connaissances qui utilise une ontologie de domaine (taxonomie et partonomie) et qui acquiert une représentation visuelle de ces classes en faisant appel à plusieurs ontologies de concepts visuels, notamment relations spatiales, couleurs, textures (figure 2.14) et géométriques (figure 2.15),
- la phase d'apprentissage qui apprend à détecter les concepts visuels dans les images. Cette phase permet donc de faire une association visuelle entre les pixels des images (le bas niveau) et les ontologies de concepts visuels à l'aide d'algorithmes d'apprentissage,
- la phase de catégorisation ou classification qui détecte des concepts visuels dans les images afin de les comparer avec l'ontologie de domaine, pour en déduire des hypothèses sur les objets présents dans les images.

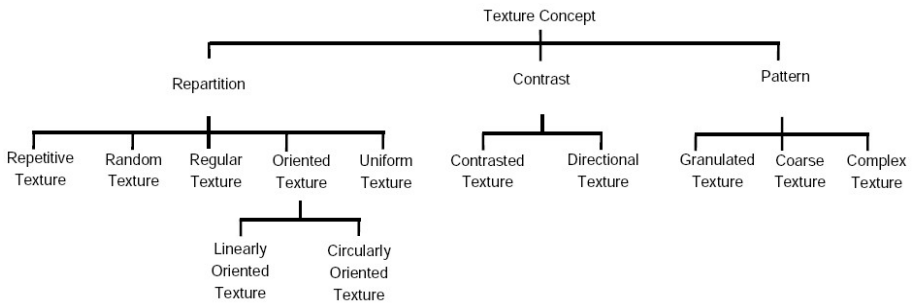


FIG. 2.14 – Ontologie de textures proposée par N. Maillot [87].

Cette structure peut être appliquée par exemple en palynologie, et pour la reconnaissance de véhicules. Elle a été aussi utilisée notamment par C. Hudelot [66] dans sa thèse pour faire du diagnostic précoce de pathologies végétales automatiquement par traitement d'images.

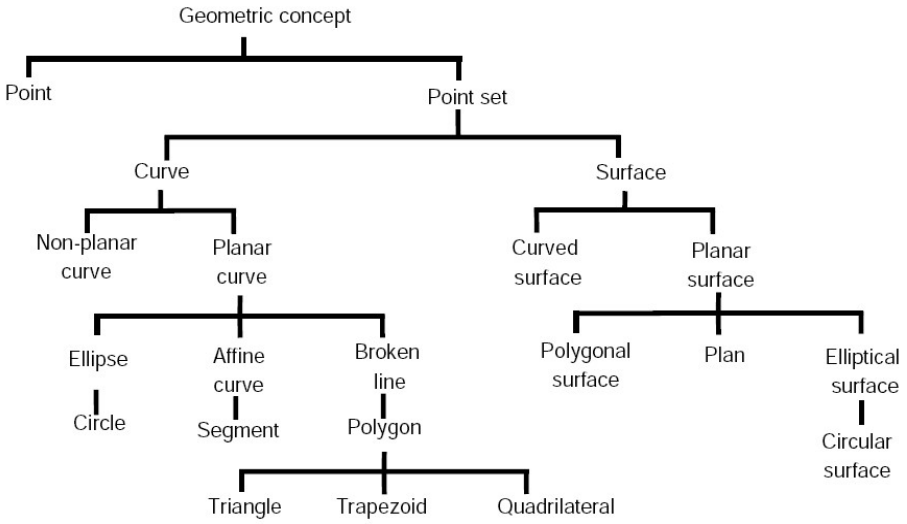


FIG. 2.15 – Ontologie de concepts géométriques proposée par N. Maillot [87].

2.3.5 Apprentissage de concepts automatiquement en utilisant Internet

Les bases de données d'images annotées manuellement et accessibles librement pour faire des travaux de recherche sont encore peu nombreuses et contiennent typiquement de l'ordre de 100 000 images, comme par exemple la base Corel [149] qui contient 60 000 images avec une courte phrase décrivant chaque image, ou la base IAPR TC-12 [58] de 20 000 images contenant des descriptions plus longues traduites en anglais et en allemand.

Il est donc tentant de se tourner vers les images disponibles sur Internet dont le nombre croît toujours plus vite. Les moteurs de recherche d'images tels que Google, Yahoo ! ou Picsearch ont recensé quelques milliards d'images[4]. D'autre part, même si ces images ne sont pas annotées directement, elles sont souvent placées dans une page Internet où elles sont entourées de texte relatif à l'image, ce qui rend possible la recherche d'images par requête textuelle sur Internet telle que nous la voyons actuellement dans la plupart des moteurs de recherche d'images.

L'idée d'utiliser ces images pour apprendre automatiquement des concepts date de 2004 avec les travaux de Fergus et al. [49, 47]. Le concept qu'ils souhaitent apprendre à reconnaître est posé en question à un moteur de recherche d'images par mot-clé sur Internet tel que Google Image Search, et les images retournées sont éventuellement filtrées puis utilisées comme base d'apprentissage. L'une des difficultés majeures est que le pourcentage d'images ne correspondant pas à la catégorie désirée retournées par un tel moteur est important, dépassant souvent 50% [49]. De plus, même en ne considérant que les

[4]Google a annoncé 2,1 milliards d'images le 9 août 2005, Yahoo ! : 1,67 milliard d'images le 10 août 2005, Picsearch : 1,7 milliards en novembre 2006

images pertinentes par rapport à la requête, la diversité de qualité les rend moins idéales en pratique pour l'apprentissage que pour les bases d'images habituelles construites et annotées manuellement. Notamment, le nombre d'objets dans chaque image est inconnu et variable, et la prise de vue et l'échelle sont très variables. Cependant, si l'on arrive à apprendre à partir d'un ensemble aussi bruité, cela signifierait qu'on serait capable d'apprendre n'importe quelle catégorie visuelle.

Fergus et al. ont d'abord proposé [49] d'apprendre un modèle de l'objet en utilisant le détecteur de points d'intérêt de Kadir et Brady [69], ainsi qu'un détecteur de segments courbes. Ce modèle est ensuite utilisé pour réordonner les images collectées sur Internet. Ils ont démontré ainsi qu'on peut obtenir une amélioration de la précision d'environ 15% par rapport au résultat brut renvoyé par Google Image Search, pour un rappel fixé à 15%.

Ils ont poursuivi ces travaux [47] pour les appliquer à la classification d'images. La méthode choisie est d'appliquer l'algorithme d'analyse sémantique latente (pLSA) à ce problème, en estimant les probabilités du modèle avec l'algorithme EM. Ils motivent le choix de cet algorithme en expliquant que ce qu'ils font revient à extraire des composantes cohérentes depuis un grand corpus de données d'une manière non supervisée et que le pLSA est justement utilisé dans ce cadre en analyse textuelle et donne de bons résultats. Ils utilisent un descripteur SIFT calculé sur des régions circulaires trouvées dans l'image à l'aide de différents détecteurs. Dans leur cas, l'algorithme pLSA fonctionne de la manière suivante : soit D un ensemble de documents (ici, des images), contenant chacun un certain nombre de régions représentées par une quantification en M dimensions (appelés « mots visuels », et qui sont effectivement des mots dans le cas du LSA appliqué à l'analyse de textes). La base d'images est représentée par une matrice de co-occurrence de dimension $D \times M$. Pour un mot m et un document d, on introduit la variable latente z ainsi :

$$P(m,d) = \sum_{z=1}^{Z} P(m|z)P(z|d)P(d)$$

Les densités $P(m|z)$ et $P(z|d)$ sont ensuite apprises avec l'algorithme d'optimisation EM afin de maximiser la log vraisemblance L suivant :

$$L = \prod_{d=1}^{D} \prod_{m=1}^{M} P(m,d)^{n(m,d)}$$

où $n(m,d)$ est le nombre de mots m dans le document d. La formule de Bayes permet ensuite d'en déduire simplement $P(m|d)$ afin de classer une nouvelle image.

Dans leur article [47], Fergus et al. étendent ce modèle afin d'inclure l'information sur la position spatiale d'une région dans l'image. L'apprentissage de 7 classes de la base Caltech-101 en utilisant des images provenant du web pour l'apprentissage et des images de la base Caltech-101 pour le test donne des taux d'erreurs de classification en moyenne entre 15% et 20%.

2.3.6 Vocabulaire

La plupart des systèmes existants essaient de reconnaître une dizaine d'objets (entre 10 et 20). Fergus et al. [47] (2005) par exemple font des tests sur 7 catégories.

L'une des bases les plus utilisées actuellement en annotation automatique d'images est la base Corel [5]. Cette base contient 60000 images, regroupées en 600 classes de 100 images chacune, mais en général seul un sous-ensemble de cette base est considéré.

Duygulu et al. [35] ont sélectionné une sous-base de 5000 images extraites de la base Corel, correspondant à un total de 371 mots annotés au niveau image. Cette base est divisée en 4500 images pour l'apprentissage et 500 images pour l'évaluation. Une comparaison des différents travaux évalués sur cette base et présentés précédemment est donnée dans le tableau 2.1

Auteur	Mots avec rappel > 0	Précision	Rappel
Duygulu et al. [35]	80	?	?
Yavlinsky et al. [153]	104	16%	19%
Lavrenko et al. [72]	107	16%	19%
Metzler et al. [95]	112	17%	24%
Feng et al. [45]	122	24%	25%

TAB. 2.1 – Comparaison des résultats d'annotation automatique sur la base de 5000 images extraites de la base Corel.

Yang et al. [151] font remarquer que les mots clés les plus fréquents dans la base d'apprentissage sont mieux appris.

Li et Wang [74] ont testé la catégorisation d'images en utilisant la base Corel complète de 60 000 images. Le but est, pour une image donnée, de retrouver la catégorie de cette image parmi 600 catégories. Ils ont utilisé 40 images par catégorie pour l'apprentissage, et 500 images au total pour l'évaluation. La bonne catégorie est prédite dans 11,9% des cas.

Henning Müller [105] a critiqué l'utilisation de la base Corel, en remarquant que la plupart des travaux faisant référence à cette base en utilisent des sous-ensembles différents. Il a également montré que, même sur un ensemble fixé, il est facile d'obtenir différentes performances, notamment en choisissant de bonnes images pour l'évaluation (ce qui revient à enlever celles qui donnent de mauvais résultats) ou en réduisant le nombre de catégories pour l'apprentissage. L'amélioration ainsi obtenue artificiellement est de 25% à 72% pour la précision sur 20 images, de 17% à 50% pour la précision sur 50 et de 13% à 42% pour le rappel sur 100 images, en gardant le même algorithme, mais en changeant la méthode d'évaluation. Il conseille donc de définir précisément un standard pour les évaluations sur une base donnée, et notamment les images à utiliser pour l'apprentissage et le test.

Récemment, en 2004, Fei-Fei et al. [41] ont constitué la base de données Caltech-

[5] http ://wang.ist.psu.edu/docs/home.shtml

101[6] à partir d'images collectées depuis Internet, comprenant 101 classes d'objets et une classe de fonds, avec de 31 à 800 images par classe, dont la taille est variée mais est pour la plupart des images de l'ordre de 300x300. Cette base a été utilisée depuis pour évaluer plusieurs travaux [57, 73]. Les meilleurs publications affichent un taux de bonnes reconnaissances d'environ 66% pour un apprentissage avec 30 objets par classe (cf. figure 2.16).

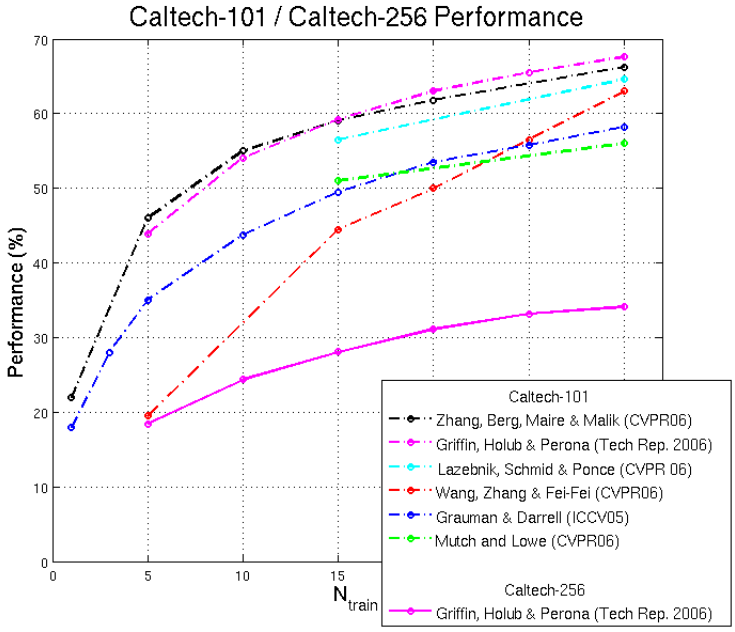


FIG. 2.16 – Pourcentage de bonnes classifications de différentes publications sur les bases Caltech-101 et Caltech-256 en fonction du nombre d'images par classe utilisées pour l'apprentissage. Source : http ://www.vision.caltech.edu/Image_Datasets/Caltech256/

Lazebnik et al. [73] utilisent, pour leurs meilleurs résultats, des descripteurs SIFT dans des régions de taille 16x16 dont les centres sont répartis sur une grille régulière espacée de 8 pixels. Les sacs de mots sont calculés avec un regroupement de ces descripteurs en 200 classes. L'image est considérée en 3 échelles différentes en développant ce qu'ils appellent une pyramide : en entier, divisée en 4 puis divisée en 16, et des histogrammes issus de la classification suivant les sacs de mots sont calculés pour chaque sous-image de chaque échelle. L'apprentissage est fait avec des séparateurs à vaste marge, avec la méthodologie un-contre-tous. La plupart des autres travaux exploitant la base Caltech-101 utilisent des techniques similaires, i.e. de l'apprentissage à partir de descripteurs s'appuyant sur des points d'intérêt.

Une nouvelle base de 256 objets, Caltech-256 [57], créée en 2006 de la même manière que Caltech-101, est la base actuellement étudiée contenant le plus d'objets. Le nombre minimal d'images par classes est de 80. Actuellement, seuls les travaux de Lazebnik et al.

[6] http ://www.vision.caltech.edu/Image_Datasets/Caltech101/Caltech101.html

[73] décrits ci-dessus ont été évalués sur cette base par Griffin et al. [57]. Leurs résultats de classification sont environ de 35% de bonnes reconnaissances en utilisant toujours 30 images par classe pour l'apprentissage. D'autres évaluations seront bientôt comparées à celle-ci à l'issue du *Challenge Caltech 2007*[7].

Nous sommes encore loin d'atteindre les 30000 objets que l'homme peut reconnaître [9], mais les résultats de ces dernières années, obtenus grâce au développement des bases d'images, à l'augmentation des puissance des machines et à l'efficacité des techniques s'appuyant sur les points d'intérêt, sont plutôt prometteurs.

2.3.7 Techniques d'évaluation

Une mesure classique en recherche d'information, et qui peut être appliquée ici, est la mesure de la précision et du rappel. Ces deux valeurs, comprises entre 0 et 1, sont définies de la manière suivante. Le rappel est le rapport entre le nombre d'images pertinentes retrouvées et le nombre d'images pertinentes présentes dans la base. Un rappel de 1 sera donc obtenu si toutes les images pertinentes ont été retrouvées. La précision est le rapport entre le nombre d'images pertinentes retrouvées et le nombre d'images retournées. On aura donc une précision de 1 si toutes les images retrouvées par le système sont pertinentes.

Soit p le nombre total d'images pertinentes, N le nombre d'images retournées par le système et r le nombre d'images pertinentes retournées (on a $r \leq p$ et $r \leq N$), le rappel R et la précision P s'écrivent :

$$R = \frac{r}{p} \text{ et } P = \frac{r}{N}$$

On présente souvent ces mesures sous la forme d'une courbe $P = f(R)$, qui est en général décroissante. Pour obtenir différentes valeurs, on fait varier le nombre d'images que le système retourne qui est un paramètre du système.

Dans le cas précis de l'annotation d'images, il faut évaluer chaque mot séparément en utilisant une vérité terrain, c'est-à-dire une annotation manuelle de la base de données d'images. Pour un mot donné qui a été annoté automatiquement, p est le nombre d'images ayant été annotées manuellement avec ce mot, N est le nombre d'images ayant été annotées automatiquement avec ce mot, et r est le nombre d'images appartenant à ces deux ensembles, c'est-à-dire dont l'affectation automatique de ce mot est correcte. On fait ensuite une moyenne sur tous les mots pour avoir un score unique pour le système.

La MAP (*mean average precision*) se définit de la manière suivante :

$$MAP = \frac{\sum_{n=1}^{N}(P(n) \times rel(n))}{N}$$

où N est le nombre d'images que le système doit retourner, $P(n)$ la précision calculée sur les n premières images retournées et $rel()$ une fonction binaire qui vaut 1 si le document

[7] http ://vision.caltech.edu/CaltechChallenge2007/

au rang n est pertinent et 0 sinon. Cette mesure est utilisée notamment dans la campagne ImagEval (section 4.2.3).

Dans le cas où toutes les réponses retournées sont bonnes, $P(n)$ vaut toujours 1 quel que soit n, et donc la MAP vaut 1 également. L'intérêt de la MAP est de pénaliser davantage une erreur faite sur une des premières images retournées qu'une erreur faite sur une des dernières. Dans un système où deux réponses sont données, la MAP vaut 1 si les deux réponses sont justes, 0,75 si seule la deuxième réponse est fausse, 0,25 si seule la première réponse est fausse, et 0 si les deux réponses sont incorrectes.

Une autre mesure que l'on retrouve notamment dans Barnard et al. [6] est le score normalisé (*normalized score*, NS). En plus des notations précédentes, on note N le nombre total d'images de la base. La définition du score normalisé est la suivante :

$$NS = \frac{r}{p} - \frac{n-r}{N-p}$$

Le premier terme est le nombre d'éléments pertinents retrouvés divisé par le nombre d'éléments pertinents total : c'est le rappel défini ci-dessus. On cherche à ce que ce terme se rapproche de 1. Le deuxième terme est le nombre d'éléments non pertinents retrouvés divisé par le nombre d'éléments non pertinents total. Ce terme doit être minimisé. Ce score est compris entre -1 et 1. Il vaut 1 si toutes les images pertinentes et uniquement celles-ci ont été retrouvées ($r = n = p$). Il vaut -1 si, en revanche, toutes les images non pertinentes ont été retrouvées, sans en retrouver une seule de pertinente ($n = N - p$ et $r = 0$). Il peut valoir 0 si toutes les images de la base sont retournées ($r = p$ et $n = N$), ou aucune ($r = 0$, $n = 0$).

L'intérêt de cette mesure par rapport à la mesure de précision rappel classique est qu'elle permet de prendre en compte les éléments non pertinents (le bruit) présents dans la base : ce n'est pas la même difficulté de retrouver 10 éléments pertinents parmi 100 que d'en retrouver 10 parmi 1000.

2.3.8 Systèmes de démonstration en ligne

Des systèmes en ligne sur Internet concernant l'annotation automatique d'images n'ont vu le jour que très récemment. Je n'en ai recensé que deux au moment où j'écris ces lignes (2007), et parmi ces deux systèmes, un seul propose d'annoter n'importe quelle image de l'utilisateur, ce qui rajoute au système la contrainte d'être capable d'annoter les images en un temps admissible pour un utilisateur quelconque, c'est-à-dire en un temps de l'ordre de quelques secondes.

Le système *ALIPR*[8] (*Automatic Linguistic Indexing of Pictures - Real Time*) créé par le professeur James Wang de l'université de Pennsylvanie vise à annoter des images de type photographies en couleur. Nous fournissons en entrée au système une image, et en sortie, nous obtenons la liste des 15 mots les plus probables parmi les 332 mots qu'il a appris à reconnaître[9]. Les détails techniques sont donnés dans Li et al. [75]. La

[8] Site web de ALIPR : http ://www.alipr.com/

[9] La liste des mots est disponible sur http ://www.alipr.com/words.html

base de données utilisée pour l'apprentissage est la base Corel contenant 60 000 images classées en 599 classes décrites par 332 mots distincts. Pour l'apprentissage, 80 images par concept (mot) sont utilisées. Chaque pixel de l'image est classé en utilisant ses 3 valeurs dans l'espace de couleur LUV et 3 coefficients (HL, LH et HH) de sa transformée en ondelettes de Daubechie. Ces valeurs sont regroupées pour fournir une segmentation de l'image. Pour chaque région, la couleur moyenne, la texture moyenne et la surface sont extraites. Un modèle de mélange est ensuite estimé pour chaque mot à partir de ces informations. L'annotation, consistant en l'extraction des caractéristiques d'une image et en la classification suivant le modèle appris, dure en moyenne 1,4 secondes par image sur un processeur cadencé à 3GHz. Ce système offre également la possibilité de soumettre nos propres images.

L'article [75] décrivant ce système déclare que parmi les 15 mots-clés proposés, au moins un est pertinent dans 98% des cas. Cependant, en pratique, cette statistique n'est pas significative. Si nous considérons par exemple une image où 1 mot-clé proposé est correct, et les 14 autres sont faux, cette image fait partie des 98% annoncés, mais la précision correspondante pour les mots-clés n'est en moyenne que de 6,7%. D'autre part, des mots-clés très généraux ou obscurs sont générés, tels que *photo*, *natural* et *thing*, et la chance d'avoir des mots-clés pertinents peut être facilement augmentée en rajoutant systématiquement des concepts souvent présents dans les photographies tels que *sky*, *wild life*, *man made*, *people* et *building*.

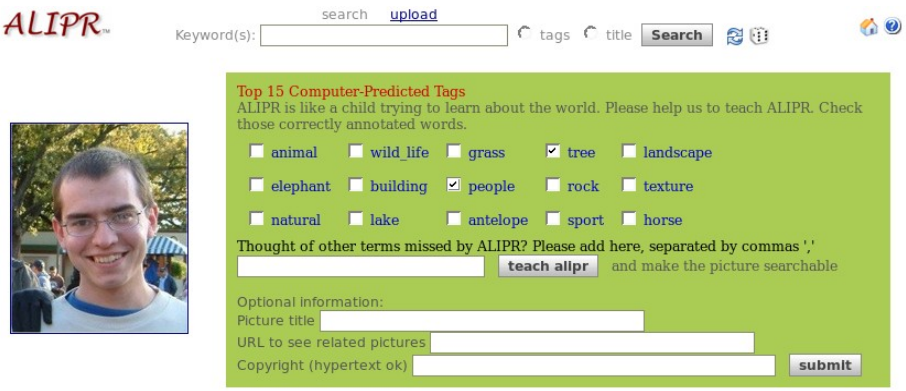


FIG. 2.17 – Exemples de résultats avec une photographie personnelle. Parmi les 15 mots clés générés, seuls *people* et *tree* sont corrects, et éventuellement *building* et *natural*.

Le résultat de l'annotation pour une photographie personnelle est montré sur la figure 2.17. Étant donné que les mots ne sont pas classés alphabétiquement, nous pouvons supposés qu'ils sont classés par probabilité, comme c'était le cas dans le système ALIP[10], prédécesseur de ALIPR. Dans ce cas, il est étonnant de voir que les trois premiers mots clés sont incorrects alors que le visage et les arbres sont a priori faciles à détecter. Il serait intéressant dans ce système d'associer des régions aux mots clés, pour comprendre notamment quelle partie de l'image est responsable des mots *elephant*, *antelope* et *horse*.

[10]http ://wang.ist.psu.edu/IMAGE/alip.html

Ce système est représentatif de l'état de l'art dans le domaine et de la difficulté de la tâche : des mots-clés sont générés avec une précision faible quand le nombre d'objets à reconnaître dépasse la centaine, et la liste des mots-clés n'est pas cohérente et contient souvent des contradictions. Notamment, pour la description de l'image sur la figure 2.17, il y a une opposition entre tout ce qui concerne la nature (*animal*, *wild life*, *elephant*, *antelope*) et la détection de *building*, qui sont également difficilement compatibles avec *sport*. Toutefois, ALIPR permet de faire du retour de pertinence en demandant à l'utilisateur de sélectionner parmi les mots-clés proposés ceux qui sont corrects afin de raffiner l'apprentissage des concepts et nous pouvons espérer que le système tendra à s'améliorer avec le temps.

Le système *Behold*[11] [152] propose d'apprendre certaines scènes ou certains concepts et d'utiliser cela pour améliorer la recherche d'images par similarité. Ce système ne propose pas directement de l'annotation d'image comme le système *ALIPR*, car l'auteur a jugé que cette application ne donne pas à l'heure actuelle des résultats suffisamment bons pour être utiles dans une application pratique de description d'images, mais qu'en revanche, la reconnaissance de scènes ou de concepts généraux est suffisamment bonne pour être utilisée pour raffiner une recherche d'images par similarité. *Behold* utilise comme ressource deux bases d'images au choix : 101 000 images de Flickr ou 1 131 605 images de sites web d'universités anglaises, américaines et canadiennes. Pour ces deux bases, les images sont accompagnées d'annotations manuelles qui sont également utilisées par Behold. Le système propose deux modes d'interrogations dont tout l'intérêt est de pouvoir les combiner :

- un mode qui utilise simplement les annotations manuelles de Flickr, où l'on est libre du concept demandé et qui retourne les images correspondant à ce concept,
- un mode retournant des images dont le contenu visuel ressemble à celui des images annotées avec un certain mot.

Par exemple, le premier mode permet de demander des images annotées « Londres ». Le deuxième mode permet de retrouver des images qui ressemblent à des images de « bâtiment ». Pour ce deuxième mode, un apprentissage est effectué sur certains concepts, en utilisant comme base les images annotées avec ce concept dans Flickr, par exemple ici les images annotées « bâtiment », mais permet de retrouver des images de « bâtiment » qui ne seraient pas annotées comme telles. Enfin, la combinaison de ces deux modes donne la possibilité par exemple de demander des images de « Londres » semblables à des images de « bâtiment ». Pour le moment (juin 2007), 56 concepts ont été appris pour la recherche par le contenu, dont 38 ont le statut de mots-clés expérimentaux, mais le nombre de mots-clés augmente au fil des mois. Pour donner une idée, les 18 mots-clés non expérimentaux sont : *plage, bateau, bâtiment, ville, nuage, visage, champ, fleur, forêt, herbe, lac, silhouette, ciel, coucher de soleil, texture, tour, sous-marin, vague*[12]. Des exemples de mots-clés expérimentaux sont : *vue aérienne, animal, art, oiseau, voiture.*

[11] Behold signifie en anglais « regarder avec attention ». Site web de Behold : http ://photo.beholdsearch.com

[12] En anglais : *beach, boat, building, city, cloud, face, field, flower, forest, grass, lake, silhouette, sky, sunset, texture, tower, undersee, wave*

La combinaison des deux modes peut aussi être vue comme un moyen de réorganiser les images issues d'une requête d'après leur contenu. Un exemple de comparaison entre une requête n'utilisant que les annotations manuelles et la même requête où les images sont visuellement réorganisées est présenté dans la figure 2.18.

Techniquement, *Behold* utilise les travaux de la thèse de Yavlinsky [153] pour annoter des images et attribuer un pourcentage de confiance à chaque mot-clé. Ces travaux ont été décrits précédemment dans la section 2.3.1.

Notons enfin que les moteurs de recherche classiques d'images sur Internet commencent à intégrer des fonctions nécessitant de la classification d'images, notamment de la reconnaissance de visages chez *Exalead* et *Google* afin de n'afficher que des images contenant des visages lors de requêtes sur des noms de personnes, quoique pour le moteur *Google*, il n'est pas clairement précisé s'il s'agit de reconnaissance automatique, ou de l'utilisation des annotations manuelles de *Google Image Labeler*. *Google*, *Yahoo !* et *Exalead* proposent également de n'obtenir que des images de l'une des trois classes suivantes : couleur, niveaux de gris ou noir et blanc (la classe niveaux de gris n'est pas disponible sur le moteur de *Yahoo !*).

2.4 Conclusion de l'état de l'art

Nous avons vu dans cet état de l'art que les descripteurs calculés pour faire de l'annotation automatique d'images sont très variés, mais tendent à converger pour les méthodes affichant les meilleurs résultats vers le calcul de caractéristiques sur la texture et la couleur autour de points d'intérêt et la quantification de ces caractéristiques par des sacs de mots pour ensuite faire de l'apprentissage. Les descripteurs que nous utiliserons dans cette thèse sont pour la plupart des histogrammes de textures et de couleurs calculés sur toute l'image ou sur des régions issues d'une segmentation automatique ou d'une segmentation en grille. Les techniques s'appuyant sur les points d'intérêt et les sacs de mots n'ont malheureusement pas pû être évaluées par manque de temps.

En ce qui concerne les algorithmes d'apprentissage, nous avons choisi d'en fixer un et de se limiter à celui-ci afin de ne pas passer du temps à évaluer la différence entre tous les algorithmes. Notre choix s'est porté sur les SVMs dont l'efficacité pour les problèmes de classification, notamment d'images, a déjà été prouvée par de nombreux travaux. D'autre part, de nombreuses librairies, dont libSVM que nous utilisons, sont librement disponibles et bien documentées. Nous avons également mis en œuvre la technique de boosting (Adaboost) pour la reconnaissance de visages.

En ce qui concerne la base d'images pour évaluer nos algorithmes, nous avions commencé avec la base Corel puis avons changé rapidement pour Caltech-101 lorsqu'elle fut disponible, car elle est mieux adaptée à l'évaluation de la classifications d'images. Nous avons également mené quelques tests sur Caltech-256.

Enfin, les contributions principales de cette thèse portent sur deux points qui sont actuellement peu développés dans la littérature comme nous l'avons vu : d'une part la possibilité d'exploiter Internet pour créer de manière complètement automatique des bases d'images pour l'apprentissage d'objets et d'autre part l'apport de la sémantique (les

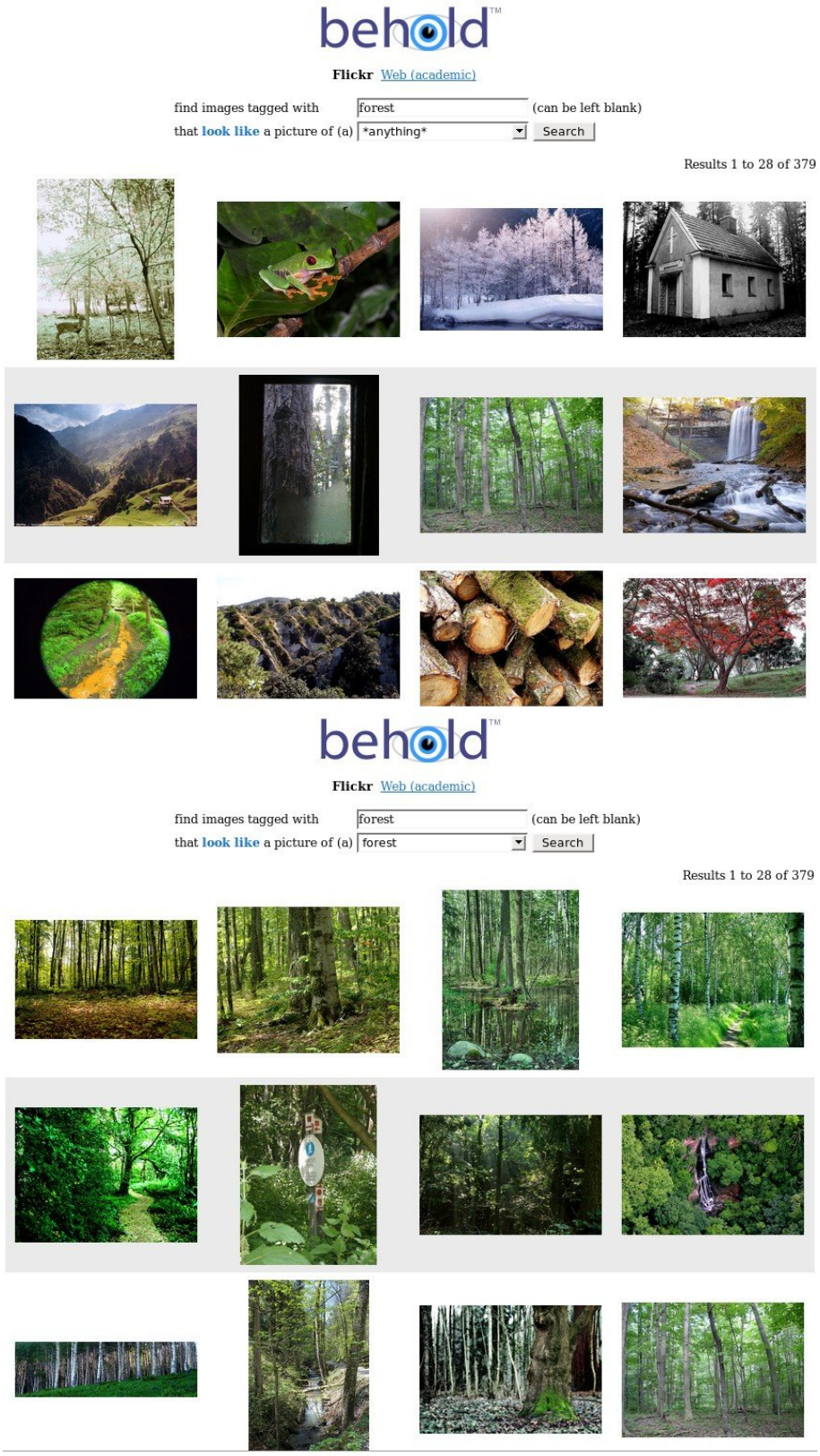


FIG. 2.18 – Exemples de résultats obtenus avec le système Behold en utilisant le contenu des images pour réorganiser les images pour la requête « forêt ». En haut : utilisation uniquement des annotations, en bas : utilisation du contenu pour trouver parmi les images annotées « forêt » celles qui ressemblent effectivement le plus au concept « forêt ».

relations spatiales et le contexte des objets) dans un système d'annotation automatique par rapport à un système reconnaissant les différents objets de la même manière sans considérer leur signification.

Chapitre 3

Descripteurs utilisés dans cette thèse

La perception d'une image naturelle est basée sur la reconnaissance des formes ou des structures sous-jacentes qui y figurent.
Ernst Cassirer

De nombreux descripteurs existent, regroupés en général suivant les catégories texture, couleur ou forme. Nous donnons d'abord quelques références vers des états de l'art des descripteurs utilisés dans la littérature, puis détaillons dans ce chapitre ceux que nous utiliserons par la suite. Nous décrivons également le descripteur des sacs de mots que nous n'avons pas utilisé au cours de nos travaux, mais qui prend de plus en plus d'importance en annotation automatique d'images.

Un état de l'art des descripteurs les plus utilisés dans les travaux d'indexation d'images est disponible dans l'article de Smeulders et al. [126] publié en 2000. Ils divisent les descripteurs en trois catégories classiques : couleurs, forme et texture. Un état de l'art un peu plus récent (2002), et en français est présenté dans la thèse de Jérôme Fournier [50]. Il rajoute une quatrième catégorie : les descripteurs par points d'intérêts, qui sont une technique récente donnant des résultats prometteurs, notamment avec les descripteurs SIFT [83] appris avec la technique des sacs de mots développée en section 3.4. Il discute également de la possibilité d'inclure des informations spatiales dans les signatures des descripteurs classiques, et détaille les mesures de similarités classiques qui sont utilisées en recherche d'images par similarité.

Nous avons choisi dans le cadre de nos travaux sur l'annotation automatique d'images de nous concentrer principalement sur des descripteurs rapides à calculer. Nous souhaitions en effet conserver la contrainte d'être capable d'annoter une nouvelle image en un temps acceptable pour un utilisateur, c'est-à-dire en quelques secondes (moins de 10).

3.1 Couleur

3.1.1 RVB

Le descripteur RVB le plus utilisé est RVB-64 où chacune des trois composantes rouge, vert et bleu est quantifiée en quatre valeurs. L'application d'une telle transformation sur l'image exemple 3.1 est montrée sur la figure 3.2.

FIG. 3.1 – Image originale exemple utilisée pour illustrer les différents descripteurs présentés par la suite.

FIG. 3.2 – Quantification des trois plans rouge, vert, bleu de l'image 3.1 en quatre valeurs. En-dessous : image à 64 couleurs correspondante.

3.1.2 TSVal

L'espace TSVal – pour Teinte, Saturation, Valeur – est un espace de couleur qui peut être calculé à partir du RVB, et qui est plus commode pour associer un nom à une

couleur que l'espace RVB. Nous calculons les trois composantes T, S et Val à partir de RVB par les relations suivantes :

$$
\begin{aligned}
Val &= max(R,V,B) \\
S &= \frac{max(R,V,B)-min(R,V,B)}{max(R,V,B)} \\
T &= 60 * \begin{cases} \frac{V-B}{max(R,V,B)-min(R,V,B)} & \text{si } R = max(R,V,B) \\ 2 + \frac{B-R}{max(R,V,B)-min(R,V,B)} & \text{si } G = max(R,V,B) \\ 4 + \frac{R-V}{max(R,V,B)-min(R,V,B)} & \text{si } B = max(R,V,B) \end{cases}
\end{aligned}
$$

De même que pour RVB, il est possible de construire un histogramme dans cet espace, mais toutes les composantes n'ont pas la même importance. Notamment, la composante de la teinte doit être quantifiée plus finement que les autres. Nous proposons de faire une quantification de la teinte en 18 valeurs (17 + 1 pour la teinte négative, correspondant aux pixels achromatiques comme expliqué en annexe), et les deux autres composantes sont quantifiées en 3 valeurs chacune, ce qui nous donne un histogramme à 162 composantes.

3.1.3 RVB-64-9

L'idée du descripteur que nous appelons RVB-64-9 est de rajouter des informations spatiales au descripteur RVB-64 précédemment décrit.

Chaque image est d'abord découpée en 9 régions régulières (3×3), puis le descripteur *RVB-64* est appliqué sur chacune des sous-images obtenues, comme illustré sur la figure 3.3.

FIG. 3.3 – Découpage de l'image en 9 régions égales.

Les histogrammes de chaque région sont mis bout à bout, résultant en un histogramme à 576 composantes.

3.1.4 BIC

BIC est l'abréviation de *border interior classification*. Ce descripteur proposé par Stehling et al. [128] commence par une quantification classique de la couleur des pixels, telle que par exemple l'une des quantifications dans l'espace RVB ou TSVal décrites ci-dessus. Ensuite, deux types de pixels sont distingués : les pixels intérieurs, qui sont

des pixels ayant la même couleur que leurs quatre voisins en 4-connexité, et les pixels de bord, qui sont les autres pixels, dont au moins un voisin n'a pas la même couleur. Pour se donner une idée des pixels ainsi considérés, ces deux types sont séparés comme illustré sur la figure 3.4.

FIG. 3.4 – Représentation de l'ensemble de chacun des deux types de pixels considérés pour le descripteur BIC. À gauche : ensemble des pixels intérieurs ; à droite : ensemble des pixels de bord.

Un histogramme est construit pour chacun des deux types de pixel. Par exemple, avec une quantification RVB en 64 couleurs, l'histogramme total obtenu a 128 composantes. Nous utilisons en pratique le plus souvent ce descripteur avec une quantification RVB en 216 (6^3) couleurs, correspondant à un histogramme à 432 composantes.

3.2 Texture

3.2.1 LEP

LEP (*local edge pattern*, motifs des contours locaux) est un descripteur de texture fournissant un histogramme à 512 composantes proposé par Cheng et al. [23].

Une image des gradients avec des valeurs entre 0 et 255 est d'abord calculée avec un filtre de Sobel 3x3, puis cette image est binarisée par seuillage avec $s = 100$. Un exemple est montré sur la figure 3.5.

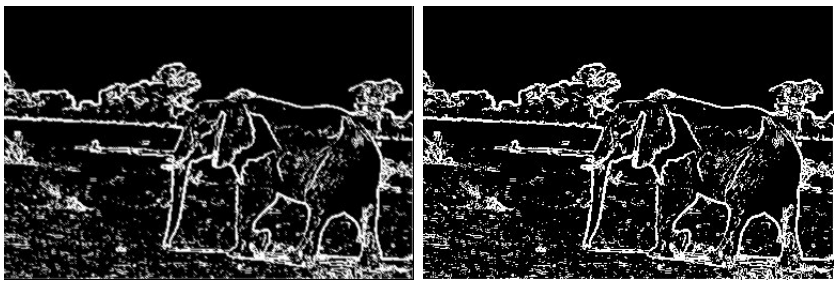


FIG. 3.5 – À gauche : résultat de l'application d'un filtre de Sobel sur l'image 3.1. À droite : obtention d'une image binaire à partir de l'image de Sobel, par seuillage.

Ensuite, pour chaque pixel de cette image, on considère la fenêtre 3×3 autour de ce pixel. Il y a $2^9 = 512$ configurations possibles que l'on numérote en utilisant le masque

3×3 présenté sur la figure 3.1, et on associe au pixel central le numéro de la configuration dans laquelle il est. On peut alors construire un histogramme de 512 composantes.

1	2	4
8	256	16
32	64	128

TAB. 3.1 – Filtre binomial LEP

3.2.2 Gabor

Les filtres de Gabor (ou ondelettes de Gabor) ont été introduits par Daugman [28] dans les années 1980 en s'inspirant de la perception des fréquences et des orientations par le cortex visuel. Ils se définissent comme étant le produit d'une gaussienne et d'une sinusoïde orientée dont on fait varier la fréquence et l'orientation pour définir un banc de filtres. Chaque filtre $\Psi(x, y)$ se définit par :

$$\Psi(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2})} e^{j2\pi Wx}$$

Le paramètre W permet de changer l'orientation. σ_x et σ_y contrôlent l'échelle suivant deux directions orthogonales, d'angles $2\pi W$ et $2\pi W + \pi$. Les filtres sont d'une taille de 8×8. Nous utilisons les formules de Zhang et al. [155] afin de déterminer les différents paramètres W, σ_x et σ_y à utiliser en fonction du nombre d'orientations et d'échelles désirées. Avec 4 directions et 3 échelles, nous obtenons le banc de filtres représenté sur la figure 3.6.

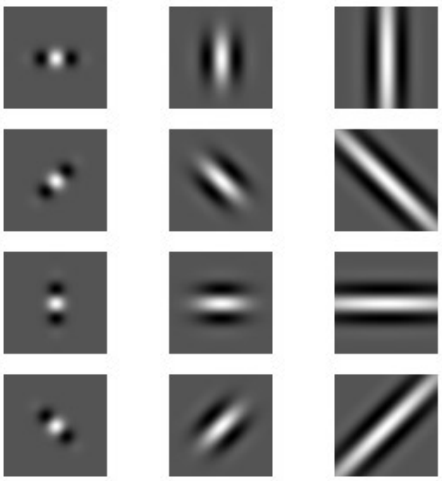


FIG. 3.6 – Banc de 12 filtres de Gabor obtenus avec 4 directions et 3 dimensions.

Les informations retenues pour ce descripteur sont alors l'énergie et la variance de l'image obtenues en sortie de chaque filtre.

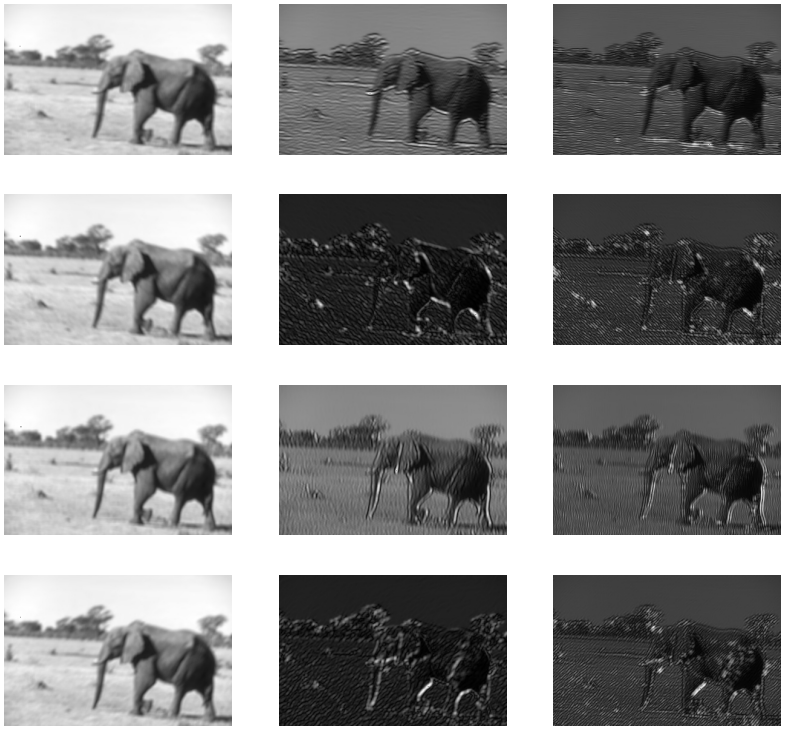


FIG. 3.7 – Réponses correspondant aux filtres de la figure 3.6.

L'utilisation des filtres de Gabor, contrairement aux descripteurs présentés précédemment, est très coûteuse en temps de calcul, à cause de la convolution de l'image entière avec des filtres de taille 8×8. À cause de cela, nous nous limitons en pratique à l'utilisation de 24 filtres au plus : 6 directions et 4 échelles, générant un histogramme à 48 dimensions. Il faut en effet garder à l'esprit que l'une des contraintes des travaux proposés dans cette thèse est d'être capable d'annoter une image en un temps raisonnable, typiquement inférieur à 10 secondes. Le temps nécessaire pour calculer la réponse des 24 filtres est alors d'environ une seconde pour une image 300×300.

3.3 Forme

Les descripteurs classiques de forme cherchent à caractériser un contour. C'est le cas par exemple du descripteur de Fourier. Les contours ainsi étudiés sont en général issus d'une image binaire représentant une forme (blanche) sur un fond (noir), ou d'une région délimitant précisément la surface occupée par un objet dans une image donnée. Dans le cadre de cette thèse, nous nous plaçons dans l'optique où l'on cherche à reconnaître les objets contenus dans une image donnée, mais sans avoir de connaissance précise sur les contours de ces objets dans l'image. Nous utilisons, il est vrai, un algorithme de segmentation automatique pour extraire des régions de l'image et chercher à y reconnaître des objets, mais les contours de telles régions sont trop imprécis pour permettre d'utiliser

des descripteurs tels que celui de Fourier. Le descripteur *projection* que nous détaillons ici n'est pas à proprement parler un descripteur de forme, mais est lié toutefois aux contours contenus dans l'image.

3.3.1 Projection

L'image est d'abord redimensionnée en 100×100 puis les contours sont extraits avec un filtre de Sobel et une binarisation avec $s = 100$. Les deux moitiés verticales sont d'abord considérées : pour chacune des moitiés, les sommes de chaque ligne servent à construire un vecteur de projection de dimension 100. Le même procédé est appliqué aux deux moitiés horizontales en faisant la somme suivant les colonnes. Nous obtenons au final un vecteur de dimension 400.

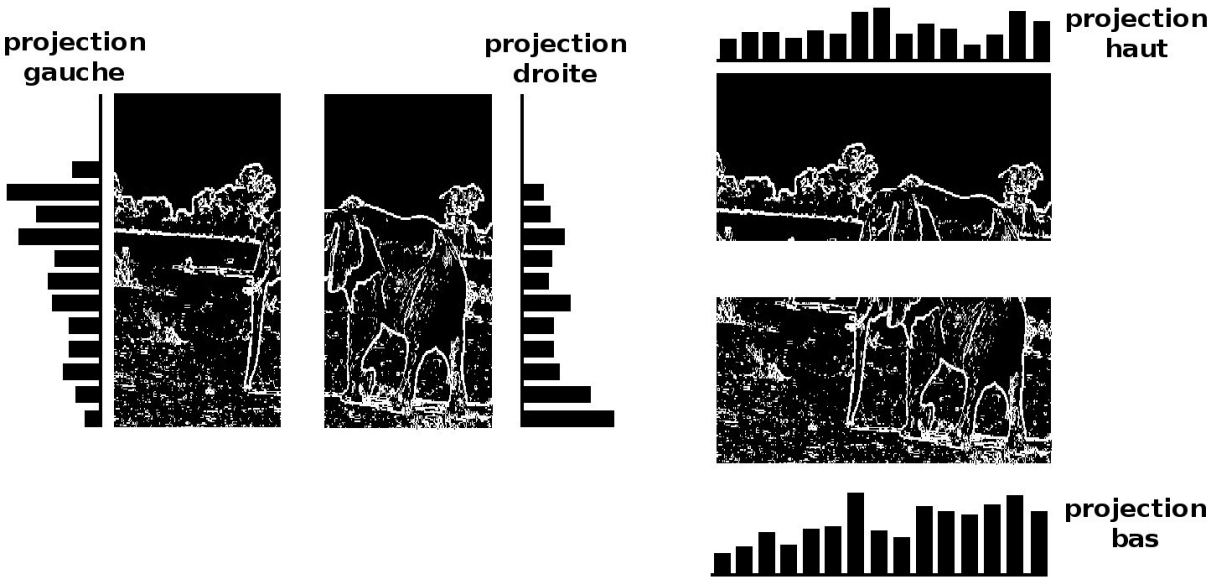


FIG. 3.8 – Calcul du descripteur projection sur l'image 3.1.

3.4 Sacs de mots

Contrairement aux descripteurs précédemment, nous n'utilisons pas ce descripteur dans le cadre de cette thèse, mais nous le présentons, car nous pensons qu'il s'agit d'un descripteur important dans le domaine de l'annotation d'images, qui pourrait notamment être exploité dans les perspectives de nos travaux.

La technique dite de « sacs de mots » (en anglais : *bag of words*) est issue du domaine de l'analyse de documents textuels. Elle a été adaptée à l'analyse d'images pour la première fois par Csurka et al. [25] qui lui ont alors donné le nom de *bag of keypoints*.

L'expression « sac de mots » signifie que les mots sont considérés statistiquement, c'est-à-dire par leur présence et leur fréquence dans un document, sans prendre en compte la position de chaque mot en absolu dans le document, ou par rapport aux autres mots. Pour les documents textuels, l'idée est de réunir dans un même « sac » tous les synonymes correspondant à un même concept, ou les mots issus d'un même lemme, puis de considérer ce sac comme l'élément à utiliser pour l'apprentissage.

Dans le cas de l'utilisation en traitement d'images, un dictionnaire visuel est construit par une quantification de caractéristiques visuelles telles que la couleur ou la texture, pour faire l'analogie avec les mots dans le cas de l'analyse de documents textuels. Csurka et al. [25] énumèrent les étapes suivantes pour cet algorithme :

- sélectionner des régions ou des points d'intérêt, par exemple avec un détecteur de Harris [60],
- calculer des descripteurs locaux à partir de ces régions ou points, notamment avec le descripteur SIFT [83]
- attribuer une classe à chaque descripteur local par rapport à des groupes (le vocabulaire) qui ont été déterminés au préalable à partir d'un regroupement des descripteurs locaux calculés sur une base d'images
- construire le sac de mots en comptant le nombre de régions ou points associés à chaque groupe
- utiliser un classifieur multiclasse considérant le sac de mots comme un vecteur de descripteurs.

Ces étapes sont illustrées sur la figure 3.9.

Les centres des groupes, i.e. le vocabulaire visuel, obtenus par Fei-fei et Perona [42] sont affichés sur la figure 3.10. Les auteurs font la remarque qu'il est intéressant de constater que la plupart de ces régions locales sont de simples orientations et gradients d'illumination, qui correspondent à ce que peuvent discriminer les premiers neurones de notre système visuel.

Il est également possible de pousser l'analogie avec le texte jusqu'à l'élimination de « *stop words* », c'est-à-dire de mots visuels qui sont très fréquents et perturbent les résultats comme l'a montré Josef Sivic dans sa thèse [124].

En classification d'images, la technique des sacs de mots est le plus souvent combinée avec le descripteur local SIFT [15, 25, 42, 73, 117, 125]. C'est la technique que l'on retrouve actuellement le plus souvent dans les travaux qui obtiennent les meilleurs résultats en reconnaissance d'objets.

3.5 Techniques pour choisir

Dans cette thèse, nous utilisons les descripteurs présentés ci-dessus pour faire de l'apprentissage en entraînant des séparateurs à vaste marge. Il est facile dans ce cadre de combiner plusieurs descripteurs : il suffit de les mettre bout à bout pour constituer un vecteur de grande dimension et d'utiliser ce vecteur pour faire de l'apprentissage et de la classification. Toutefois, le temps d'apprentissage d'un SVM est d'autant plus long que la dimension du vecteur est importante. Il est alors intéressant de pouvoir limiter la

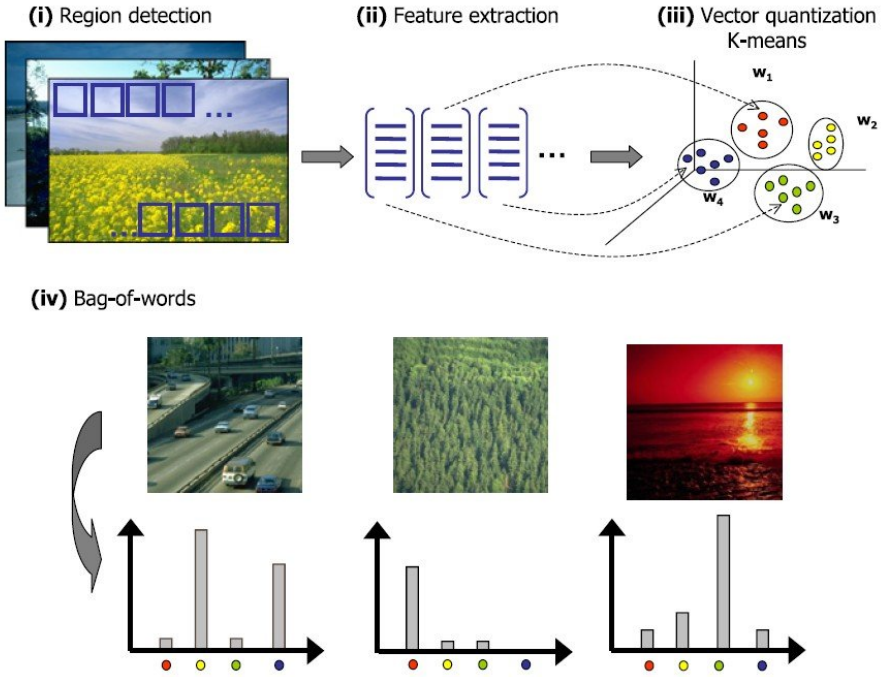


FIG. 3.9 – Schéma illustrant le calcul des sacs de mots en analyse d'images, d'après Bosch et al. [16].

taille de ce vecteur. Nous avons expérimenté deux solutions. La première consiste à faire une analyse en composantes principales des données, la seconde à limiter le nombre de descripteurs que nous combinons.

L'analyse en composantes principales (ACP) est une technique qui permet de réduire la dimension d'un espace de données en créant un nouvel espace de plus faible dimension construit à partir des directions de l'espace initial selon lesquelles on observe les plus fortes variations des données. Si l'on souhaite par exemple réduire l'ensemble des données en un espace à 100 dimensions, le nouvel espace est construit en utilisant les 100 premiers vecteurs singuliers correspondant aux 100 plus grandes valeurs singulières de la matrice rectangulaire contenant les vecteurs des descripteurs pour toutes les images de la base d'apprentissage.

Nos expériences sur la base Caltech-100 nous ont montré que nous n'avions pas besoin d'utiliser une telle ACP dans notre cas. En effet, les résultats en classification sont à peu près les mêmes sur l'espace réduit et sur l'espace initial (cf. annexe C). Cela signifie que l'ACP a conservé l'essentiel des informations utiles à la classification, mais aussi que l'apprentissage du SVM se passe bien également avec de très grandes dimensions, notamment sans que nous observions de problème de sous-apprentissage, qui tendrait à dégrader les performances de classifications lorsque la dimension du vecteur de données (donc le nombre de descripteurs) augmente, à nombre d'images égal. D'autre part, le

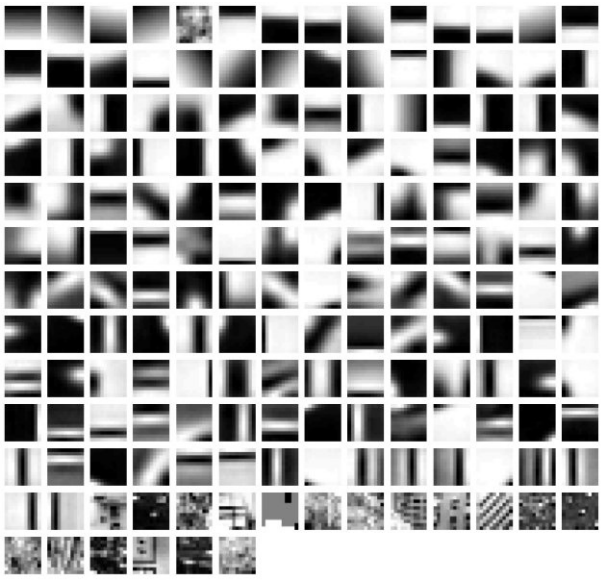


FIG. 3.10 – Centre des sacs de mots obtenus par Fei-fei et Perona [42], avec les sacs contenant le plus grand nombre d'éléments en premier.

gain au niveau du temps de calcul n'est pas assuré. L'apprentissage sur nos données avec un vecteur de taille 2000 dure quelques dizaines de minutes, et il n'est plus que de quelques minutes avec l'espace réduit à 100 dimensions. En revanche, l'ACP a duré plusieurs heures. Ces temps n'interviennent que pour l'apprentissage, et il faut comparer également les temps de classification. Pour la classification d'une nouvelle image, il faut d'abord calculer les descripteurs de cette image, ce qui prend typiquement quelques secondes, mais les mêmes descripteurs sont à calculer, que l'on se place dans l'espace initial ou l'espace réduit. Ensuite, les temps de classification sont très courts (inférieurs à la seconde), quelle que soit la taille de l'espace utilisé, et la réduction du vecteur est également très rapide. Il n'y a donc pas de gain intéressant de temps, l'essentiel du temps étant occupé par le calcul des descripteurs. Des techniques dérivées plus performantes que l'ACP existent telles que l'analyse en composantes indépendantes, mais nous n'avons pas testé cela.

La deuxième possibilité que nous avons expérimenté pour réduire la taille de l'espace des données est de limiter le nombre de descripteurs utilisés. Pour cela, il faut chercher à savoir lesquels sont redondants, et lesquels, au contraire, font ressortir des aspects différents de l'image, apportant davantage d'information lorsqu'ils sont combinés. Typiquement, et pour comprendre cette notion, la combinaison des descripteurs RVB-64 et RVB-125 donne des résultats comparables à l'utilisation de l'un des deux uniquement. Au contraire, la combinaison d'un descripteur de couleur et d'un descripteur de texture donne une très nette amélioration des résultats par rapport à ceux obtenus avec un seul descripteur. Notre étude sur les diverses combinaisons de nos descripteurs est détaillée

dans l'annexe C.

Chapitre 4

Proposition d'une classification d'images hiérarchique

Inanimate objects are classified scientifically into three major categories - those that don't work, those that break down and those that get lost.
Russell Baker

Afin d'annoter les images, la première étape que nous proposons est d'analyser la scène représentée par l'image, pour savoir quelle est la nature et le type de l'image que l'on doit traiter. Savoir quel type de scène est représenté par l'image (scène d'intérieur / d'extérieur, de ville, de nature, etc.) permettra par la suite de guider la reconnaissance d'objets, et notamment de lever les ambiguïtés pour améliorer cette reconnaissance, comme nous le verrons au chapitre 7. Afin de classer ces images, nous proposons un arbre de classification.

4.1 Arbre de classification

L'arbre de classification que nous avons défini dans cette thèse s'inspire des travaux de classification de scènes que nous avons décrits dans l'état de l'art (section 2.3.1). Il s'inspire également des observations que nous avons faites sur de nombreuses images provenant d'Internet, et nous avons construit cet arbre dans l'optique de pouvoir classifier la plupart de ces images. Cet arbre est représenté sur la figure 4.1.

Au premier niveau de l'arbre, deux classifications indépendantes sont effectuées : la classification sur la nature de l'image (carte, clipart, photographie ou peinture) et sur le type de l'image (couleur, noir et blanc, noir et blanc colorisé). Nous supposons

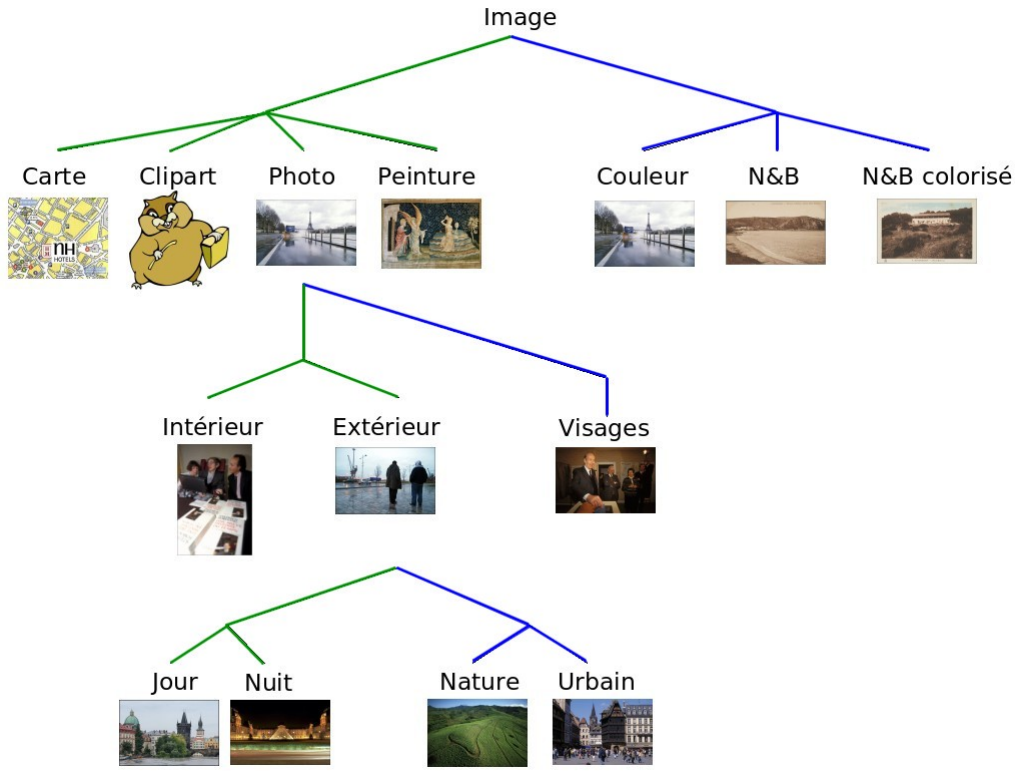


FIG. 4.1 – Arbre utilisé pour la classification des scènes en catégories sémantiques.

que chaque image appartient à deux classes : une nature et un type. Pour la nature de l'image, les classes sont définies comme suit :

- **Carte** : Images similaires à des cartes routières ou des cartes géographiques. Ces images se caractérisent en général par la présence de traits (les routes ou les fleuves) dans plusieurs directions.
- **Clipart** : Images qui sont des dessins simples tracés avec un logiciel. Les copies d'écran et les schémas de type scientifiques sont aussi inclus dans cette catégorie.
- **Peinture** : Œuvres d'art peintes qui ont été photographiées.
- **Photo** : Scènes photographiées par un appareil photographique numérique, ou numérisation de photographies prises par un appareil argentique, et qui n'appartiennent pas à la catégorie « peinture ».

Nous distinguons également trois types d'images :

- **Noir et blanc** : Catégorie constituée en fait de trois sous-catégories distinctes : les images binaires qui ne contiennent que deux couleurs, le noir et le blanc ; les images en niveaux de gris, par exemple constituées de 256 couleurs qui sont des variantes de blanc, de gris et de noir ; les images de type vieille carte postale, comme par exemple les photographies sépia, qui sont équivalentes à une image en niveaux de gris à laquelle on aurait rajouté une teinte uniforme sur toute la surface.

- **Noir et blanc colorisé** : Images originellement en niveaux de gris, qui ont été retouchées à la main pour y rajouter des couleurs par endroit. Ce sont en général également de vieilles cartes postales, aux couleurs peu vives.
- **Couleur** : Images en couleur qui n'appartiennent à aucune des deux autres classes ci-dessus.

Ces deux classifications servent à séparer des types d'images pour lesquels des traitements différents devraient être vraisemblablement appliqués. Par exemple, des descripteurs de couleurs n'ont d'intérêts que pour les images de la catégorie « couleur », et la recherche d'objets dans les photographies est a priori très différente de la recherche d'objets dans les cliparts où la symbolique a plus d'importance. En pratique, nous nous intéressons principalement aux photographies en couleur et les autres classes servent à mettre de côté les images que nous ne souhaitons pas traiter.

Ensuite, les images identifiées comme photographie (en couleur ou en niveaux de gris), sont séparées en deux groupes : les photographies prises à l'intérieur d'un bâtiment ou dans un lieu clos, et celles prises à l'extérieur, dans un lieu ouvert. Indépendamment, un détecteur de visages de face permet de compter les visages présents dans l'image et de les localiser. Enfin, deux groupes de deux catégories chacun sont distingués pour les photographies d'extérieur : les photographies de jour ou de nuit et les photographies de nature ou de ville. Nous décrivons dans la suite plus en détail ces catégories, ainsi que les algorithmes que nous avons appliqués à ces classifications de scènes.

La tâche 5 de la campagne ImagEVAL sur la classification de scènes a été définie en s'inspirant des travaux de cette thèse, et reprend donc une partie de cet arbre de classification. Nous reviendrons ultérieurement sur la campagne ImagEVAL et ses résultats dans la partie 4.2.3 sur l'évaluation.

4.1.1 Reconnaissance de cliparts

Il n'est pas facile de décrire réellement ce qu'on sous-entend par clipart et il n'en existe pas de traduction en français. Il semble de plus que la définition varie d'une personne à l'autre. Je vais donc définir ce qui, pour moi, est un clipart pour poser clairement les objectifs de cette classification.

Un clipart est une image qui ne se veut pas réaliste. Au contraire, il est plutôt proche d'un schéma. Visuellement, on ne distingue que très peu de couleurs, et les régions de l'image sont uniformes, ou alors présentent un dégradé régulier, dont on s'aperçoit très facilement qu'il a été réalisé par ordinateur contrairement aux rendus réalistes que l'on classifiera comme photographie. Les bords des objets sont également très nets, et les contours sont parfois surlignés en noir.

Les images qui sont en partie des photographies et des cliparts (par exemple des photographies auxquelles on a rajouté un cadre, ou du texte par traitement d'image) seront considérées comme des photographies par convention. Les figures 4.2 et 4.3 montrent des exemples d'images appartenant à chacune de ces deux classes.

Une méthode pour faire la distinction entre les cliparts et les photographies a été proposée par Rainer Lienhart et Alexander Hartmann [78]. Ils considèrent pour cela huit critères sur l'image qui sont potentiellement de bons indicateurs caractéristiques de

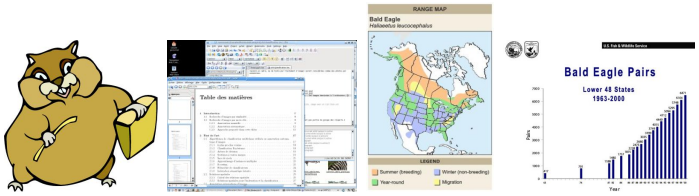


FIG. 4.2 – Exemples d'images considérées comme cliparts : les images dessinées par ordinateur, les copies d'écran, les graphiques.

FIG. 4.3 – Exemples d'images considérées comme ne faisant pas partie du groupe des cliparts : les photographies, notamment celles avec un fond quasi-uniforme ou celles avec un cadre uniforme rajouté par un logiciel, ainsi que les rendus réalistes.

la différence entre un clipart et une photographie : le nombre de couleurs, la taille des régions uniformes, la netteté des contours, etc. Ils utilisent ensuite un algorithme d'apprentissage AdaBoost sur ces critères pour déterminer ceux qui sont les plus pertinents, et n'en garder que quatre.

Avant cela, Michael J.Swain, Charles Frankel et Vassilis Athitsos [5, 131] avaient combiné de nombreux critères (appelés métriques) pour apprendre de manière automatique des arbres de décision à partir d'une base d'apprentissage. Ces arbres permettent de déterminer la classe à laquelle appartient une image à partir de ces métriques. Les différentes métriques utilisées sont : le nombre de couleurs, la surface occupée par la couleur dominante, la distance RVB entre un pixel et son voisin (en 4-connexité) le plus éloigné, la saturation des pixels, la corrélation avec l'histogramme moyenné sur toutes les photographies et celle avec l'histogramme moyenné sur les cliparts, l'histogramme des voisins les plus éloignés, le rapport entre la longueur de l'image et sa largeur, et la plus petite dimension de l'image. Les résultats qu'ils obtiennent, comparés avec les nôtres, sont donnés dans la section de résultats (section 4.2).

Les méthodes présentées dans la littérature pour effectuer l'identification de cliparts [5, 78, 131] m'ont paru bien trop complexes pour le but qu'elles devaient atteindre, notamment car elles font appel à des algorithmes d'apprentissage nécessitant de grandes bases d'images pour les entraîner, avec un temps de calcul relativement long, par rapport à ce que je propose par la suite. J'ai donc développé ma propre méthode en essayant de caractériser ce qui définit un clipart à partir de l'observation des histogrammes en niveaux de gris afin d'en déduire un bon critère discriminant. J'ai retenu seulement deux critères très simples : le nombre de couleurs et l'écart type des couleurs.

Méthode 1 : Nombre de couleurs

L'espace de couleur utilisé dans la suite est le modèle TSVal (Teinte Saturation Valeur). L'intérêt de cet espace est qu'il représente beaucoup mieux la façon dont nous percevons les couleurs que l'espace classique RVB.

Le nombre de couleurs est la première idée qui vient à l'esprit lorsque l'on cherche ce qui différencie un clipart d'une photographie : en 256 couleurs, un clipart n'a visuellement qu'une dizaine de couleurs alors que la plupart des photographies auront au moins 100 couleurs. Compter simplement le nombre de couleurs non nulles dans parmi les 256 couleurs ne fonctionne pas, notamment à cause des images compressées : sur une région uniforme où nous ne voyons qu'une couleur dominante, il y en a en fait une dizaine d'autres qui sont créées autour de cette couleur à cause de la compression. De nombreux cliparts dépassent donc également la centaine de couleurs. Ces couleurs ne sont a priori pas très présentes, et nous pouvons espérer qu'en seuillant l'histogramme – c'est-à-dire en mettant à zéro les couleurs dont la présence dans l'image est inférieure à un certain seuil – on éliminera ces effets de la compression.

Avec un seuil bien choisi, nous avons obtenu que les images de moins de 50 couleurs soient en général des cliparts, et celles au-dessus de 150 couleurs, des photographies. Cependant, d'une part, nous ne pouvons pas dire pour les images entre 50 et 150 couleurs s'il s'agit d'un clipart ou d'une photo, et d'autre part, il existe des photographies comme par exemple un oiseau sur fond de ciel bleu uni dont le nombre de couleurs est inférieur à 50 (cf. figure 4.4).

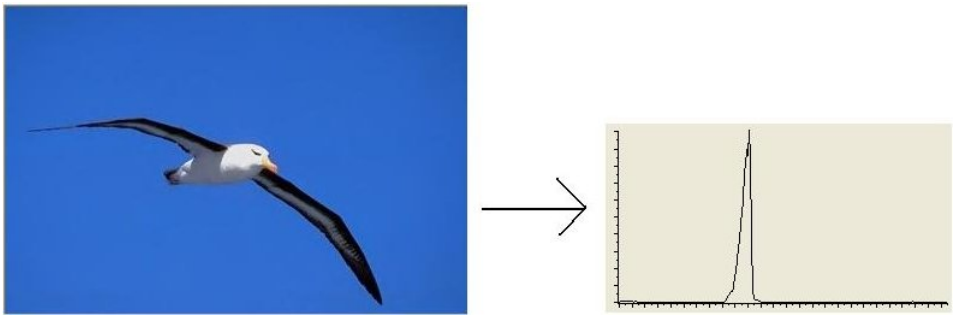


FIG. 4.4 – Une photographie dont le nombre de couleurs (après seuillage) est inférieur à 50. L'histogramme des niveaux de gris (avant seuillage) de la photographie de gauche est représenté à droite.

Nous avons donc cherché un autre critère discriminant, dont l'idée viendra en comparant l'histogramme de la figure 4.4 avec un histogramme de clipart, figure 4.5.

Un clipart possède des pics dans son histogramme alors que l'histogramme d'une photographie paraît plus continu.

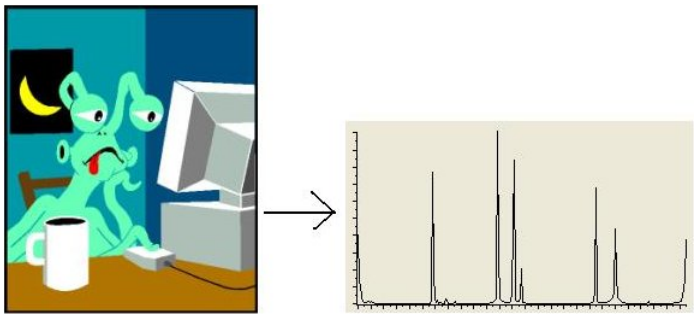


FIG. 4.5 – Un clipart et son histogramme des niveaux de gris.

Méthode 2 : Écart type des couleurs

La méthode que nous avons développée pour détecter la présence de pics est la suivante : d'abord, nous cherchons le maximum p de l'histogramme des niveaux de gris H :

$$p = argmax_{x\in[0,255]}(H(x)),$$

puis calculons un écart type σ^2 autour de ce maximum. À cause des problèmes de bords (en 0 et en 255) nous avons calculé en réalité deux écarts types : un écart type à droite du pic et un écart type à gauche. Lorsque les deux écarts types sont calculables, nous considérons la moyenne des deux :

$$\sigma^2 = \begin{cases} \sum_{x=p-5}^{p-1} r(x) & \text{si } p > 250 \\ \sum_{x=p+1}^{p+5} r(x) & \text{si } p < 5 \\ \frac{1}{2}(\sum_{x=p-5}^{p-1} r(x) + \sum_{x=p+1}^{p+5} r(x)) & \text{sinon} \end{cases}$$

avec

$$r(x) = \left(\frac{H(x)}{H(p)} * (x-p)\right)^2$$

Si l'écart type est faible, cela signifie que la couleur dominante se présente sous la forme d'un pic dans l'histogramme, et qu'il s'agit donc d'une couleur unie, sans variations. Dans le cas contraire, il s'agit d'un histogramme plutôt continu comme celui que possèdent les photographies. La détection d'un deuxième pic n'est pas envisageable, car nous souhaitons qu'une image uniforme d'une couleur puisse être détectée comme clipart. Dans une telle image, la détection d'un deuxième pic serait soit impossible, soit localisée sur du bruit dû à la compression de l'image.

Un problème se pose lorsque l'on utilise cette technique en considérant toute l'image comme un seul bloc. En effet, de nombreuses photographies ont par exemple un cadre qui a été rajouté par ordinateur, et qui sont donc d'une couleur pure. Par exemple, avec un cadre blanc, il y aurait un pic sur la couleur blanche dans l'histogramme, et la photographie serait alors classée dans la catégorie clipart.

Nous avons proposé la solution suivante pour résoudre ces cas :

1. diviser l'image en 16 parties égales : 4 colonnes et 4 lignes,
2. calculer l'écart type proposé précédemment sur chaque sous-image,
3. décider alors que l'écart type global pour l'image est le maximum des 16 écarts types ainsi calculés.

Afin de comprendre pourquoi nous considérons un découpage en 4×4 blocs, prenons le cas d'une image avec un cadre blanc. Avec une division en 2×2 blocs, une partie du cadre serait présente dans chaque sous-image, donc le problème ne serait pas résolu. Une division en 3×3 blocs aurait suffi, la partie centrale ne possédant pas de partie de cadre. Néanmoins, cela ne laisse qu'une seule région pour calculer l'écart type de l'image, les autres ayant un écart type trop faible pour être prises en compte à cause du cadre. Une division en 4×4 blocs est donc meilleure puisqu'elle donne 4 parties centrales. Pour des divisions plus grandes, les sous-images sont de plus en plus petites, et moins on a de pixels, moins les écarts types (comme tout opérateur statistique) sont significatifs, donc il vaut mieux se restreindre à 4.

Un clipart n'aura que des écarts types faibles sur chacune de ses sous-images, donc l'écart type global restera faible, alors qu'une photographie aura certains écarts types importants (voir tous, pour des photographies sans couleur unie rajoutée), donc l'écart type global sera lui aussi important. Cela se vérifie après quelques observations : les cliparts ont en général un écart type global inférieur à 5, tandis que pour les photographies, il est souvent supérieur à 40. Nous avons donc pris un seuil de discrimination de 15. Ces valeurs sont fournies à titre indicatif uniquement, et dépendent notamment de la transformation TSVal choisie, et d'autres conventions propres à chaque système de traitement d'images.

Malgré ce critère de décision, nous pouvons nous demander s'il faut garder ou non le premier critère sur le nombre de couleurs. Dans un premier temps, nous avions gardé ce critère : s'il y a plus de 150 couleurs (après seuillage), alors l'image est une photographie, et nous ignorons l'étape du calcul des écarts types. Cependant, au fur et à mesure que nous avons agrandi la base d'images cliparts, il est apparu un certain nombre de cliparts qui dépassaient ce seuil, et les résultats étaient devenus statistiquement meilleurs en utilisant uniquement le critère de l'écart type. Le choix de conserver ou non le premier critère dépend donc des images cliparts dont on dispose et plus particulièrement de la qualité de ces dernières, en termes de compression.

4.1.2 Noir et Blanc vs Couleur

Il y a plusieurs sortes d'images en noir et blanc :

- les images qui n'ont pas de couleurs, c'est-à-dire que les couleurs utilisées ne sont que des niveaux de gris.
- les images qui n'ont presque pas de couleurs : les couleurs sont tellement peu saturées, i.e. tellement sombres qu'elles nous paraissent grises : « la nuit tous les chats sont gris ». Pour ces images, l'ordinateur verra de vraies couleurs si l'on n'utilise que l'achromatisme théorique : $R = V = B \Leftrightarrow$ teinte non définie, mais $R \approx V \approx B \Leftrightarrow$ teinte définie.

– les images en noir et blanc qui ont été vieillies ou colorisées. Dans la classe des images vieillies, ce sont toutes les images en noir et blanc qui ont une coloration jaune ou sépia, quant aux images colorisées, il peut s'agir d'une colorisation voulue, et on pourra donc avoir diverses couleurs (bleu, vert, ...)

Pour chacune de ces trois catégories, il faut appliquer un critère spécifique, et nous étudierons successivement l'achromatisme, la saturation et la teinte. Remarquons que dans le cas des images vieillies, ainsi que dans le cas de certaines photographies achromatiques, savoir que l'image est en noir et blanc nous donne une information sémantique d'assez haut niveau : cela permet de dater la photographie. Certes, pas avec une précision absolue, mais déjà, pouvoir dire qu'une photographie date du début du 20ème siècle est non négligeable, et pourrait nous orienter notamment sur la nature des objets que nous sommes susceptibles de trouver dans cette image.

Achromatisme

Lors de la conversion d'une image de l'espace des couleurs RVB en TSVal, nous pouvons régler un seuil de tolérance d'achromatisme. Ce seuil s est tel que

$$\begin{cases} |R-V| < s \\ |R-B| < s \\ |V-B| < s \end{cases} \quad \text{i.e. } R \approx V \approx B \Leftrightarrow \text{pixel achromatique}$$

Dans ce cas, nous attribuons au pixel une teinte $T = -1$ pour signaler qu'il est achromatique, la saturation S ne peut pas être calculée et la valeur Val est alors égale au niveau de gris du pixel :

$$Val = \frac{R+V+B}{3}.$$

Le seuil s qui décide de l'achromatisme d'un pixel doit être choisi de telle manière que l'image de la figure 4.6 ait un achromatisme de 100%. En pratique et en supposant que R, V et B sont normalisés entre 0 et 1, nous avons déterminé expérimentalement qu'un seuil $s = 0,03$ donne des résultats satisfaisant.

Après avoir fixé s notamment pour que la figure 4.6 soit considérée comme complètement achromatique, le pourcentage de pixels servant à prendre la décision « image noir et blanc / couleur » doit être assez élevé pour qu'une image comme celle de la figure 4.7 qui ne possède que très peu de pixels chromatiques soit tout de même classée parmi les images en couleur. Le critère de décision que nous avons défini est le suivant : si plus de 99% des pixels de l'image sont achromatiques (au sens du seuil s fixé ci-dessus), alors l'image est en noir et blanc.

Ce critère s'applique aussi bien aux photographies qu'aux cliparts. En revanche, les deux critères que nous développons dans la suite, la faible saturation et la dominance de teinte, ne s'appliquent qu'aux photographies, car un clipart avec un rose faiblement saturé ne doit pas être reconnu comme étant noir et blanc, alors que pour une image avec une teinte rose et faiblement saturée, on aura tendance à dire qu'elle est en noir et blanc.

FIG. 4.6 – Un exemple de photographie achromatique.

FIG. 4.7 – Un clipart couleur à ne pas confondre avec un clipart noir et blanc.

Faible saturation

Pour commencer, voici, figure 4.8, un exemple de photographie dont les pixels ne sont pas achromatiques, mais que nous percevons en noir et blanc.

FIG. 4.8 – Un exemple de photographie faiblement saturée.

En réalité, cette image est en couleur, mais ses couleurs sont faiblement saturées, et nous donnent une sensation de niveaux de gris. Cela apparaît clairement sur son histogramme de saturation, figure 4.9.

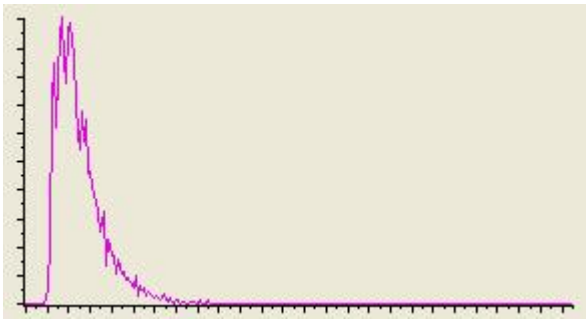


FIG. 4.9 – Histogramme de saturation de la photographie de la figure 4.8. En abscisse la valeur de la saturation (de 0 à 255), et en ordonnée, le nombre de pixels.

Nous nous apercevons que la saturation des pixels, qui est normalisée entre 0 et 255, ne dépasse pas 90 sur cet histogramme. En fait, quelques pixels ont une saturation supérieure à 90, mais ils ne sont pas assez nombreux pour être visibles sur l'histogramme.

Afin de savoir si une image est faiblement saturée, nous comptons simplement le pourcentage des pixels de l'image dont la saturation est inférieure à 100. Si ce pourcentage est supérieur à 95%, alors nous décidons que l'image est en noir et blanc. Ces seuils ont été déterminés empiriquement à partir d'observations sur des images de chaque classe.

Dominance de teinte

Une photographie jaunie telle que celle de la figure 4.10 devrait également être classée parmi les images monochromes.

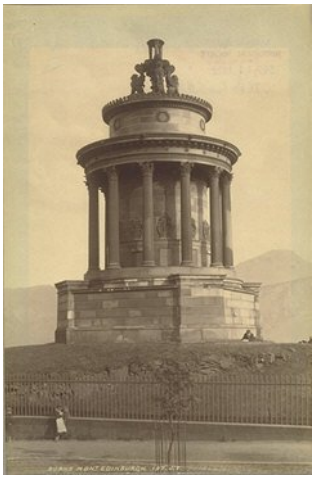


FIG. 4.10 – Photographie jaunie.

La saturation d'une telle image (figure 4.11) n'est pas forcément assez faible pour qu'elle soit classée comme noir et blanc seulement avec le critère de saturation développé ci-dessus.

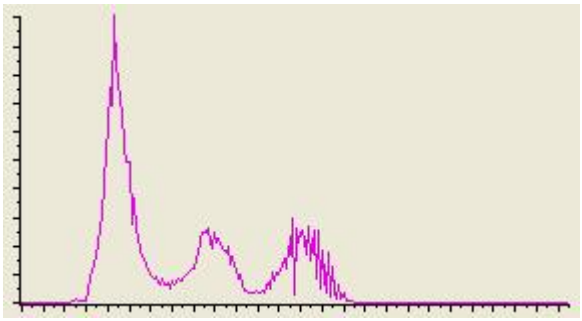


FIG. 4.11 – Histogramme de saturation de la figure 4.10. En abscisse la valeur de la saturation (de 0 à 255) et en ordonnée le nombre de pixels.

Cependant, sa teinte (figure 4.12) est assez caractéristique d'une image jaunie :

Pour savoir si l'image possède une teinte dominante, nous recherchons le maximum de la teinte, puis calculons le pourcentage de pixels de l'image dont la teinte est à une distance inférieure à 20 de la teinte du maximum. Si ce pourcentage est supérieur à 95%, alors l'image est considérée comme noire et blanche, et nous pouvons également donner

FIG. 4.12 – Histogramme de teinte de la figure 4.10. En abscisse la valeur de la teinte (de 0 à 255), et en ordonnée, le nombre de pixels.

la couleur de sa teinte qui correspond à la teinte du maximum.

Nous pouvons en effet facilement établir une correspondance entre les valeurs des pixels dans l'espace TSVal et le nom des couleurs, principalement d'après la teinte, même s'il faut aussi parfois se servir de la saturation et de la valeur. Les correspondances suivantes ont été élaborées en s'inspirant d'une part de l'article [26] et d'autre part en utilisant des observations personnelles.

teinte(T)	**Couleur**
$T = -1$	$achromatique$
$0 \leq T < 14$	$Rouge$
$14 \leq T < 29$	$Orange$
$29 \leq T < 45$	$Jaune$
$45 \leq T < 113$	$Vert$
$113 \leq T < 149$	$Cyan$
$149 \leq T < 205$	$Bleu$
$205 \leq T < 235$	$Violet$
$235 \leq T < 242$	$Rose, Magenta$
$242 \leq T \leq 255$	$Rouge$

Cela donne à peu près le découpage représenté sur la figure 4.13. Le découpage n'est pas parfait : les couleurs n'occupent pas des proportions égales, le rouge, le vert et le bleu sont les couleurs qui occupent le plus de place dans l'espace des teintes, alors que le orange, le jaune et le rose sont à peine représentés. De plus, les frontières entre les couleurs varient d'une personne à une autre.

Dans le cas d'une couleur achromatique, la seule information disponible est la valeur. Le nom de la couleur est alors déterminé de la manière suivante (pour une valeur normée entre 0 et 255) :

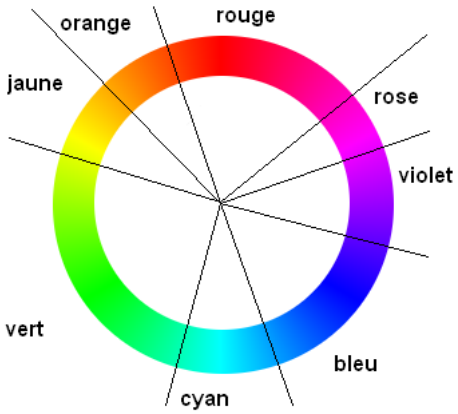


FIG. 4.13 – Représentation du nom de la couleur en fonction de la teinte.

Valeur(V)	**Couleur**
$0 \leq V < 82$	$Noir$
$82 \leq V < 179$	$Gris$
$179 \leq V < 255$	$Blanc$

On remarquera enfin qu'il manque la couleur marron dans le premier tableau. Cette couleur est plus difficile que les autres à nommer, car elle correspond en réalité plus ou moins à du orange foncé. Pour une teinte orange fixée, on trouvera le marron à la limite entre le orange et les couleurs achromatiques. Nous proposons de faire la distinction entre le orange et le marron de la manière suivante : pour une teinte comprise entre 14 et 29, nous définissons un orange « pur » par un couple saturation-valeur $(S, Val) = (255, 125)$ et un marron « pur » aux coordonnées $(S, V) = (184, 65)$. Pour un nouveau pixel (T, S, Val) avec T comprise entre 14 et 29, nous calculons la distance L_1 entre ce pixel et les deux couleurs de référence orange « pur » et marron « pur ». Nous affectons à ce nouveau pixel le nom de la couleur dont il est le plus proche. Cela se résume par la formule suivante :

critère(S,Val)	**Couleur**
$\|184 - S\| + \|65 - Val\| < \|255 - S\| + \|125 - Val\|$	$Marron$
$\|184 - S\| + \|65 - Val\| \geq \|255 - S\| + \|125 - Val\|$	$Orange$

De même que le orange foncé nous apparaît marron, le jaune foncé (qui correspond à une teinte jaune, avec une valeur faible) nous apparaît vert. Nous réglons ce problème de la manière suivante :

- si la teinte est jaune, et que $Val < 80$, alors la couleur est verte,
- si la teinte est jaune, et que $Val \geq 80$, alors la couleur est jaune.

Nous pouvons désormais nommer les différentes couleurs, en se rappelant toutefois qu'aux frontières entre les couleurs, les noms ne sont définis que de manière subjective :

cela dépend de celui qui nomme la couleur, et de l'environnement dans lequel la couleur est perçue – environnement qui tient compte des variations d'illumination, des couleurs accolées à la couleur observée, ou tout autre artefact de ce genre.

Pour en revenir à la classification des images en noir et blanc, afin que des photographies qui seraient colorisées en jaune vif ne soient pas comptées parmi les photographies en noir et blanc, nous rajoutons au critère sur la teinte un critère sur la saturation : plus de 50% des pixels de l'image doivent avoir une saturation faible au sens du critère explicité dans la section précédente.

4.1.3 Intérieur vs Extérieur

A présent, nous allons nous intéresser aux photographies en couleur pour essayer de les trier suivant leur contenu sémantique, selon qu'il s'agisse d'une photographie prise à l'intérieur d'un bâtiment, ou à l'extérieur.

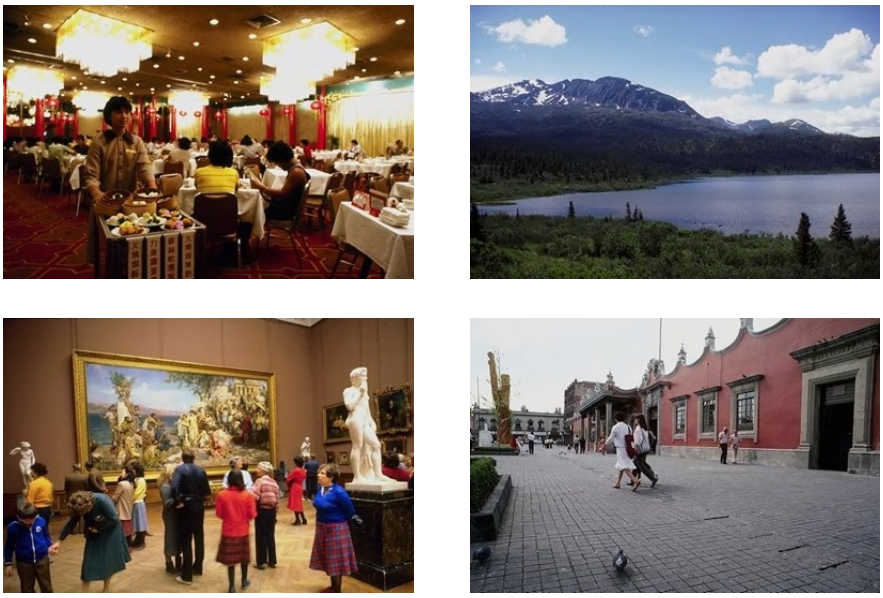


FIG. 4.14 – Exemples de photographies d'intérieur (à gauche) et de photographies d'extérieur (à droite) extraites de la base Corel.

Les premières expériences que nous avons menées pour la classification « intérieur / extérieur » utilisaient les descripteurs LEP et RVB-64, et la classification était faite grâce à un algorithme de k-plus proche voisins. Il est très vite apparu que l'utilisation d'un SVM avec un noyau gaussien améliorait les résultats : avec le même descripteur, les résultats passaient de 86,0% de bonnes détections à 88,7%. En outre, le SVM a l'avantage de nous décharger du problème difficile de choisir une distance entre deux images, c'est-à-dire entre deux vecteurs de descripteurs. La combinaison de plusieurs descripteurs devient alors beaucoup plus simple : il suffit de donner en entrée du SVM

une concaténation des vecteurs de descripteurs, alors qu'avec les k-plus proches voisins, il faudrait trouver une pondération entre les distances propres à chaque descripteur pour définir une distance générale. D'autre part, chaque descripteur ajouté augmentait le pourcentage de bonnes reconnaissances. Nous avons donc choisi pour la classification de combiner les six descripteurs suivants pour entraîner un SVM : LEP, RVB-64, Tsv-125, projection, Bic et Gabor. Ces descripteurs ont été décrits précédemment dans le chapitre 3.

4.1.4 Localisation de visages

Pour l'apprentissage des visages, nous avons recours au détecteur d'objets de Viola et Jones [144], avec les améliorations proposées par Lienhart [79]. Ces auteurs ont déjà montré les bons résultats obtenus avec cette méthode pour la détection de visages.

Ce détecteur fonctionne pour des images en niveaux de gris. Le principe est d'abord d'utiliser une fenêtre glissante de taille 24×24 pour détecter les visages dans l'image. On fait varier l'échelle de l'image afin de détecter des visages de différentes tailles. Sur chaque fenêtre de taille 24×24, les filtres très simples de type Haar représentés sur la figure 4.15 sont appliqués. Ces filtres calculent la somme des pixels dans la partie blanche à laquelle il faut soustraire la somme des pixels dans la partie noire. Toutes les valeurs possibles de largeur et de hauteur pour ces filtres sont exploitées, ce qui résulte selon [79] en 117 941 filtres possible. À partir d'une base d'apprentissage contenant des images de visages normalisées et des images n'appartenant pas à cette classe, redimensionnées en 24×24, la méthode Adaboost permet de retenir un nombre restreint de ces filtres dont la combinaison donne de bonnes performances.

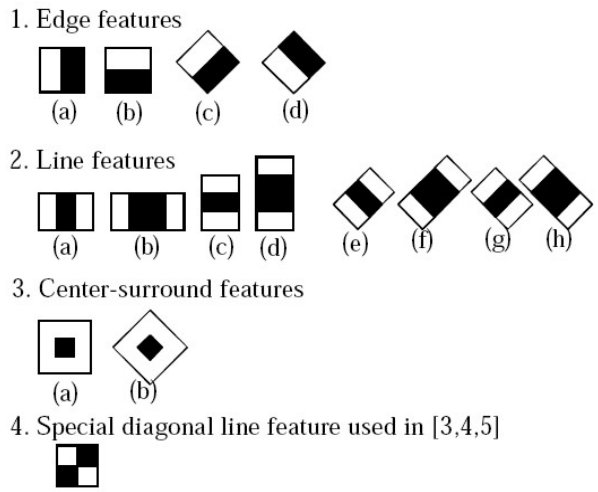


FIG. 4.15 – Descripteurs simples de type Haar introduits par Lienhart [79] pour la détection d'objets.

Viola et Jones [144] ont entraîné une cascade de classifieurs constituée de plu-

sieurs étapes dont chaque étape est une telle combinaison de filtres entraînée par l'algorithme Adaboost pour classer correctement 99,9% des visages et rejeter 50% des autres images. Le rôle des étapes suivantes est alors d'essayer de rejeter correctement les fausses détections de l'étape précédente. Le classifieur de l'étape 2 est par exemple entraîné uniquement avec les images que le classifieur de l'étape 1 a classées comme visages (théoriquement constituée d'environ 99,9% des visages et 50% des autres images de la base initiale ayant servi à entraîner l'étape 1).

L'apprentissage d'une telle cascade de classifieurs est très long, durant une semaine dans notre cas, mais la détection est à la fois de très bonne qualité et rapide : de l'ordre de cinq secondes pour des images de taille 400×400. Notre base d'apprentissage est constituée de 2000 images de visages de la base FERET [111]. Afin d'améliorer la robustesse de la reconnaissance, 4000 images supplémentaires ont été générées à partir de ces 2000 en considérant des rotations de 20 degrés dans le sens horaire et 20 degrés dans le sens anti-horaire. Les quelques 10 000 images de la classe négative proviennent de diverses images de paysages ne contenant pas de visages, et de fausses détections à corriger. Toutes les images sont redimensionnées en 20×20 pixels.

Onze étapes de la cascade ont été apprises, avec pour chaque étape la contrainte de classer correctement 99,9% des visages, et de rejeter 70% des images qui ne sont pas des visages. Le détecteur final a donc théoriquement un taux de bonnes détections de $0,999^{11} = 98,9\%$ et un taux de fausse alerte (un visage est détecté à un endroit où il n'y en a pas) de $0,30^{11} = 1,8 * 10^{-6} = 0,00018\%$, c'est-à-dire une fenêtre 20×20 sur un million. Sur une image 400×400 (pour la détection de visages, les grandes images sont réduites pour que leur surface vale 400×400, en conservant le rapport largeur / hauteur, afin d'améliorer les temps de calcul), en considérant des fenêtres glissantes de taille 20×20, de pixel en pixel avec des réductions d'échelles de $1,1$ après chaque passe, il y a environ 760 000 fenêtres à analyser, soit en moyenne un taux théorique de 1,37 erreur par image.

Ce taux est en pratique plus faible. En effet, lorsqu'un visage est présent dans une image, plusieurs visages sont en général détectés à des positions et des échelles très proches autour de ce visage. Nous requérons d'avoir 5 détections « proches » autour d'une même position pour décider de conserver le visage détecté et rejetons la détection dans le cas contraire. Il faut donc qu'il y ait au moins 5 erreurs au même endroit dans une image pour résulter en une fausse détection. Nous définissons par « proches » un ensemble de fenêtres dont les centres sont tous dans une même zone de taille 10×10 et dont la variation maximale d'échelle entre la plus petite et la plus grande de ces fenêtres est inférieure à 1.5. La moyenne des centres et des tailles de ces fenêtres est alors considérée comme la position du visage détecté.

Nous n'évaluerons pas cet algorithme dans les résultats. En effet, nous n'avons pas introduit de modification sensible par rapport à celui développé par Lienhart [79] et nous pourrions donc espérer obtenir des résultats similaires aux leurs avec une base d'apprentissage comparable, que nous n'avons pas eu le temps de construire.

Nous présentons à titre d'exemple une détection de plusieurs visages obtenue avec notre algorithme sur la figure 4.16.

FIG. 4.16 – Détection automatique de plusieurs visages dans une image. Image originale : © zesmerelda sur Flickr, licence : Creatives Commons by-sa.

4.1.5 Autres classifications

Pour les autres classifications, la même approche que celle proposée pour faire la différence entre les scènes d'intérieur et les scènes d'extérieur a été appliquée : plusieurs descripteurs sont mis bout à bout et utilisés pour entraîner un séparateur à vaste marge. Pour la classification « carte / clipart / photographie / peinture », il a été observé qu'entraîner un (ou des) SVM pour ces quatre classes était moins efficace que d'utiliser la méthode précédemment décrite pour détecter les cliparts, puis d'entraîner un SVM pour classifier « carte / photographie / peinture » sur les images reconnues comme n'étant pas des cliparts.

Plus de détails sont donnés dans la section suivante, notamment dans la partie concernant l'évaluation sur la campagne ImagEVAL.

4.2 Résultats

Nous évaluons dans cette partie les différents algorithmes de classification décrits ci-dessus, à l'exception de la détection de visages.

4.2.1 Classification de cliparts

Les résultats pour la classification de cliparts sont au moins aussi bons que ceux obtenus dans Lienhart et al. [78]. Cependant, la comparaison n'est pas vraiment possible, puisque les bases d'images qu'ils ont utilisées ne sont pas disponibles en accès libre. A titre indicatif, leurs résultats sont de 98,97% de photographies correctement classées et de 91,92% pour les cliparts. Les résultats de Athitsos et al. [5] varient de 91% à 94% suivant que l'image soit au format JPEG ou GIF.

Nous obtenons les résultats suivants : sur la base Corel qui est composée de 11252 photographies, seules 25 images sont mal classifiées (99,78% de réussite), et sur une base de 5402 cliparts de provenances diverses, dont Internet, nous dénombrons 377 erreurs (93,02% sont bien reconnus).

Les images qui posent problème sont de quatre types (cf. figure 4.17) : deux types pour les cliparts, et deux types pour les photographies.

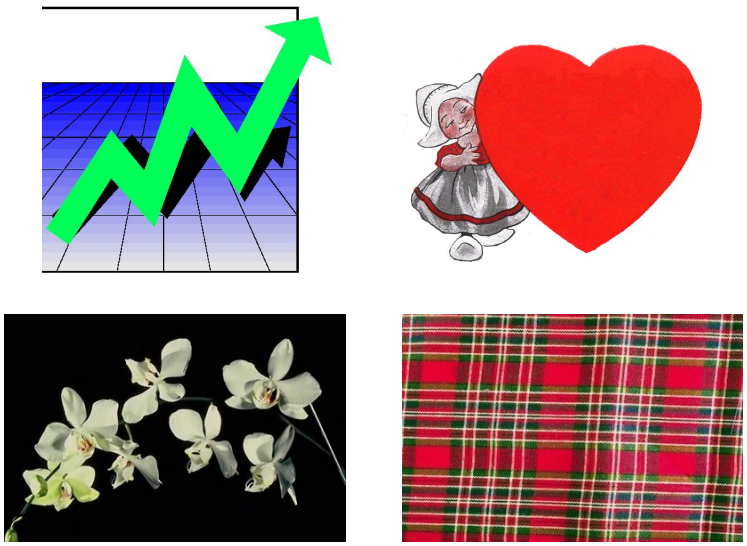


FIG. 4.17 – Exemple d'images pour lesquelles l'algorithme de classification automatique Clipart vs Photo n'a pas pris la bonne décision. Les deux premières images en haut sont des cliparts reconnus comme étant des photographies : la première à cause de son dégradé, la deuxième du fait de sa mauvaise qualité. Les deux images du bas sont des photographies affectées à la catégorie clipart. La première a un fond noir uniforme présent dans presque toute l'image, la seconde a une couleur rouge presque uniforme et prédominante.

Parmi la base de cliparts, ceux avec un dégradé auront du mal à être reconnus. En effet, un dégradé, aussi régulier soit-il – et s'il est régulier, on voit assez facilement qu'il s'agit d'un dégradé créé par un logiciel de traitement d'image – ne se présente pas dans l'histogramme comme un pic, mais au contraire comme un plateau. L'écart type sera donc important pour ce type d'images. Les autres cliparts pour lesquels nous ne

détectons pas de pics dans l'histogramme sont les cliparts de très mauvaise qualité. Cette mauvaise qualité se retrouve assez souvent dans les images fortement compressées qui se rencontrent assez souvent sur Internet pour des raisons de taille.

Parmi les photographies, c'est la présence d'une ou de plusieurs couleurs uniformes dans l'image qui constitue un leurre pour l'algorithme. Souvent, de telles images représentent un objet de faible surface relativement à la taille de l'image, sur un fond uniforme qui occupe le reste de l'image. Une telle uniformité est souvent la preuve que l'image a été traitée, mais peut aussi bien être d'origine naturelle. Dans ce cas, il arrive que même après un découpage de l'image en 4×4 blocs, le fond reste présent en quantité suffisante sur les 16 sous-images, et l'écart global calculé ne tiendra compte que de ce fond uniforme. Plus rarement, la couleur uniforme est présente sur l'objet lui-même. Ce n'est pas naturel, car les jeux d'ombres et de lumières devraient l'empêcher d'apparaître, mais cela peut arriver si cette couleur sature l'appareil photographique numérique (ou le scanner), ce qui est apparemment le cas dans la dernière image de la figure 4.17.

Le taux de bonnes reconnaissances obtenu est comparable à ce qui se fait actuellement dans la littérature. Nous pouvons donc nous appuyer sur les résultats de classification automatique pour faire d'autres classifications comme celle qui vient ensuite : déterminer si une image est en noir et blanc, ou en couleur.

Nous avons comparé également la méthode développée ici avec un apprentissage par un séparateur à vaste marge, en apprenant à partir de caractéristiques de textures et de couleurs (les descripteurs nommés LEP et Bic dans le chapitre 3) et cela donne de moins bons résultats avec un temps d'apprentissage plus long.

4.2.2 Classification noir et blanc / couleur

Nous n'avons pas trouvé d'autres articles traitant de ce sujet, et ne pouvons donc pas comparer nos résultats à ceux de la littérature. Nos résultats sont les suivants : sur notre base d'images de cliparts, il n'y a que 2 images en noir et blanc et seules ces deux images sont classées comme telles, ce qui fait 100% de bonne classification. Sur une base contenant des photographies en noir et blanc, qui sont soit achromatiques, soit faiblement saturées, soit teintées (et dont sont extraites les photographies 4.8 et 4.10), 93,78% des images sont correctement considérées comme étant en noir et blanc. Il est cependant difficile d'évaluer un tel système : la différence entre une image noire et blanche colorisée et une photographie en couleur – par exemple une photographie sombre, prise dans un lieu avec très peu d'illumination, et sans flash – est parfois très subjective.

4.2.3 Campagne ImagEVAL

ImagEVAL[1] est une campagne d'évaluation axée sur les technologies de traitement d'images, notamment celles utilisées pour la recherche d'images par le contenu et l'annotation ou classification d'images. La première édition d'ImagEVAL a eu lieu en 2006. Elle était composée de cinq tâches :

[1] http ://www.imageval.org/

1. Reconnaissance d'images transformées : retrouver à partir d'une image source toutes les images d'une base qui sont issues d'une transformation de cette image source. Les transformations sont par exemple la rotation, la translation, le passage en niveaux de gris, le changement de la qualité JPEG, l'insertion dans une autre image, etc.
2. Recherche combinant le texte et l'image : à partir d'une question textuelle et de quelques images d'exemple, le but est de trouver les images répondant à la requête parmi un corpus de pages web contenant du texte et des images, notamment des pages de Wikipédia.
3. Détection de zones de texte dans une image : le but est de localiser, dans des images données et le plus précisément possible, la boîte englobante délimitant une zone de cette image contenant du texte. Il n'est pas demandé dans cette tâche de reconnaître le texte, mais simplement de le localiser.
4. Détection d'objets : à partir de 743 images d'apprentissage, l'objectif est d'être capable de reconnaître dix objets dans une base de données de 14000 images. Ces dix objets sont : arbre, minaret, tour Eiffel, vache, drapeau américain, voiture, véhicule blindé, lunettes de soleil, panneau routier, avion. Une difficulté supplémentaire est que parmi les 14000 images de tests, certaines ne contiennent aucun de ces dix objets, alors que d'autres en contiennent plusieurs.
5. Reconnaissance d'attributs : la tâche consiste à classer une scène parmi 10 catégories. Ces catégories sont : photographie en noir et blanc, photographie en couleur, photographie en noir et blanc colorisée, reproduction artistique, photographie d'intérieur, photographie d'extérieur, jour, nuit, nature, urbain. Ces catégories sont organisées dans un arbre sémantique représenté sur la figure 4.18. Pour cette tâche, il y a 5474 images d'apprentissage et 23572 images de test.

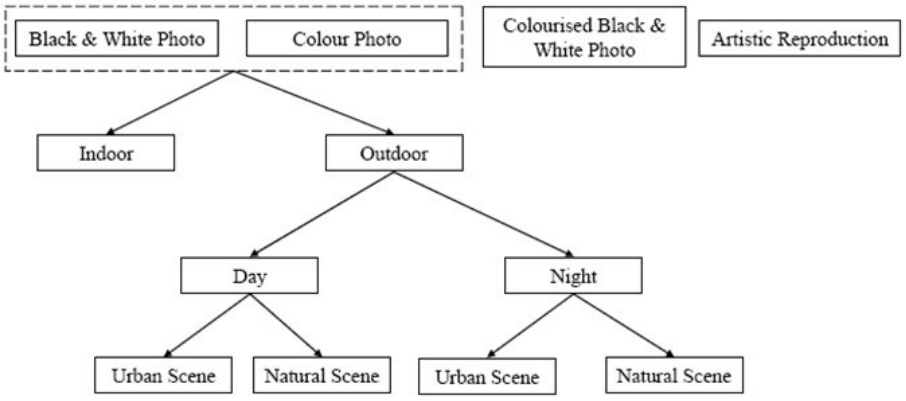


FIG. 4.18 – Arbre sémantique de classification utilisé dans la campagne imagEVAL.

Dans le cadre de cette thèse, nous nous sommes plus particulièrement occupés de la tâche 5. Cette tâche se compose de 13 requêtes. Pour chacune des requêtes, le système

doit retourner les 5000 images les plus probables répondant à cette requête parmi les 23572 images de test, en classant ces images par ordre de probabilité. Le système est ensuite évalué par rapport à la vérité terrain en calculant la MAP (*mean average precision*) :

$$MAP = \frac{\sum_{n=1}^{N}(P(n) \times rel(n))}{N}$$

où N est le nombre d'images pertinentes pour la requête considérée (majoré à 5000 si le nombre d'images pertinentes dépasse cette valeur), $rel()$ une fonction binaire qui vaut 1 si le document au rang n est pertinent et 0 sinon, et $P(r)$ est la précision au rang r, c'est-à-dire le taux de réponses pertinentes parmi les r premières réponses retournées par le système.

Parmi les requêtes à traiter dans la tâche 5 d'ImagEVAL, certaines requêtes demandent de trouver des images appartenant à une seule catégorie, telle que des images de représentations artistiques (requête 1), ou des images en noir et blanc colorisées (requête 2), et d'autres requêtes font une intersection entre plusieurs classes, pouvant aller jusqu'à quatre classes, comme par exemple trouver des images en couleur d'une scène de nature prise à l'extérieur et de nuit (requête 13).

L'expérience soumise à ImagEVAL utilise la même stratégie pour toutes les classifications : pour chaque image, nous calculons plusieurs histogrammes issus des descripteurs décrits dans le chapitre 3. Nous utilisons quatre de ces descripteurs : Tsv, Bic, LEP et Projection. Les histogrammes correspondant sont mis bout à bout formant un vecteur de 1202 dimensions pour chaque image qui est utilisé pour entraîner des séparateurs à vaste marge probabilistes (SVM). Probabiliste signifie qu'une probabilité est assignée à chaque classe lors de la classification, de telle manière que la somme des probabilités sur toutes les classes soit égale à 1.

Quatre SVMs ont été entraînés : un SVM à quatre classes pour la classification « photographie en noir et blanc / photographie en couleur / photographie en noir et blanc colorisée / reproduction artistique », puis trois SVMs binaires pour les classifications « photographie d'intérieur / photographie d'extérieur », « jour / nuit », « nature / urbain ». Le SVM multiclasse à quatre classes utilise la méthodologie un-contre-un, c'est-à-dire qu'il entraîne un SVM binaire pour chaque couple de classe (donc au total six SVMs), puis combine les résultats de toutes ces classifications pour obtenir une classification en quatre classes. La librairie libSVM [21] nous a permis de conduire nos tests, et nous avons choisi de prendre toujours un noyau gaussien dont les paramètres ont été déterminés par validation croisée sur la base des images d'apprentissage.

Les probabilités retournées par les SVMs ont servi à trier les réponses. Dans le cas de requêtes où plusieurs classes sont combinées, le minimum de toutes les probabilités a été utilisé comme probabilité finale pour l'image. Par exemple, pour la requête 13 :

$$P(13|I) = \min[P(color|I), P(outdoor|I), P(night|I), P(natural|I)]$$

Nous sommes arrivés en deuxième position parmi 6 candidats dans les résultats officiels d'ImagEVAL. Nos résultats par rapport aux deux autres meilleurs participants sont affichés dans le tableau 4.1.

	Requête	**MAP**		
		CEA LIST	**INRIA**	**ENSEA ETIS**
1	Art	0,7651	0,9275	0,7055
2	NB colorisé	0,5154	0,8682	0,4715
3	NB, intérieur	0,8274	0,8839	0,7353
4	NB, extérieur	0,7695	0,841	0,6504
5	couleur, intérieur	0,6602	0,7546	0,5589
6	couleur, extérieur	0,5203	0,5691	0,5112
7	NB, extérieur, nuit	0,0719	0,0819	0,0954
8	NB, extérieur, jour, urbain	0,7225	0,811	0,6434
9	NB, extérieur, jour, naturel	0,4247	0,6109	0,2036
10	couleur, extérieur, jour, urbain	0,6557	0,7764	0,5469
11	couleur, extérieur, jour, naturel	0,7956	0,8849	0,6639
12	couleur, extérieur, nuit, urbain	0,6125	0,6499	0,5228
13	couleur, extérieur, nuit, naturel	0,1612	0,1601	0,077
	Moyenne	0,5771	0,6784	0,4912

TAB. 4.1 – Résultats des participants sur la tâche 5 de la campagne ImagEVAL d'après le site officiel (NB = noir et blanc). Seuls les résultats des trois meilleures équipes sont rendus public. Une réponse parfaite aurait une MAP de 1.

L'équipe de l'INRIA a utilisé le même type d'approche que la nôtre, avec des descripteurs globaux différents, plus longs à calculer que les nôtres, mais apparemment plus efficaces pour la classification [101].

Nous avons par la suite mené quelques expériences supplémentaires pour analyser nos résultats et essayer de les améliorer. Tout d'abord, dans le cas de requêtes qui correspondent à des intersections de plusieurs classes, nous avons arbitrairement choisi ci-dessus d'utiliser le min de chacune des probabilités d'appartenance de l'image à chacune des classes pour déterminer la probabilité qu'a l'image d'appartenir à l'intersection de ces classes. Cela n'a pas vraiment de fondement théorique, et nous pouvons comparer cela avec la multiplication des différentes probabilités de chaque classe, ce qui revient à considérer, en termes probabilistes, que toutes les classes sont indépendantes. Pour la requête 13, la probabilité calculée devient alors :

$$P(13|I) = P(color|I) * P(outdoor|I) * P(night|I) * P(natural|I)$$

Les résultats affichés dans le tableau 4.2 montrent que cela améliore légèrement la classification. Seule la requête 11 a une perte de MAP (-0,0033), alors que le meilleur gain est obtenu pour la requête 12 (+0,0238).

Nous avons ensuite effectué deux autres études sur la possibilité d'améliorer nos résultats. La première étude consiste à étudier s'il est préférable d'entraîner un SVM multiclasse, comme nous l'avons fait précédemment, pour faire la distinction entre « photographie en noir et blanc, photographie en couleur, photographie en noir et blanc colorisée, reproduction artistique », ou s'il vaut mieux plutôt entraîner un SVM binaire

	Requête	min (MAP)	mult (MAP)	gain
1	Art	0,7651	0,7651	
2	NB colorisé	0,5154	0,5154	
3	NB, intérieur	0,8274	0,8489	+0,0215
4	NB, extérieur	0,7695	0,7734	+0,0039
5	couleur, intérieur	0,6602	0,6677	+0,0075
6	couleur, extérieur	0,5203	0,5212	+0,0010
7	NB, extérieur, nuit	0,0719	0,0885	+0,0166
8	NB, extérieur, jour, urbain	0,7225	0,7256	+0,0031
9	NB, extérieur, jour, naturel	0,4247	0,4402	+0,0155
10	couleur, extérieur, jour, urbain	0,6557	0,6668	+0,0111
11	couleur, extérieur, jour, naturel	0,7956	0,7923	-0,0033
12	couleur, extérieur, nuit, urbain	0,6125	0,6363	+0,0238
13	couleur, extérieur, nuit, naturel	0,1612	0,1839	+0,0227
	Moyenne	0,5771	0,5868	+0,0095

TAB. 4.2 – Comparaison entre l'utilisation du minimum ou de la multiplication des probabilités individuelles de chaque classe pour intersecter plusieurs classes.

pour chacune de ces quatre classes. Dans le premier cas, le SVM multiclasse affecte une probabilité pour chacune de quatre classes de telle manière que la somme de ces probabilités vale 1, et donc la probabilité d'appartenance à une classe est dépendante des autres. Dans le second cas, les SVMs binaires font la distinction entre par exemple « reproduction artistique » et « autres », où cette classe « autres » est définie comme la réunion des trois autres classes.

La deuxième étude consiste à évaluer si l'ajout d'un cinquième descripteur permet encore d'améliorer les résultats de la classification. Nous avons en effet fait quelques expériences qui nous ont montré que parmi les quatre descripteurs que nous avons combinés ci-dessus, le retrait de n'importe lequel de ces quatre descripteurs détériore les performances en classification, et nous pouvons donc légitimement nous demander si l'ajout d'un nouveau descripteur peut améliorer encore les performances. Nous avons choisi d'ajouter le descripteur *RVB-64-9* décrit dans la section 3.1.3, qui est un descripteur principalement fondé sur la couleur, contenant quelques informations spatiales. Les résultats de ces deux expériences sont donnés dans le tableau 4.3.

Il ressort de ces deux études que, dans la plupart des cas, l'utilisation du cinquième descripteur améliore sensiblement les performances en classification. Seule la requête 13 a une MAP meilleure avec seulement quatre descripteurs, mais il est difficile de dire pourquoi, car cette requête consiste en l'intersection de quatre classes, et les autres requêtes faisant intervenir ces quatre classes obtiennent de meilleurs résultats avec le cinquième descripteur.

Pour ce qui est de la différence entre l'utilisation de 4 SVMs binaires au lieu d'un SVM multiclasse, les résultats sont légèrement meilleurs en moyenne avec l'utilisation de SVMs binaires, mais avec très peu de différence que ce soit avec quatre ou cinq descripteurs,

Requête	**MAP**			
	4 descripteurs		5 descripteurs	
	SVM multi	4 SVMs bin	SVM multi	4 SVMs bin
1	0,7651	0,8532	0,8012	**0,8590**
2	0,5154	0,5587	0,6143	**0,6278**
3	0,8489	0,8443	0,8726	**0,8761**
4	0,7734	0,7744	**0,8073**	0,7992
5	0,6677	0,6587	0,7330	**0,7350**
6	0,5212	0,5302	0,5553	**0,5566**
7	0,0885	0,1145	0,1204	**0,1498**
8	0,7256	0,7253	**0,7821**	0,7786
9	0,4402	0,4373	**0,4553**	0,4431
10	0,6668	0,6544	**0,7389**	0,7329
11	0,7923	0,8166	0,8045	**0,8248**
12	0,6363	0,6305	**0,6637**	0,6612
13	**0,1839**	0,1443	0,1305	0,1296
Moyenne	0,5866	0,5956	0,6215	**0,6287**

TAB. 4.3 – Expérience sur la possibilité d'améliorer les résultats de deux façons : en utilisant des SVMs binaires au lieu du SVM multiclasse, et en introduisant un cinquième descripteur.

sauf pour la requête 1 (représentations artistiques), où les images appartenant à cette classe sont apparemment mieux identifiées avec un SVM binaire.

4.3 Conclusion

Nous avons développé et évalué dans cette partie des classifieurs donnant des informations sur la scène représentée par l'image. Parmi ces classifieurs, certains sont spécifiques à la tâche considérée. Notamment, notre algorithme de détection de cliparts et celui de classification « noir et blanc, noir et blanc colorisé, couleur » sont fortement liés à la nature de ces images. L'algorithme de détection de visages permet de déduire si la scène est sans visage, est un portrait, ou une photographie de groupe. L'article à l'origine de cet algorithme affirme qu'il peut être utilisé pour divers objets, mais il n'a été utilisé dans la littérature principalement que pour la détection de visages. Il est en effet assez spécifique dans le sens où l'objet que l'on cherche à détecter doit avoir très peu de variations visuellement. Il faut par exemple entraîner deux classifieurs si l'on souhaite obtenir de bons taux de classification à la fois pour les visages de face et les visages de trois-quart. C'est à cause de cette spécificité que nous avons classé cet algorithme dans la partie sur la description de scène, et non dans la reconnaissance d'objets.

Le dernier classifieur utilisé, qui combine plusieurs descripteurs pour entraîner un SVM, est beaucoup moins spécifique et peut facilement être appliqué à la détection de

plusieurs types de scènes comme nous l'avons montré ci-dessus. Ainsi, l'arbre de classification que nous avons établi ne montre que des grandes classes d'images très générales, mais il est possible d'introduire d'autres classes de scènes en fonctions des besoins et des applications visées. Nous montrerons notamment ultérieurement une application où les six scènes suivantes seront apprises avec ce même algorithme : savane, champ, région polaire, désert, forêt et milieu aquatique.

Enfin, outre l'annotation en soit que procure la connaissance de la scène de l'image, nous verrons dans le chapitre 7 comment elles peuvent également intervenir pour désambiguïser la reconnaissance d'objets.

Chapitre 5

Création automatique de bases d'images pour l'apprentissage

Training is everything. The peach was once a bitter almond ; cauliflower is nothing but cabbage with a college education.
Mark Twain, Pudd'nhead Wilson

La reconnaissance de catégories d'objets est un problème très difficile en traitement d'images. Le paradigme actuel [6, 48] consiste à constituer manuellement un grand ensemble d'images d'apprentissage correspondant aux objets à reconnaître. Un classifieur est alors entraîné sur ces images afin de permettre la reconnaissance sur d'autres images. Le recours à un procédé manuel pour constituer ces bases se justifie par le fait que l'apprentissage est une tâche difficile mais qui n'a besoin d'être faite qu'une fois et il est donc envisageable de passer du temps à construire les bases manuellement. Cependant, une limite aux progrès courants est la nécessité d'obtenir des bases suffisamment larges de tous les objets que l'on souhaiterait reconnaître. La taille de ces bases augmente en effet proportionnellement avec le nombre de concepts que l'on souhaite reconnaître d'une part, et avec le nombre d'images exemples pour chaque concept d'autre part. Comme nous l'avons vu dans la section 2.3.6, les plus grandes bases d'images construites manuellement contiennent de 10 000 à 60 000 images avec un nombre de concepts allant de 100 à 600. Il s'agit là d'images annotées manuellement, mais non segmentées : la localisation du ou des objets identifiés dans l'image n'est pas disponible. Les nombres d'images et de concepts pour lesquels il existe des bases d'images segmentées manuellement sont encore plus petits. Nous avons par exemple constitué, avec le réseau d'excellence MUSCLE, une base d'environ 15 000 images d'animaux annotée manuellement dont « seulement » 1 300 images de 14 animaux ont été segmentées manuellement. Cela a nécessité le travail d'une

dizaine de personnes et a été coûteux en temps : la segmentation d'une image dure en moyenne 2 minutes.

Internet, en revanche, contient beaucoup d'images qu'il pourrait être intéressant d'exploiter, notamment pour cette tâche d'apprentissage d'objets. Il est actuellement possible d'accéder et de rechercher les images sur le web à travers des moteurs de recherche où l'utilisateur envoie une requête constituée d'un ou de plusieurs mots-clés. Les images sont annotées automatiquement par ces moteurs de recherche en fonction de leur titre, et en fonction du texte des pages webs qui contiennent cette image. Les moteurs de recherche d'images sur Internet donnent ainsi accès à plusieurs milliards d'images. Cependant, la liste brute des images retournées par ces moteurs est difficilement utilisable telle quelle pour l'apprentissage d'objets, car elle contient trop d'images qui soit ne correspondent pas à la requête, soit sont de trop mauvaise qualité pour l'apprentissage, par exemple si l'objet est trop petit dans l'image pour être utile. Une rapide évaluation de quatre moteurs de recherche d'images sur le web (Google, Yahoo !, Ask et Exalead) nous a permis de constater qu'en moyenne, pour des requêtes simples constituées d'un seul mot clé portant sur des animaux ou des objets, environ 50% des images obtenues sont du bruit : elles ne correspondent pas à la requête effectuée. Cela s'explique notamment par le fait que l'indexation et la recherche de ces images utilisent uniquement le texte et ne tiennent pas compte du contenu des images. Fergus et al. [49] avaient également mesuré ce phénomène, observant un taux d'images non-pertinentes pouvant aller de 18% à 71% pour certaines requêtes simples (e.g. *bottle*, *cars*, *leopards*, *mugs*).

Afin d'améliorer la qualité de la recherche d'images sur Internet, des chercheurs de l'université Carnegie Mellon ont développé un logiciel d'annotation manuelle d'images sous forme de jeu : ESP game [147]. Ce jeu propose une approche collaborative : afin d'améliorer la pertinence et l'objectivité par rapport aux annotations classiques, deux joueurs sélectionnés aléatoirement doivent proposer un même mot-clé décrivant une image qui leur est donnée. Ce jeu a débuté en octobre 2003 et selon les dernières statistiques (août 2007), environ 30 millions de mots ont été affectés aux images, mais le nombre d'images n'est pas divulgué. Un outil sur leur site permet de visualiser parmi 30 000 images celles ayant été annotées avec un mot-clé donné, et l'on se rend compte que la qualité des images retournées est finalement similaire à ce que l'on obtient avec les moteurs de recherche classique. D'autre part, leurs données ne sont pas disponibles librement, donc nous avons toujours besoin de créer notre propre outil pour construire des bases d'apprentissage à partir des images provenant d'Internet.

Une autre difficulté survenant avec de telles images, que nous ne rencontrons pas en général dans des bases construites manuellement, est la présence d'images pertinentes, mais qui ne sont pas idéales pour l'apprentissage d'objets. Par exemple, si l'objet est trop petit dans l'image d'origine, l'extraction des caractéristiques visuelles, et notamment de la texture, sera difficile voire impossible. Il arrive à l'inverse que l'objet soit trop près, et ne soit pas entièrement visible dans l'image, ce qui en fait un mauvais représentant de la classe. Ces deux exemples sont illustrés sur la figure 5.1. Il faut donc être capable de filtrer également les images pertinentes en les réordonnant suivant leur utilité pour l'apprentissage.

FIG. 5.1 – Exemple d'images qui devraient être rejetées si nous envisageons de les utiliser pour l'apprentissage de concepts. Dans l'image de gauche, les zèbres sont trop petits pour permettre d'extraire des caractéristiques visuelles. Dans l'image de droite, nous n'avons qu'une vue partielle de l'objet.

Nous proposons d'exploiter les images provenant d'Internet pour l'apprentissage en plusieurs étapes :

1. collecte d'images d'Internet,
2. segmentation automatique des images pour isoler l'objet à apprendre,
3. mise en ordre de ces images segmentées pour améliorer la précision et éliminer les images non-pertinentes.

Nous montrons ensuite quelques résultats permettant d'évaluer l'utilité et l'efficacité de chaque étape.

5.1 Collecte d'images sur Internet

Nos premiers travaux sur la récupération d'images sur Internet pour former une base sont décrits dans Zinger et al. [156]. Il s'agit d'utiliser la base de données lexicale WordNet [97] pour rajouter automatiquement un mot à une requête, ce mot correspondant à l'hyperonyme direct dans WordNet du concept recherché. Nous avons en effet constaté que l'ajout de l'hyperonyme dans la requête améliore la précision des images obtenues. Depuis, le procédé a été un peu modifié, et se fait par étapes successives, toujours en utilisant WordNet.

5.1.1 Choix de la requête

Pour collecter des images à partir d'Internet, nous séparons les requêtes en trois types, selon qu'il existe ou non une couleur spécifique pour le concept recherché :

- la requête est un objet naturel qui est une feuille dans la base de données lexicale WordNet (par exemple : pomme Granny Smith, chien berger allemand). Cette requête est assez spécifique pour avoir une couleur unique et est utilisée telle quelle pour récupérer des images ;
- la requête est un objet naturel qui n'est pas une feuille dans WordNet (par exemple : pomme, chien). Cet objet a des hyponymes, et nous utilisons tous les hyponymes de cet objet qui sont des feuilles comme requêtes, ce qui revient au scénario précédent.

L'ensemble des images de chaque hyponyme est d'abord traité séparément pour la segmentation et le filtrage suivant le procédé que nous décrivons par la suite, puis toutes les images sont regroupées ;

– la requête concerne un objet artificiel, c'est-à-dire créé par l'homme (par exemple : voiture, tasse, maison). La plupart des objets artificiels existent en différentes couleurs, et n'ont donc pas de couleur spécifique. Dans ce cas, nous spécifions la couleur dans la requête, récupérons un ensemble d'images pour chaque couleur comme pour le premier scénario, traitons ces ensembles séparément puis regroupons le tout.

Afin d'améliorer la précision des images récupérées, la catégorie du concept est ajoutée à la requête comme détaillé par Popescu [114]. Par exemple, *"golden retriever" dog* sera la requête pour chercher des images de *golden retriever*. Cela aide également à lever les ambiguïtés sur les requêtes : *jaguar cat* et *jaguar car* sont deux concepts différents, et chercher *jaguar* dans un moteur de recherche retourne des images des deux concepts mélangées. Le mot servant à indiquer la catégorie est un hyperonyme du concept dans la hiérarchie WordNet, mais n'est pas toujours l'hyperonyme direct car cet hyperonyme n'est pas forcément le meilleur choix pour raffiner la requête. Avec l'exemple ci-dessus de *"golden retriever"*, ajouter l'hyperonyme direct forme la requête *"golden retriever" retriever*, ce qui retourne exactement les mêmes images que *"golden retriever"*. Nous choisissons donc des mots génériques pour les catégories. Ces noms sont très limités et déterminés manuellement, par exemple : *dog*, *cat*, *fish*, *horse*, *zebra*.

Pour chaque concept feuille, nous lançons une requête pour chaque synonyme de ce concept dans WordNet et regroupons toutes ces images dans le même ensemble avant le filtrage. Par exemple : *horned viper*, *cerastes*, *sand viper*, *horned asp* et *Cerastes cornutus* sont la même espèce de serpent selon WordNet. En pratique, le nombre d'images retournées par un moteur de recherche sur Internet est limité à 1000 et nous avons décidé de nous limiter à 300 images : prendre trop d'images augmente le temps nécessaire pour les récupérer et les filtrer et diminue la précision moyenne des images récupérées ; il nous faut toutefois suffisamment d'images pour pouvoir les réordonner afin de sélectionner les N meilleures pour l'apprentissage ($N = 20$ dans nos tests).

5.1.2 Préfiltrage : retrait des cliparts

Nous avons déjà défini, section 4.1.1, ce que nous entendons par clipart et l'algorithme que nous avons mis en place pour les détecter. Les cliparts sont en général des représentations symboliques plus difficiles à traiter, il me semble, que les photographies. Aussi nous limitons nous pour la reconnaissance d'objets aux photographies, et enlevons les cliparts après les avoir détectés automatiquement avec notre algorithme (cf. chapitre 4).

Les cliparts représentent une bonne partie des images retournées lors de requêtes sur Internet. Une évaluation rapide (table 5.1) montre que l'on peut obtenir jusqu'à 25% de cliparts.

Les cliparts obtenus sont de différents types et ont en général un rapport avec la requête. Un exemple pour la requête *bald eagle* est donné dans la figure 5.2.

bald eagle	11%
Bengal tiger	24%
castor canadensis	19%
cerastes	4%
common dolphin	20%
common zebra	7%
dromedary	26%
mean	*15,9%*

TAB. 5.1 – Pourcentage de cliparts parmi les 100 premières images de diverses requêtes sur Google Image Search.

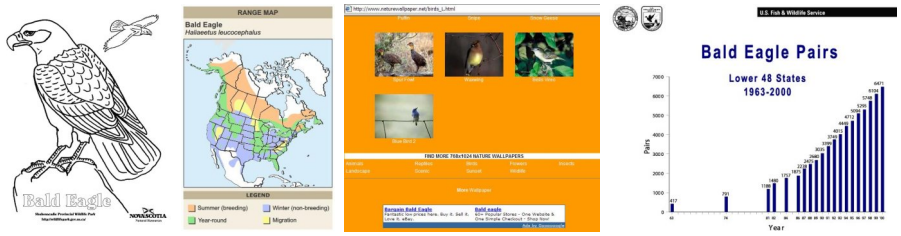


FIG. 5.2 – Exemples de cliparts retournés par Google Image Search pour la requête *"bald eagle"*. Quatre sortes de cliparts sont montrés ici : (de gauche à droite) une représentation de l'animal dessinée à la main, une carte des endroits où l'animal vit, une capture d'écran d'un site proposant des informations sur l'animal et des statistiques provenant d'études scientifiques sur l'animal.

Les images ainsi collectées correspondent à un objet donné, mais il nous reste à localiser cet objet dans les images, puis en fonction de cette localisation, de décider si l'image est une bonne représentante ou non de la classe, tout cela automatiquement. Nous proposons, dans la partie suivante, d'étudier des méthodes pour extraire automatiquement des informations sur la ou les couleurs d'un objet à partir d'un corpus textuel, ou d'un corpus image. Connaître la couleur d'un objet peut, en effet, être utile pour le localiser dans les images comme nous le verrons.

5.2 Extraction de la couleur des objets

Il est plus facile de trouver un objet dans une image lorsque nous connaissons sa couleur. Nous sommes donc intéressés par savoir si un objet donné a une ou plusieurs couleurs spécifiques et quelles sont ces couleurs. Cette couleur peut ensuite être utilisée pour trouver l'objet dans une image en faisant une correspondance entre les noms de couleurs et les espaces de couleur. Cette partie a été publiée dans l'article [98].

Connaître la couleur d'un objet automatiquement, par exemple savoir qu'un gorille est noir, n'est pas facile, et il n'existe actuellement pas de base de données créée automatiquement donnant ce genre d'information. Notamment, tous les objets n'ont pas une

couleur unique bien définie : certains objets comme le zèbre ou le panda sont constitués d'au moins deux couleurs, et d'autres objets, tels qu'une maison ou une chaise, existent en différentes couleurs. En général, les objets avec des couleurs bien définies sont les objets naturels, comprenant les minéraux, les animaux et les plantes, tandis que les objets sans couleur spécifique sont ceux créés par l'homme. En effet, il faut faire une différence entre ces deux types d'objets :

- les objets naturels sont définis par leur aspect visuel. Les végétaux sont classifiés suivant la forme, la couleur et la texture de leurs feuilles, de leurs fruits, etc.
- les objets fabriqués par l'homme sont définis par leur fonction. Ainsi, un téléphone sera défini comme un appareil permettant d'établir une communication orale à distance et non comme un appareil rectangulaire noir, car on peut imaginer toutes les formes et couleurs pour cet objet.

Dans le cas des animaux, certaines sous-espèces ont été nommées en fonction de leurs couleurs, et donc le recours à une base de données lexicale contenant les relations d'hyponymie peut nous aider à déterminer si un animal peut exister en différentes couleurs. Par exemple, WordNet [97] nous indique que « loup » possède les sous-espèces « loup gris », « loup rouge » et « loup blanc » ; « ours » peut être « ours brun », « ours noir », « ours polaire » (blanc). Pour cette raison, il est préférable de ne s'intéresser qu'aux objets qui sont des feuilles de WordNet, c'est-à-dire qui n'ont pas d'hyponymes, afin que la couleur ne présente pas trop de variations.

Étant donné que les objets fabriqués par l'homme peuvent avoir diverses couleurs, la couleur de l'objet est souvent spécifiée dans la page web contenant l'image, ou dans le nom de l'image. Ainsi, la requête *"red car"* retourne plusieurs milliers d'images. Cela n'est pas valable en revanche pour les objets naturels. Par exemple, les requêtes *"brown lion"* et *"tan lion"* retournent très peu d'images (respectivement 58 et 38 sur ask.com), dont la plupart sont des ours en peluche, qui font donc partie de la catégorie « fabriqué par l'homme ». Il n'y a que très peu d'animaux dans les résultats, car les moteurs de recherche d'images utilisent le texte proche de l'image et le nom de l'image pour effectuer la recherche, mais l'information sur la couleur ne sera en général pas présente étant donné que peu de personnes annotent les informations triviales du type « voici un lion de couleur fauve ». Au contraire, les couleurs rares pour les animaux ont plus de chances d'être annotées : la requête *"white lion"* retourne beaucoup plus d'images (3820 sur ask.com) que *"brown lion"* ou *"tan lion"*.

Par conséquent, lors de la récupération des images sur Internet, nous spécifions la couleur dans la requête pour les objets fabriqués par l'homme, mais pas pour les objets naturels. Les requêtes, en italique dans le texte, sont effectuées en anglais, car les moteurs de recherche retournent beaucoup plus d'images dans cette langue. Nos algorithmes sont testés sur les différents types d'objets suivants :

- des animaux avec une couleur prédéterminée : *fire ant* (fourmi rouge), *beaver* (castor), *cougar* (puma), *crab* (crabe), *crocodile* (crocodile), etc.
- des animaux avec deux couleurs : *blue-and-yellow macaw* (ara bleu-et-jaune), *ladybug* (coccinelle), *leopard* (léopard), *panda* (panda), *zebra* (zèbre), etc.
- des objets créés par l'homme : *camera (white, black)* (appareil photo), *cell phone*

(black, blue, green, red, white) (téléphone portable), *chair (black, blue, green, red, white)* (chaise), *cup (black, green, red, white)* (tasse), *Porsche (red)* (Porsche), etc.

Dans cette section, nous testons deux méthodes permettant de connaître automatiquement la couleur d'un objet : la première fondée sur les mots à partir de corrélations dans un corpus textuel, et la deuxième à partir du contenu des images. Nous décrivons au préalable la correspondance que nous avons établie entre les noms des couleurs et les valeurs des pixels dans l'espace TSVal.

Notons que cette correspondance est différente de la correspondance précédemment établie en ce qu'ici nous cherchons, à partir d'un nom de couleur, à obtenir la région de l'espace TSVal qui correspond à cette couleur, avec superposition possible entre les différentes régions définies par chaque couleur. Au contraire, précédemment, nous avions établie une division sans superposition de l'espace TSVal pour attribuer à chaque pixel un nom de couleur unique.

5.2.1 La couleur d'un pixel

Nous élaborons dans cette partie un modèle permettant de mettre en correspondance les valeurs des pixels et les noms des couleurs. Attribuer un nom aux couleurs n'est pas si facile, d'une part car le nombre de couleurs à considérer est à définir, et d'autre part, la séparation entre les couleurs n'est pas bien définie. Berk et al. [8] ont comparé plusieurs systèmes de nommage de couleur selon la facilité de leur utilisation pour un utilisateur, et ont conclu que les systèmes existants contenaient trop de noms de couleurs pour pouvoir être facilement mémorisés et utilisables par un annotateur humain. Ils ont alors proposé un système de nommage de couleur (CNS) consistant en 10 couleurs fondamentales : *gris, noir, blanc, rouge, orange, marron, jaune, vert, bleu, violet.* À ces 10 couleurs peuvent se rajouter des adjectifs tels que *foncé, moyen, clair, vif, ...*, ou les formes en -âtre, tels que *marron rougeâtre*, ce qui leur permet de nommer un total de 627 couleurs.

Un autre système [127] obtient 180 couleurs de référence en utilisant 12 teintes, dont 6 sont considérées comme fondamentales : jaune, rouge, vert, bleu, orange, violet, et les 6 autres sont obtenues par combinaisons linéaires de ces dernières. La luminance est quantifiée en 5 valeurs, et la saturation en 3 valeurs. Les couleurs sont groupées suivant qu'elles soient chaudes-froides, claires-sombres, ce qui permet de faire des requêtes en ces termes.

Cependant, lorsque nous nommons une couleur, nous n'utilisons en général qu'une des 10 couleurs fondamentales énumérées ci-dessus, sans adjoindre d'adjectif. Nous avons donc décidé de considérer seulement 11 couleurs, en rajoutant le rose. Il nous semble en effet que le rose est une couleur très utilisée pour nommer les objets : beaucoup d'habits (principalement féminins) sont de cette couleur, ainsi que certains animaux tels que le cochon domestique ou le flamant rose. Afin d'établir la correspondance entre ces 11 couleurs et les pixels, nous avons recours à l'espace de couleurs TSVal (teinte, saturation, valeur) qui a l'avantage d'être plus sémantique que l'espace RVB. En particulier, la teinte est très proche du concept de noms des couleurs. Dans notre espace TSVal, chaque composante est dimensionnée entre 0 et 255. Une teinte négative est attribuée aux pixels achromatiques (R=V=B).

Nous avions d'abord pensé à séparer clairement les différentes couleurs dans l'espace TSVal, en assignant une unique couleur à chaque triplet (t,s,v) comme nous l'avions fait au chapitre 4. Cependant, nous avons remarqué que certains pixels pouvaient être nommés différemment selon leur contexte, et avons donc décidé de prendre cette ambiguïté en compte, et d'associer parfois plusieurs noms pour un seul triplet (t,s,v). Ainsi, au lieu d'associer un nom pour chaque valeur (t,s,v), nous associons une plage de valeur (t,s,v) pour chaque nom de couleur. Le tableau 5.2 détaille cette association.

Couleur	Teinte	Saturation	Valeur
noir	< 0	tout	0 – 85
	0 – 255	tout	0 – 40
gris	< 0	tout	80 – 180
blanc	< 0	tout	175 – 255
	0 – 30	0 – 90	200 – 255
rose	235 – 245	tout	tout
rouge	0 – 15	tout	tout
	240 – 255	tout	tout
orange	14 – 30	tout	tout
jaune	20 – 50	tout	190 – 255
vert	20 – 50	tout	0 – 200
	50 – 125	tout	tout
bleu	110 – 200	tout	tout
violet	200 – 235	tout	tout
marron	0 – 40	25 – 140	75 – 200
	230 – 255	25 – 135	55 – 190
	< 0	10 – 30	60 – 165

TAB. 5.2 – Correspondance entre les noms des couleurs et les valeurs des pixels dans l'espace TSVal.

Les valeurs ont été choisies expérimentalement après observation sur plusieurs centaines d'images. D'autres méthodes telles que la logique floue pourraient servir à modéliser les frontières floues entre des couleurs, mais la méthode que nous avons développée suffit pour nos applications. De plus, dans cette méthode, les définitions des couleurs sont indépendantes les unes des autres, ce qui rend possible d'ajouter facilement de nouvelles couleurs (cyan, turquoise, ...).

5.2.2 À partir du texte

Les couleurs des objets peuvent être extraites automatiquement à partir de très grand corpus textuels, et nous proposons de le faire en considérant le web comme corpus. L'idée est d'étudier si le nom d'un objet donné apparaît souvent près du nom de certaines couleurs dans les textes. Nous avons expérimenté deux variantes : si nous

souhaitons connaître la couleur d'un castor, la première variante est de chercher les occurrences de *"brown beaver"* où *brown* peut être remplacé par n'importe quelle couleur. Cette méthode avait d'abord été mise en œuvre et testée dans notre laboratoire par Grefenstette [56]. La deuxième variante consiste à chercher le texte *"beavers are brown"*. La présence des guillemets signifie ici que les mots doivent être accolés : d'autres noms de couleurs apparaissant dans la même page web peuvent ne pas être liés à l'objet considéré. La catégorie de l'objet peut être utilisée pour réduire le bruit, et nous pouvons remplacer les exemples ci-dessus par *"brown beaver" animal* et *"beavers are brown" animal*.

Nous avons un vocabulaire de 14 noms de couleurs : *black, blue, brown, gray, green, grey, orange, pink, purple, red, rose, tan, white, yellow*. C'est plus que les 11 couleurs considérées dans le système de nommage des couleurs des pixels [8], mais les couples suivants : *gray/grey*, *brown/tan* et *rose/pink* sont considérés comme synonymes. Pour ces couleurs, les deux nombres d'occurrences sont additionnés. Dans le tableau 5.3, nous listons les cinq couleurs principales retournées pour *beaver* en utilisant Yahoo ! Search, et le nombre d'occurrences entre parenthèses. *Brown* puis *black* sont les deux couleurs principales que nous nous attendons à obtenir, et c'est ce qui est retourné par la seconde variante.

"C beaver"	**C "beaver" an.**	**"beavers are C"**	**"beavers are C" an.**
brown (43 200)	green (10 800)	brown (98)	brown (26)
black (28 100)	brown (7 550)	red (6)	black (1)
green (20 400)	black (2 800)	black (3)	-
gray (11 400)	red (1 050)	blue (1)	-
red (9 610)	gray (783)	orange (1)	-

TAB. 5.3 – Recherche de la couleur des objets par méthode textuelle. C est à remplacer par le nom de la couleur, an. est mis pour « animal ».

Cet exemple est représentatif de ce que nous observons en général pour les autres objets : la deuxième variante (colonnes 3 et 4) donne des résultats plus précis, mais retourne des occurrences plus faibles que la première. L'inconvénient de la première variante est qu'elle est sensible aux noms propres et aux locutions. Par exemple, *Green Beaver* est le nom d'une entreprise et le nom d'un cocktail ; *white house* retourne de nombreux résultats, mais nous ne devons pas en déduire que la plupart des maisons sont blanches. Pour les locutions, l'existence de l'animal *blue whale* fait de bleu la couleur prédominante pour les baleines, et *white chocolate* a plus de résultats que *black chocolate* ou *brown chocolate*. La deuxième variante s'affranchit de ce problème, mais celle-ci parfois ne retourne aucun résultat pour certains objets, comme par exemple pour *passerine* (une espèce d'oiseau), et dans ce cas, il faudrait avoir recours à la première variante.

5.2.3 À partir de l'image

Au lieu d'utiliser le texte pour connaître la couleur d'un objet, une autre méthode consiste à utiliser le contenu des images. Nous proposons de faire une moyenne sur toutes

les images retournées par un moteur de recherche d'images sur Internet. En partant de l'hypothèse que les images contiennent habituellement l'objet centré dans l'image et entouré par un fond, nous ne prenons en compte que les pixels dans une fenêtre au centre de l'image, dont la largeur et la hauteur sont la moitié de celles de l'image, comme illustré sur la figure 5.3.

FIG. 5.3 – Fenêtre des pixels considérés pour déterminer la couleur d'un objet.

Cette fenêtre peut contenir également des pixels de l'environnement, mais nous faisons l'hypothèse qu'en prenant suffisamment d'images, la couleur de l'animal sera prédominante, étant donné qu'elle sera la même pour toutes les images alors que la couleur du fond pourra varier. Le nom de la couleur est alors déduit en utilisant les correspondances présentées dans la section 4.1.2. Le tableau 5.4 énumère les cinq premières couleurs obtenues pour quelques objets.

red Porsche	**beaver**	**crab**	**zebra**	**ladybug**
red (44.5)	brown (24.0)	brown (23.0)	brown (20.7)	red (23.1)
blue (11.8)	white (16.9)	red (22.8)	white (20.6)	white (20.7)
black (11.0)	red (13.0)	white (14.7)	black (17.3)	brown (14.7)
brown (8.7)	black (12.1)	blue (13.0)	blue (10.9)	green (9.5)
white (6.6)	blue (11.9)	black (8.7)	red (9.4)	black (8.1)

TAB. 5.4 – Recherche de la couleur des objets par méthode image. Les cinq couleurs principales pour chaque objet sont données en pourcentage. Les couleurs attendues sont soulignées.

Pour *red Porsche* (première colonne), et en général pour les objets fabriqués par l'homme où la couleur est spécifiée, nous obtenons de bons résultats. Les résultats sont plus intéressants pour les objets dont la couleur n'a pas été spécifiée dans la requête. Pour ces objets, nous avons également des résultats prometteurs, étant donné le bruit d'en moyenne 50% sur les images récupérées sur le web : la première couleur pour *beaver*, *crab* et *ladybug* est correcte. Il y a cependant quelques erreurs. La présence du blanc dans *beaver* et *ladybug* est due au fait que beaucoup de ces images sont des objets détourés sur fond blanc, à la frontière entre les cliparts et les photographies. Les autres erreurs, comme le marron pour *zebra* viennent principalement de l'environnement.

Les deux méthodes que nous avons proposées donnent des résultats prometteurs : dans la plupart des cas, la couleur correcte de l'objet arrive en premier. Toutefois, à

cause des noms propres et du manque d'informations triviales dans les corpus pour la méthode textuelle, et à cause des couleurs de l'environnement pour la méthode image, cette couleur n'est pas clairement séparée des autres couleurs et il n'est donc pas évident de savoir combien de couleurs considérer pour chaque objet. Une amélioration possible serait de combiner les deux méthodes. Une autre idée serait, pour la méthode textuelle, d'utiliser une source plus fiable que le web, telle que Wikipedia[1] ou des dictionnaires[2]. Pour les expériences qui suivent sur la segmentation automatique à partir du nom des couleurs, nous considérons que nous avons construit manuellement la base de données donnant la couleur des objets, ce qui est tout à fait envisageable en pratique, même pour des dizaines de milliers d'objets.

5.3 Segmentation automatique des images

Afin que l'apprentissage des objets ne dépende pas du contexte, c'est-à-dire afin de pouvoir différencier par exemple un objet jaune sur fond bleu d'un objet rouge sur fond bleu, il serait utile de pouvoir automatiquement segmenter les images afin d'en extraire l'objet et de le séparer de son contexte.

Une telle segmentation automatique des objets est un problème difficile : la plupart des images contiennent plus d'un objet, et donc, après avoir appliqué une segmentation automatique, il est difficile de savoir quel objet est celui que nous cherchons. Une solution serait d'utiliser la focale des appareils photographiques pour déterminer la différence entre l'objet dans le plan focal et le fond qui est flou, comme ce qu'ont proposé Swain et al [130] pour la segmentation d'images vidéo. Cependant, dans le cas d'images hors-sujet, il peut y avoir un objet dans le plan focal qui n'est pas l'objet recherché.

Afin de localiser automatiquement l'objet dans les images provenant d'Internet, Ben-Haim et al. [7] ont d'abord segmenté les images en plusieurs régions, puis ont classé ces régions pour identifier des ensembles de régions similaires présentes dans des images différentes. Le plus grand ensemble est alors considéré comme celui représentant l'objet. Une approche similaire est présentée par Russel et al. [119] en utilisant plusieurs algorithmes de segmentation pour chaque image afin de découvrir des objets dans une collection d'image et d'améliorer la qualité de la segmentation par rapport aux techniques n'utilisant qu'un seul algorithme de segmentation. Ces approches sont intéressantes, mais nous nous sommes rendu compte en testant une approche similaire par regroupement de régions issues d'une segmentation automatique que, dans le cas par exemple des animaux d'extérieur, le plus grand ensemble de régions similaires représente souvent le contexte (ciel, herbe, etc.) ou contient des régions sombres telles que les ombres ou les fonds noirs.

Nous avons développé cinq algorithmes de segmentation. Le premier, que nous avons décrit dans [99], utilise une segmentation automatique qui cherche à extraire un objet qui serait centré dans l'image. Le second, publié dans [98], utilise les couleurs possibles des objets dont nous venons de montrer la possibilité de l'extraire automatiquement, et cherche la présence d'un objet de cette (ou ces) couleurs dans l'image. Les trois suivants

[1] http ://wikipedia.org
[2] http ://dictionary.com

étendent cette méthode en segmentant à partir d'une quantification RVB en 125 couleurs au lieu d'utiliser les noms des couleurs. En particulier, le troisième se sert de toutes les images pour apprendre à différencier les couleurs du fond et celles de l'objet ; le quatrième considère chaque image individuellement pour segmenter un objet central et le cinquième est une combinaison de ces deux derniers.

5.3.1 Segmentation par recherche d'un objet central

Nous nous plaçons dans l'optique où les images que nous souhaitons récupérer sont celles qui peuvent permettre un bon apprentissage. Nous cherchons donc des images dans lesquelles l'objet occupe une surface suffisamment importante dans l'image pour permettre d'y appliquer un traitement d'image, notamment l'analyse de la texture nécessite d'avoir une région pas trop petite, mais nous ne voulons pas non plus que l'objet soit trop grand : il faut qu'il soit entièrement contenu dans l'image car nous voulons éviter de remplir la base d'apprentissage avec des parties de l'objet. Par exemple, pour « voiture », nous ne voulons pas garder les images où l'on ne verrait qu'un bout de la carrosserie. De préférence, l'objet serait centré dans l'image et complètement entouré par son environnement.

Nous faisons donc les hypothèses suivantes sur les images :
- il n'y a qu'un objet dans l'image,
- l'objet est centré,
- sa surface est supérieure à 5% de la surface de l'image.

Afin de rechercher un objet central, chaque image est alors segmentée en un maximum de 20 régions par ligne de partage des eaux, avec l'algorithme des cascades [89], qui est rapide et donne de bons résultats. Toutes les régions touchant un bord de l'image sont considérées comme étant du fond, et sont retirées. Les régions restantes sont fusionnées. Si nous obtenons plusieurs régions distinctes, nous gardons la région de plus grande surface. Si cette région a une surface inférieure à 5% de la surface de l'image, nous considérons que l'algorithme n'a pas pu localiser d'objet central, car peut-être l'image n'est pas bonne, et l'image est rejetée.

5.3.2 Segmentation par la couleur supposée de l'objet

Nous proposons ici d'utiliser des connaissances sur les objets afin de les segmenter. Étant donné que nous connaissons l'objet qui est censé être dans l'image, nous pouvons utiliser les connaissances sur cet objet, ici sa couleur, pour guider la segmentation automatique. Une autre motivation de cette méthode est la possibilité de segmenter les objets qui sont constitués de plusieurs couleurs tels que le zèbre ou le panda. La segmentation de tels objets n'est pas possible avec les algorithmes classiques à cause du fort gradient existant aux frontières entre les couleurs. Liapis et al. [77] ont démontré qu'il est possible de segmenter un zèbre en utilisant un histogramme de luminance où les pixels sont classifiés, puis les classes sont propagées. L'inconvénient de cette méthode est que le nombre de classes doit être spécifié avant la segmentation, et donc elle ne

FIG. 5.4 – Segmentation automatique d'une image d'une porsche. L'image de gauche est le résultat de la segmentation en au plus 20 régions. L'image de droite est ce qui reste après avoir enlevé les régions touchant le bord de l'image, et fusionné les autres régions. Il reste deux régions dans cette image, et nous conservons la plus grande, correspondant à la voiture rouge. Dans cet exemple, la deuxième région est également une voiture, mais dans le cas général, il s'agit de bruit.

fonctionnerait pas pour les images complexes pour lesquelles, par exemple, le fond peut être segmenté en plusieurs régions.

Étapes de la segmentation

Pour une image et une ou plusieurs couleurs données, la segmentation de l'image s'effectue par les étapes suivantes :

1. classifier chaque pixel comme appartenant ou non aux couleurs données, comme expliqué page 109. Cela produit une image binaire avec les « pixels d'objets » et les « pixels de fond » ;
2. enlever le bruit ou les objets fins avec une ouverture selon un élément structurant de taille 1 ;
3. effectuer une fermeture par un élément structurant de taille 5 afin que les régions objets proches se rejoignent ;
4. garder la plus grande région ;
5. enlever les trous, définis comme des « pixels de fond » entièrement entourés par des « pixels d'objets ». Cette étape est fondée sur l'hypothèse que l'objet n'a pas de trou, ce qui est le cas pour la plupart des objets.

L'étape se servant de l'information sur la ou les couleurs de l'objet est la première étape, résultant en une image binaire. Il est possible d'utiliser plus d'une couleur dans cette étape, permettant ainsi de segmenter des animaux tels que le zèbre en utilisant à la fois les couleurs « noir » et « blanc » comme couleurs de l'objet. Les quatre autres étapes ont pour but de ne conserver qu'une région et d'adoucir les bords de cette région.

Enlever le bruit à l'aide d'une ouverture (deuxième étape) est utile pour ne pas amplifier le bruit à l'étape suivante lors de la fermeture, comme illustré sur la figure 5.6 (a) :

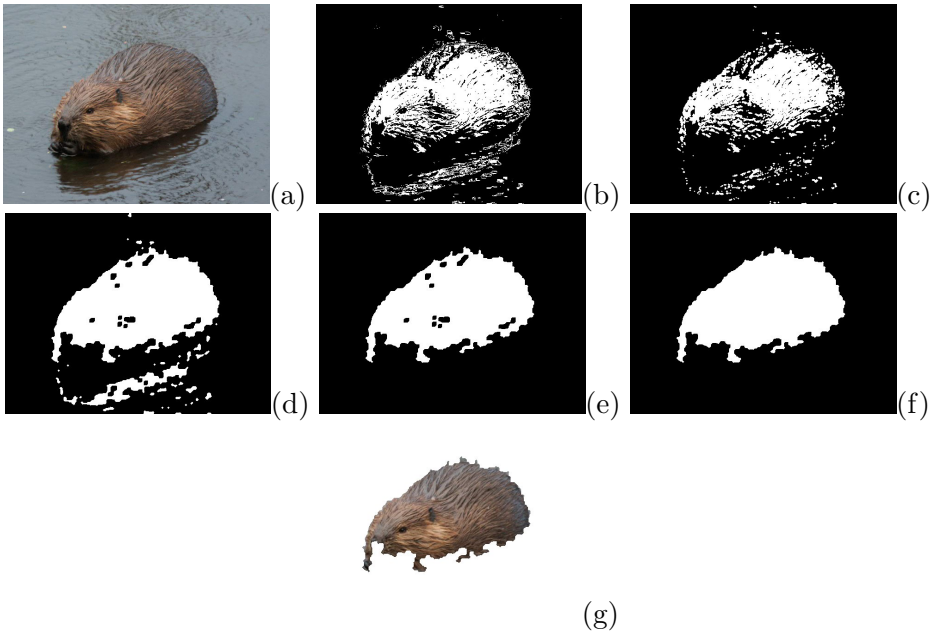


Fig. 5.5 – Étapes de la segmentation. (a) : image originale, (b) à (f) : étapes 1 à 5, (g) : image finale.

les pixels en-dessous du castor se connectent ensemble de telle manière qu'une partie de l'eau soit incluse dans la segmentation finale après avoir enlevé les trous. La fermeture est utile pour fusionner les régions proches ensemble, et améliorer les contours. En comparant le haut du castor sans fermeture (figure 5.6(b)) et avec fermeture (figure 5.5), nous observons que la fermeture permet de séparer le trou en haut du castor du fond, afin qu'il soit effacé lors de l'étape suivante. D'autres résultats sont discutés ci-dessous.

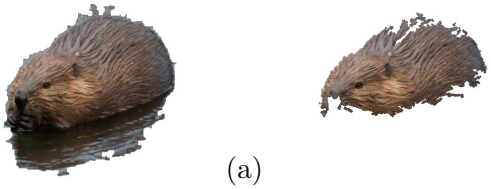


Fig. 5.6 – Même segmentation que dans la figure 5.5, mais (a) : sans éliminer le bruit, en omettant l'étape 2 ; (b) : sans faire la fermeture de l'étape 3.

Résultats de segmentation

Quelques exemples sont présentés dans les figures 5.7 à 5.12.

Ces résultats nous permettent de mieux comprendre le rôle de l'étape d'élimination : cela sert à segmenter les objets qui sont principalement composés de certaines couleurs,

FIG. 5.7 – Segmentation d'un puma avec la couleur marron. La langue est incluse dans la segmentation grâce à l'étape d'élimination des trous.

FIG. 5.8 – Segmentation d'une porsche avec la couleur rouge.

FIG. 5.9 – Segmentation d'un zèbre avec les couleurs noir et blanc.

FIG. 5.10 – Segmentation d'un ara bleu-et-jaune.

FIG. 5.11 – Ces images ont été segmentées avec la couleur noire. Cependant, elles ont un cadre noir qui est alors reconnu comme appartenant à l'objet, et lors du bouchage des trous, on récupère la quasi totalité de l'image originale. Cela ne concerne heureusement qu'une faible proportion d'images.

FIG. 5.12 – Segmentation d'un léopard avec les couleurs jaune et noir. Les animaux camouflés dans leur environnement ne sont pas correctement segmentés avec cette technique, il faudrait inclure un traitement prenant en compte la texture.

mais ont également d'autres couleurs en faible quantité. Par exemple, le puma sur la figure 5.7 est principalement marron, mais pas uniquement : sa gueule est rouge, blanche et noire. Cependant, cette région est entièrement contenue dans une région marron, ce qui permet de la fusionner avec l'objet. La possibilité de segmenter des objets ayant deux couleurs est démontrée sur les figures 5.9 et 5.10.

Néanmoins, segmenter en utilisant seulement les noms des couleurs comme nous le faisons ici, ne donne pas de bons résultats pour les objets cachés dans leur environnement, et nous pourrions probablement améliorer l'algorithme en le combinant avec un algorithme de segmentation par la texture. Nous nous sommes rendu compte également par la suite que l'introduction des noms des couleurs dans la segmentation, s'il donne un aspect sémantique à notre segmentation, présente en revanche un manque de souplesse, notamment car le nombre de couleurs est limité. Par exemple, cet algorithme ne pourra pas segmenter un objet marron sur un fond marron, même si les deux teintes de marron sont différentes, comme sur la figure 5.13.

FIG. 5.13 – Exemple de mauvaise segmentation à cause du manque de finesse résultant de l'utilisation des noms des couleurs : l'image entière est considérée comme le résultat de la segmentation avec les couleurs marron et noir. Le tigre et le fond sont vus par l'algorithme comme étant la même couleur : marron.

Il serait possible de définir plus de noms de couleurs, tels que « marron foncé », « marron clair », « marron rougeâtre » et de les associer avec une plage de valeurs dans l'espace TSVal, mais il semble difficile d'obtenir avec une telle précision la couleur d'un objet. De plus, déjà avec seulement 11 couleurs, il est difficile de nommer la couleur de certains objets afin que cette couleur soit suffisamment discriminante. Par exemple, la couleur d'un dauphin semble être à la limite entre le bleu et le gris, mais bleu est également la couleur de l'eau dans laquelle ils vivent.

Afin de résoudre ces problèmes, nous avons généralisé les algorithmes ci-dessus afin qu'ils ne travaillent plus avec les noms des couleurs, mais directement avec les valeurs des pixels. De cette manière, la méthode textuelle permettant de connaître la couleur d'un objet ne peut plus être appliquée, mais la méthode image, qui est celle qui donne les meilleures résultats, fonctionne toujours. Nous proposons trois nouveaux algorithmes : le premier, inspiré de celui que nous venons de présenter, déduit les couleurs de l'objet et celles du fond à partir de toutes les images collectées depuis Internet pour un même concept. Le second cherche, pour une image prise indépendamment des autres, les couleurs centrales et les couleurs en bordure de l'image afin d'isoler un objet centrale. Le troisième, enfin, combine les avantages de ces deux algorithmes. Nous avons schématisé

ces trois stratégies sur la figure 5.14.

5.3.3 Segmentation globale par la couleur

Cet algorithme s'inspire de la stratégie développée précédemment pour déduire automatiquement la couleur d'un objet à partir de plusieurs images correspondant à un même concept. Les pixels sont divisés en deux catégories : les pixels correspondant à l'objet traité, et les pixels correspondant au fond de l'image. Cette catégorisation permet ensuite de retrouver l'objet dans n'importe quelle autre image dont on sait qu'elle contient l'objet. Au lieu de se limiter aux 11 noms de couleurs, nous définissons 125 couleurs par quantification de l'espace RGB, et nous pouvons facilement faire varier ce nombre si nécessaire. Une autre différence avec les algorithmes précédents est que les pixels du fond de l'image sont également pris en compte par leurs contributions négatives.

La segmentation se fait par les étapes suivantes :

1. Chacun des trois plans de l'espace RVB est quantifié en 5 valeurs, formant un total de 125 couleurs.
2. Une fenêtre centrale est définie, comme étant la fenêtre centrée dont la largeur et la hauteur valent chacune la moitié de la largeur et la hauteur de l'image.
3. Pour chaque image, nous construisons deux histogrammes RVB à 125 composantes : *histocentre* prenant en compte les pixels contenus dans la fenêtre centrale et *histobord* prenant en compte les pixels en-dehors de cette fenêtre.
4. Les deux histogrammes sont normalisés indépendamment en fonction du nombre de pixels considérés pour construire chacun, afin qu'ils puissent être comparés.
5. Pour chaque valeur possible (r, v, b), nous calculons un score $S(r, v, b)$ sur toutes les images qui est augmenté de 1 si, pour une image, $histocentre(r, v, b) > histobord(r, v, b)$ et réduit de 1 sinon.
6. Finalement, un triplet (r, v, b) est considéré comme une couleur de l'objet si $S(r, v, b) > \frac{max(S(r,v,b))}{5}$ et comme couleur du fond dans le cas contraire, où $max(S(r, v, b))$ est le maximum calculé sur toutes les images.

L'ensemble des pixels obtenus est ensuite nettoyé par le même procédé que celui décrit en page 115 pour ne conserver qu'une région connexe.

En comparaison avec l'algorithme précédent qui segmentait les images en fonction du nom des couleurs de l'objet, les régions obtenues avec ce nouvel algorithme sont plus précises. Les noms des couleurs ont un sens pour nous, mais pour l'ordinateur, cela limite le nombre de couleurs considérées à 11. Parmi ces 11 couleurs, typiquement une ou deux seront sélectionnées comme couleurs représentant l'objet utilisées pour la segmentation, mais nous n'avons pas pu déterminer clairement comment savoir s'il faut conserver une ou deux couleurs. Dans l'algorithme que nous venons de présenter, 125 couleurs sont considérées, et il est facile d'augmenter ce nombre. Le nombre de couleurs à considérer comme étant les couleurs de l'objet est automatiquement déterminé à l'étape 6, et est en général supérieur à 10, mais ce nombre varie fortement selon l'objet à segmenter.

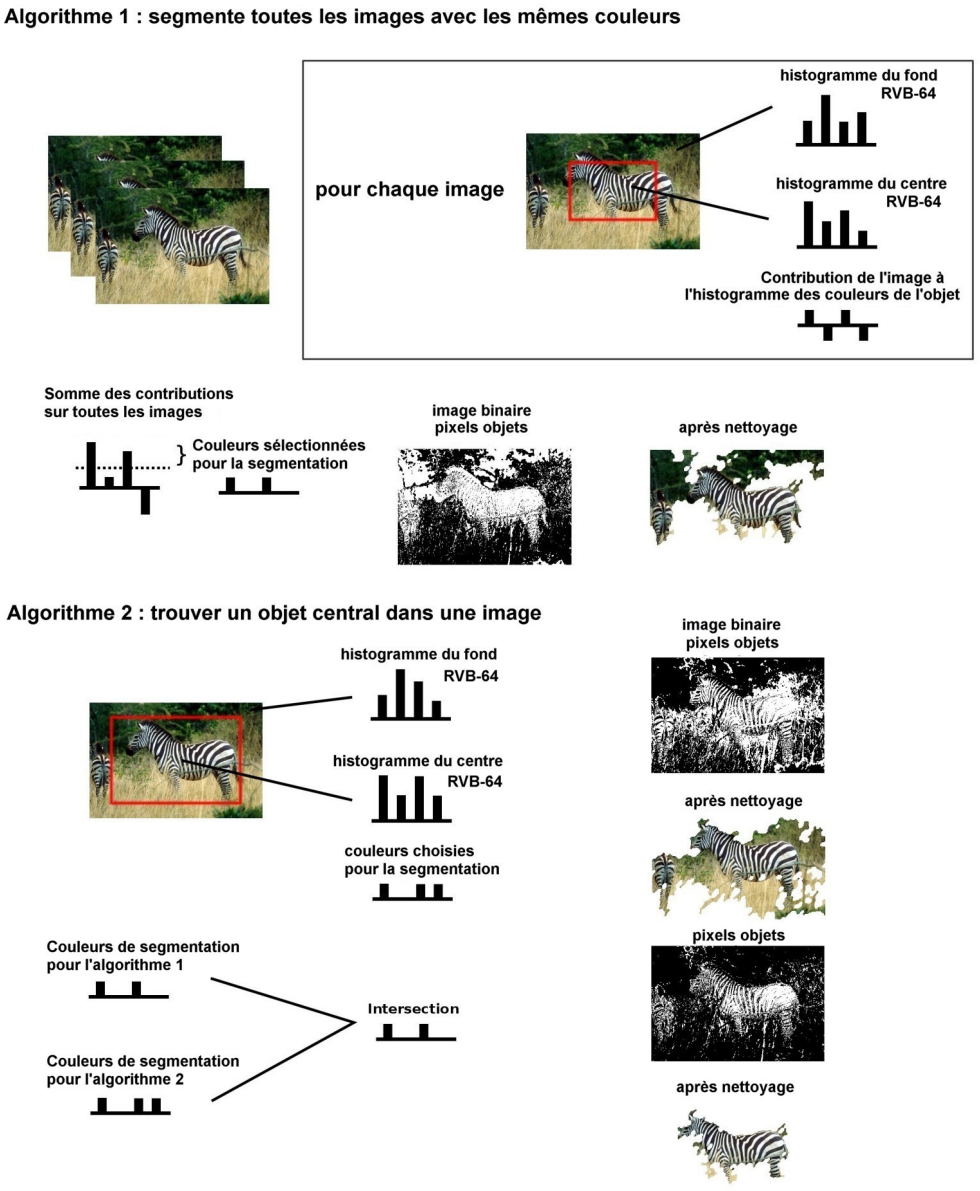


FIG. 5.14 – Schéma des trois nouveaux algorithmes de segmentation proposés s'appuyant sur les histogrammes RGB des objets et des fonds. Le premier algorithme est décrit dans la section 5.3.3, le deuxième en section 5.3.4 et le troisième en section 5.3.5.

Le figure 5.15 montre un exemple des limitations engendrées par l'utilisation des noms des couleurs. Le petit zèbre est considéré comme n'étant composé que de pixels marron clair pour les parties blanches, et marron foncé pour les parties noires. Le nouvel algorithme a pu mieux estimer l'ensemble des couleurs représentant l'animal.

FIG. 5.15 – Exemple des limitations engendrées par l'utilisation des noms des couleurs. À gauche : image originale. Au centre : résultat de la segmentation avec l'algorithme décrit en section 5.3.2 en utilisant les couleurs *blanc* et *noir*. A droite : segmentation avec l'algorithme présenté dans cette section. Les pixels du petit zèbre sont vus comme étant des teintes de marron (clair et foncé). Étendre la définition des couleurs *blanc* et *noir* pour inclure ces pixels causerait des segmentations trop larges dans d'autres images et pour d'autres objets.

Cependant, comme prévu, cet algorithme ne permet pas de corriger le problème de détection d'un objet si la couleur du fond, proche de celle de l'objet, est une couleur de l'objet dans de nombreuses autres images. Cela est illustré sur la figure 5.16 : la couleur du fond marron est aussi une couleur possible pour les tigres du Bengal et observée dans d'autres images. Le fond est de ce fait considéré à tort comme faisant partie de l'objet.

FIG. 5.16 – Résultat de l'algorithme de segmentation globale pour une image de *Bengal tiger*. Cet algorithme n'est pas capable de segmenter un objet dans une image où la couleur du fond est également une couleur de l'objet observée dans d'autres images.

L'algorithme que nous introduirons dans la section suivante se concentre sur la résolution de ce problème.

À propos des paramètres, dans la cinquième étape de la segmentation, nous avons essayé d'introduire un facteur $k > 1$ pour n'incrémenter S que si $histocentre(r, g, b) > k \times histobord(r, g, b)$ et le décrémenter si $histocentre(r, g, b) < (1/k) \times histobord(r, g, b)$ afin d'ignorer les couleurs pour lesquels un pixel n'est pas clairement classifié comme étant objet ou fond, mais cela a eu très peu d'influence sur les résultats.

Dans la sixième étape, utiliser $S(r, g, b) > \frac{\max(S)}{5}$ au lieu de $S(r, g, b) > 0$ pour décider

quelles couleurs sont considérées comme couleurs de l'objet est une autre façon d'ignorer les couleurs qui ont statistiquement presque autant de probabilité de se trouver au centre d'une image que de se trouver au bord. Prendre $S(r,g,b) > 0$ revient à considérer que le nombre de couleurs de l'objet est potentiellement égal ou supérieur au nombre de couleurs du fond. Ce critère n'est pas assez sévère et faisait que des parties du fond étaient considérées comme des parties de l'objet. Utiliser un seuil positif est plus restrictif sur le nombre de couleurs de l'objet. Faire dépendre ce seuil de $\max(S)$ au lieu du nombre d'images (la plus grande valeur que S peut atteindre est le nombre d'image) nous assure de garder au moins une couleur. Nous avons essayé plusieurs valeurs et avons trouvé que $S(r,g,b) > \frac{\max(S)}{10}$ ou $S(r,g,b) > \frac{\max(S)}{5}$ (selon l'objet) offre un bon compromis alors que $S(r,g,b) > \frac{\max(S)}{2}$ n'est pas assez tolérant. Une comparaison des effets de ces différents seuils est montrée sur la figure 5.17.

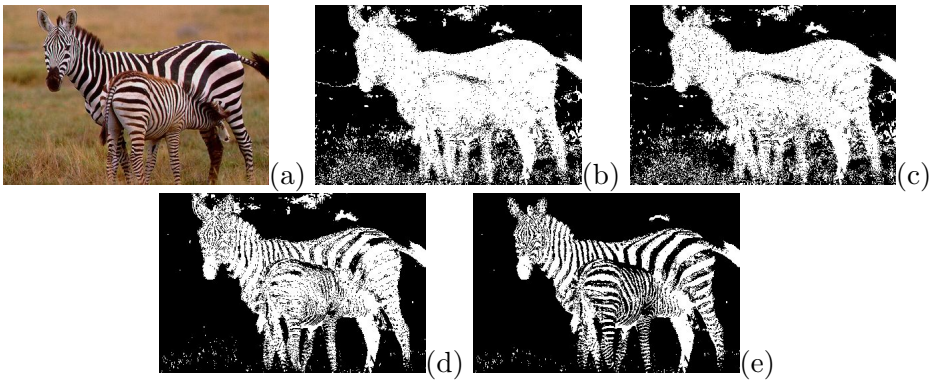


FIG. 5.17 – Variation des résultats de la segmentation pour différentes valeurs du seuil, et nombre de couleurs considérées comme étant celles de l'objet, parmi les 125 couleurs issue de la quantification RGB. (a) Image originale. (b) $S(r,g,b) > 0$: 37 couleurs. (c) $S(r,g,b) > \frac{\max(S)}{10}$: 12 couleurs. (d) $S(r,g,b) > \frac{\max(S)}{5}$: 8 couleurs. (e) $S(r,g,b) > \frac{\max(S)}{2}$: 5 couleurs. Sur cet exemple particulier, les segmentations (b) et (c) sont très proches car les 25 couleurs supplémentaires dans (b) sont en très petite quantité dans cette image. La différence est plus visible sur d'autres images où c'est souvent une partie du fond qui est considérée comme objet avec $S(r,g,b) > 0$ mais comme fond avec $S(r,g,b) > \frac{\max(S)}{10}$.

Nous remarquons que pour $S(r,g,b) > 0$ et $S(r,g,b) > \frac{\max(S)}{10}$, la plupart des pixels des zèbres ont été correctement conservés, mais il y a également du bruit provenant du fond, par exemple entre les pattes. Avec $S(r,g,b) > \frac{\max(S)}{5}$, il y a moins de bruit, et ce qui reste sera éliminé avec le post-traitement, mais une partie des bandes blanches est manquante (elles seront retrouvées avec l'ouverture qui est contenu dans le post-traitement). Avec le seuil $S(r,g,b) > \frac{\max(S)}{2}$, il n'y a presque plus de bruit, mais les parties manquantes sont plus larges, et ne pourront pas être retrouvées avec le post-traitement. Nous finalement décidé empiriquement de prendre $S(r,g,b) > \frac{\max(S)}{5}$ comme seuil pour tous les objets.

5.3.4 Segmentation individuelle par la couleur

Dans cette section, nous nous intéressons au principal défaut de l'algorithme précédent : comment segmenter un objet dans une image où les couleurs de l'objet et les couleurs du fond sont proches ? Avec l'algorithme précédent, il est par exemple difficile de segmenter un objet marron foncé sur un fond marron clair, si la couleur marron clair du fond est aussi la couleur de l'objet dans d'autres images.

L'algorithme que nous proposons ici essaie de segmenter un objet central dans l'image sans se préoccuper de ce que doivent être les couleurs de l'objet. Il permet notamment de segmenter correctement des images pour lesquelles les couleurs de l'objet et du fond sont proches, comme sur l'image de la figure 5.16 pour laquelle l'algorithme précédent n'arrive pas à séparer l'objet du fond.

La principale différence avec l'algorithme précédent est que celui-ci considère les images individuellement et n'utilise qu'une seule image pour apprendre la différence entre les couleurs de l'objet et celles du fond, alors que l'algorithme précédent utilisait plusieurs images du même objet pour cela. Comme nous utilisions plusieurs images, nous avions choisi une fenêtre centrale petite – la moitié de la largeur et de la hauteur de l'image – car cela n'était pas très important si pour certaines images l'objet n'était pas bien centré, du moment qu'en moyenne la plupart des objets l'étaient. Au contraire, dans cette section, une seule image est utilisée et il est donc plus sûr de considérer une fenêtre centrale plus grande pour trouver l'objet. L'algorithme se décompose en plusieurs étapes :

1. Chacun des trois plans de l'espace RVB est quantifié en 5 valeurs.
2. Une fenêtre centrale est définie, comme étant la fenêtre centrée dont la largeur et la hauteur valent chacune les trois quarts de la largeur et la hauteur de l'image.
3. Pour chaque image, nous construisons deux histogrammes RVB à 125 composantes : *histocentre* prenant en compte les pixels contenus dans la fenêtre centrale et *histobord* prenant en compte les pixels en-dehors de cette fenêtre.
4. Les deux histogrammes sont normalisés indépendamment en fonction du nombre de pixels considérés pour construire chacun, afin qu'ils puissent être comparés.
5. Les pixels sont classés suivant leur valeur (r, v, b) : un pixel est considéré comme appartenant à l'objet si $histocentre(r, v, b) > histobord(r, v, b)$ et comme appartenant au fond dans le cas contraire.
6. Enfin, la segmentation est nettoyée avec le même post-traitement que pour les algorithmes précédents.

Cet algorithme permet en effet de segmenter un objet dans une image où la couleur du fond est proche de celle de l'objet (figure 5.18) alors que cela n'était pas possible avec l'algorithme global (figure 5.16).

Nous avions d'abord utilisé une fenêtre centrale dont la hauteur et la largeur étaient seulement la moitié de celles de l'image, comme celle utilisée dans l'algorithme précédent. Cependant, pour de nombreuses images, des parties significatives de l'objet étaient en-dehors de cette fenêtre et n'étaient pas retenues dans la segmentation. Augmenter la

FIG. 5.18 – Exemple d'une amélioration par rapport à l'algorithme précédent (figure 5.16). Le fait que la couleur du fond soit aussi une couleur du tigre du Bengal dans d'autres images n'intervient pas dans cet algorithme.

taille de la fenêtre améliore les résultats pour la plupart des images. Sur la figure 5.19 sont comparés des résultats de la segmentation avec deux tailles différentes de fenêtre.

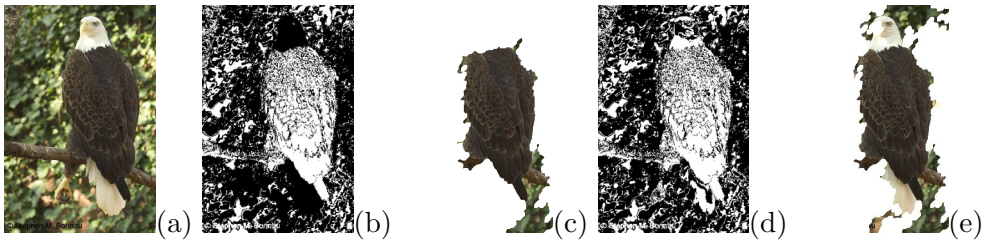


FIG. 5.19 – Effet de la taille de la fenêtre centrale sur la segmentation. (a) Image originale. (b,c) Pixels identifiés comme appartenant à l'objet avec une fenêtre dont la largeur et la hauteur sont la moitié de celles de l'image, et résultat de la segmentation après le post-traitement de nettoyage. (d,e) Pixels identifiés comme appartenant à l'objet avec une fenêtre dont la largeur et la hauteur valent les trois quarts de celles de l'image, et résultat après le post-traitement. Dans (b,c) seule la partie marron de l'aigle est identifiée comme objet. Dans (d,e) quelques parties blanches (la queue et une partie de la tête) sont également retrouvées, avec l'inconvénient d'avoir également retenu un peu de fond.

On peut se demander s'il est préférable de manquer une partie de l'objet mais inclure moins de fond, ce qui est en général ce qui se produit avec une petite fenêtre centrale, ou d'avoir une plus grande de ne pas manquer de partie de l'objet, avec le risque d'inclure du fond, avec une fenêtre plus grande. Cependant, dans la suite, nous intersecterons les résultats de cette segmentation avec ceux de l'algorithme précédent (les deux avant le post-traitement de nettoyage), et dans ce cas, il est préférable de se concentrer sur le fait de retrouver le plus de parties de l'objet possible, ce qui nous fait préférer la deuxième option (une fenêtre centrale plus grande).

Le principal défaut de cet algorithme pour la création automatique d'une base d'apprentissage segmentée est, comme nous l'avons mentionné précédemment, qu'il segmentera n'importe quel objet central, sans considérer le fait que cet objet correspond à une requête déterminée et doit être similaire à d'autres objets issus de la même requête. La cohérence entre les diverses images collectées sur Internet doit être prise en compte

afin d'identifier les images non pertinentes. Combiner l'algorithme précédent avec l'algorithme présenté dans cette section peut permettre de profiter des avantages des deux.

5.3.5 Segmentation par combinaison des approches globale et individuelle

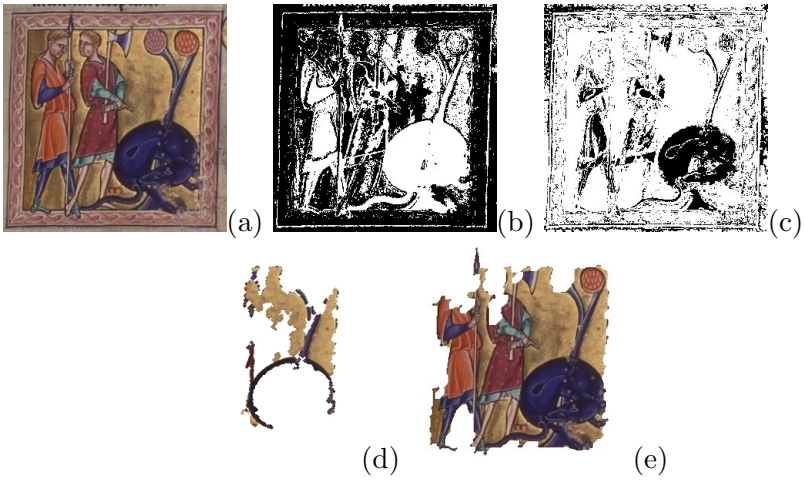
L'idée est à présent de combiner la méthode qui utilise toutes les images pour déterminer les couleurs d'un objet et trouver les objets qui ont cette couleur dans les images avec la méthode qui essaye de trouver un objet central dans une image, capable de gérer les cas où les couleurs de l'objet et celles du fond sont proches.

La combinaison que nous proposons est une intersection des résultats de segmentation de ces deux méthodes pris juste avant d'appliquer le post-traitement qui retire le bruit, remplit les trous et ne garde que la plus grande région. Elle peut s'écrire sous la forme d'un seul algorithme ainsi :

1. Chaque plan de l'espace RVB est quantifié en 5 valeurs.
2. On définit une petite fenêtre centrale W_S dont la largeur et la hauteur valent la moitié de celles de l'image et une grande fenêtre centrale W_L dont la largeur et la hauteur valent les trois quarts de celles de l'image.
3. Pour chaque image, nous construisons deux histogrammes RVB à 125 composantes : $histocentre_S$ pour les pixels contenus dans la fenêtre centrale W_S et $histobord_S$ pour ceux contenus en-dehors de cette fenêtre. Nous construisons de même $histocentre_L$ et $histobord_L$ avec la fenêtre W_L.
4. Chaque histogramme est normalisé par le nombre de pixels (la surface) considérés pour le construire, afin de les rendre comparables.
5. Pour chaque valeur (r, v, b), nous calculons un score S qui est incrémenté si, pour une image, $histocentre_S(r, v, b) > histobord_S(r, v, b)$ et décrémenté dans la cas contraire.
6. Enfin, une valeur (r, v, b) est considérée comme étant une couleur de l'objet si $S(r, v, b) > \frac{\max(S)}{5}$ et $histocentre_L(r, v, b) > histobord_L(r, v, b)$ et comme une couleur du fond sinon.
7. Une image binaire est construite en classifiant les pixels comme objet ou fond.
8. Le résultat est ensuite nettoyé avec le procédé décrit page 115 afin de retirer le bruit et de ne garder que la plus grande région.

Cela est équivalent à intersecter les pixels d'objets obtenus avec les deux méthodes combinées ici juste après l'étape de classification des pixels, et avant le post-traitement de nettoyage.

Nous avons comparé les résultats de segmentation obtenus en intersectant les résultats deux algorithmes avant post-traitement avec ceux obtenus en les intersectant après, et il est apparu que les résultats sont meilleurs si l'intersection est faite avant. Considérons par exemple l'image non pertinente sur la figure 5.20 correspondant à la requête *castor canadensis* (castor).



(a) (b) (c) (d) (e)

FIG. 5.20 – Exemple sur une image issue de la requête « castor »(a). Comparaison des résultats obtenus sur cette image lorsque l'intersection de la segmentation globale (b) et de la segmentation individuelle (c) est faite avant (d) ou après (e) le post-traitement de nettoyage. L'objet central bleu, qui n'est pas de la couleur d'un castor, ne devrait pas apparaître dans le résultat de la segmentation. Les images qui ont une faible surface (d) seront ensuite éliminées lors du reclassement des images, alors que celles avec une grande surface (e) seront conservées.

Les deux méthodes jouent leurs rôles : la segmentation individuelle trouve un objet central bleu avec un fond marron, alors que la segmentation globale trouve que marron est une couleur commune pour les castors mais pas bleu. Les deux méthodes sont donc en contradiction quant aux couleurs à utiliser pour l'objet et pour le fond. Lors du post-traitement, et notamment de l'étape qui remplit les trous des objets, la segmentation globale inclura aussi l'élément bleu qui est entièrement entouré par du marron. L'intersection des deux sera donc égale au résultat de la segmentation globale, résultant en un objet qui n'est pas de la bonne couleur. Faire l'intersection avant le post-traitement résulte en un objet marron, ce qui est cohérent avec le fait que nous traitons des images de castor. Cet exemple a été choisi pour bien mettre en évidence les conséquences du choix du moment où l'intersection est faite. Dans la plupart des cas, la conséquence n'est pas aussi visible. Nous avons remarqué cependant que faire l'intersection après le post-traitement a tendance à trouver un objet dont certaines parties ne sont pas de la bonne couleur et ne seraient pas trouvées en intersectant avant.

Dans beaucoup de cas, le résultat de l'intersection correspond à l'une des deux segmentations intersectées (la segmentation individuelle ou globale selon l'image). Faire l'intersection revient alors à choisir la plus petite des deux régions obtenues, qui est souvent la meilleure. Dans d'autres cas, chacune des deux segmentations inclut une partie du fond dans l'objet, mais deux parties différentes, et faire l'intersection permet d'obtenir une meilleure segmentation. Cela est illustré sur la figure 5.21.

Nous nous attachons dans la partie qui suit à définir des critères pour ré-organiser

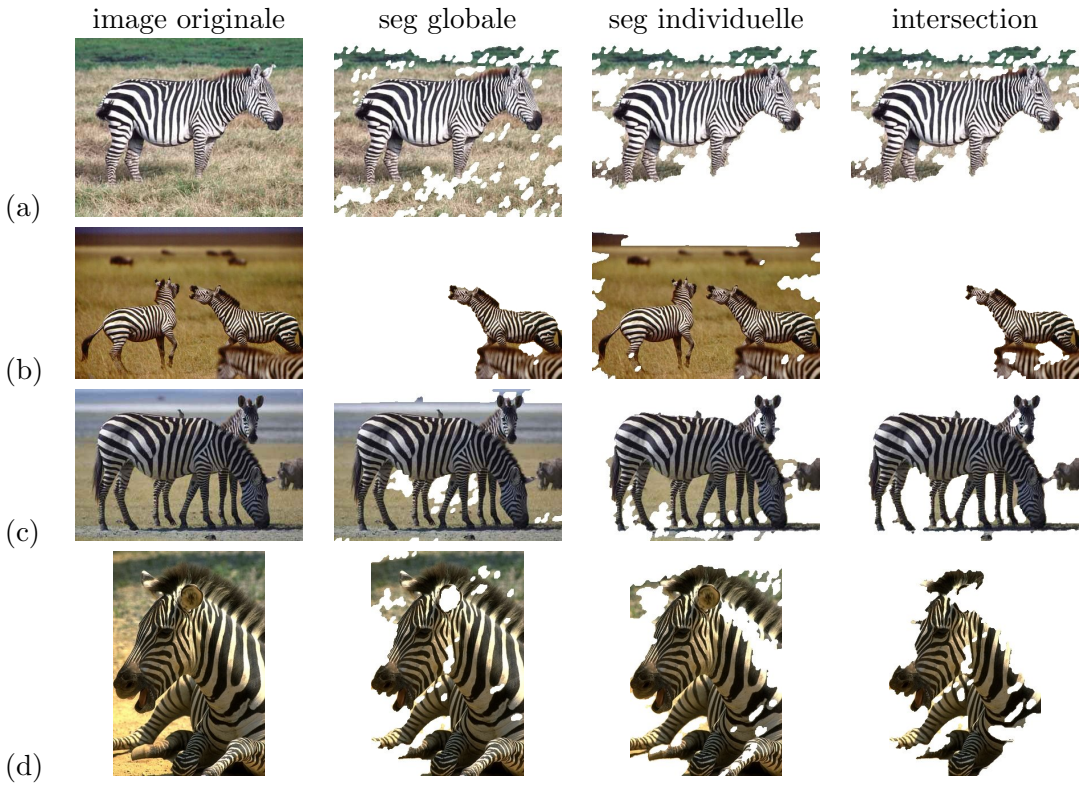


FIG. 5.21 – Comparaison des résultats de l'intersection avec ceux des segmentations globales et individuelles. (a) La segmentation globale a gardé une partie de l'herbe grise dans le résultat de la segmentation et l'intersection est égale à la segmentation individuelle qui est incluse dans la segmentation globale. (b) La segmentation individuelle a trouvé de l'herbe au centre de l'image, et la segmentation globale est meilleure puisqu'elle a déterminé que la couleur de cette zone d'herbe n'est pas celle d'un zèbre. Dans ce cas, l'intersection est proche de la segmentation globale. (c) Les segmentations individuelle et globale incluent deux parties différentes du fond. L'intersection donne donc une meilleure segmentation que les deux autres. (d) Dans cet exemple, la segmentation individuelle est la meilleure. L'intersection ne contient pas de fond, mais une partie du zèbre est manquante.

les images afin de décider lesquelles ne sont pas pertinentes.

5.4 Amélioration de la précision

L'amélioration de la précision, ou le rejet des intrus, est une étape importante afin de garantir une base « propre » adéquate pour l'apprentissage. Même s'il est bien connu que les résultats des recherches d'images sur le web sont très bruités, peu de travaux se sont intéressés à les améliorer. Lin et al. [81] ont utilisé un modèle de pertinence sur le texte des pages web contenant les images pour reclasser les résultats. Ils déclarent obtenir de 30% à 50% d'augmentation de la précision sur les 50 premières images, mais sans utiliser le contenu des images.

Cai et al. [18] ont proposé de grouper les images en utilisant le contenu des images conjointement avec le texte et les liens des pages web. Leur idée était que certaines requêtes peuvent retourner des résultats représentant plusieurs concepts. Par exemple, la requête *pluto* mélange les images à propos de la planète naine et à propos du personnage de Disney ; ou la requête *chien* retourne des images de plusieurs espèces différentes de chiens, et en les classifiant, il serait possible de séparer ces différents sujets. Cependant, Cai et al. n'ont pas essayé d'éliminer les intrus ou de reclasser les images.

Un moyen d'enlever les images non pertinentes serait d'essayer de détecter des similarités entre les différentes images, par exemple en utilisant les points d'intérêt. Cette méthode est très difficile à appliquer aux images du web car, comme nous l'avons dit, beaucoup d'images ne contiennent pas l'objet recherché : le bruit est en moyenne de 50% et peut aller jusqu'à 85% pour certaines requêtes. De plus, il y a beaucoup de variations en termes de qualité d'images, si bien qu'il est difficile de trouver un motif se répétant. Dans Ben-Haim et al. [7], les images sont reclassées selon la plus petite distance entre une région de l'image et le centre du groupe principal décrit en section 1.1.

Fergus et al. [49, 47], ont publié une approche donnant des résultats prometteurs pour nettoyer, réorganiser et faire de l'apprentissage à partir des résultats de Google Image Search. Ils appliquent dans un premier temps plusieurs détecteurs de points d'intérêt différents, puis calculent les descripteurs SIFT sur les régions entourant ces points. Ces points servent alors à entraîner un modèle sémantique latent probabiliste invariant par échelle et translation (TSI-pLSA) pour la classification d'objets. Ce modèle peut être alors appliqué sur les données brutes pour reclasser les images et enlever les images non pertinentes. Fergus et al. ont annoncé une amélioration d'environ 20% en précision avec un rappel de 15% (c'est-à-dire en rejetant 85% des images pertinentes).

5.4.1 Nettoyage par regroupements

Nous avons développé deux méthodes permettant d'éliminer les images non pertinentes récupérées sur Internet. Dans [99], nous avons expliqué comment il était possible de nettoyer et reclasser les images en utilisant des techniques de groupement. Les régions automatiquement segmentées dans chaque image sont indexées avec des descripteurs de texture et de couleur, qui servent à grouper les images avec l'algorithme des plus proches

voisins partagés (*shared nearest neighbor, SNN*). Les images qui ne se retrouvent dans aucun groupe sont rejetées, et chaque groupe est trié en comparant les couleurs qu'il contient avec la liste des couleurs probables de l'objet recherché. La figure 5.22 montre des exemples de résultats.

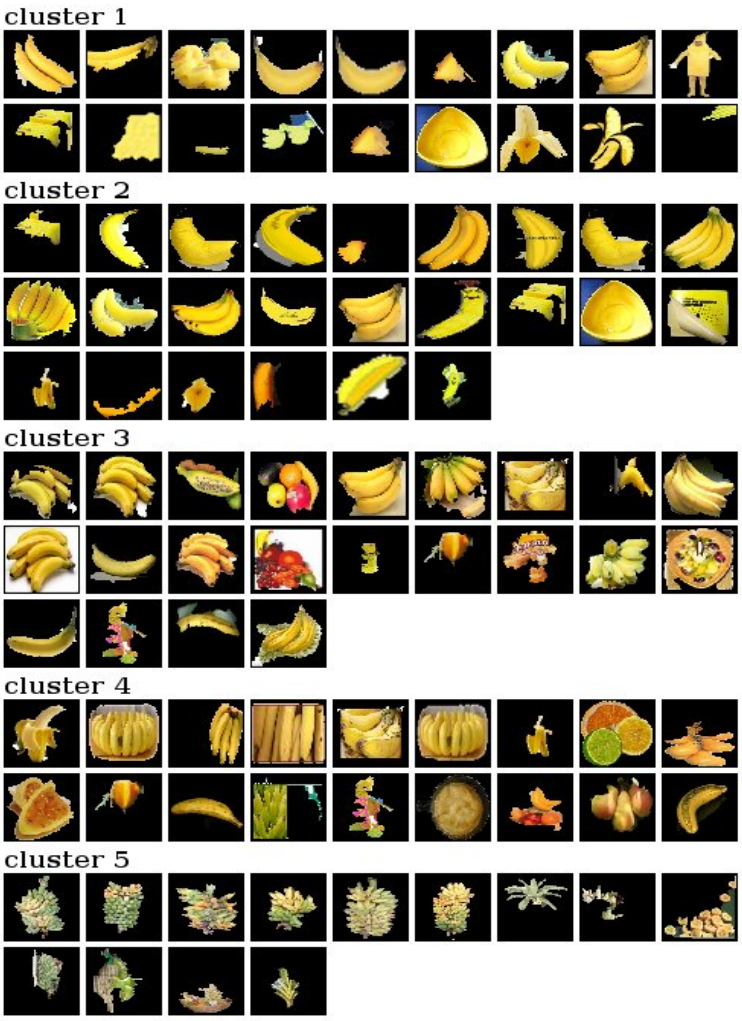


FIG. 5.22 – Exemple des cinq premiers groupes d'images obtenus pour le mot-clé *banana* en les triant suivant leur couleur. Viennent d'abord les clusters contenant principalement du jaune, puis du vert.

Nous avons observé une amélioration de la précision en utilisant cet algorithme sur le résultat de la segmentation par objet central, mais en revanche, la précision avant et après filtrage reste la même quand elle est appliquée sur les régions obtenues avec la segmentation par la couleur, même en ne conservant que le descripteur de texture pour l'algorithme SNN – en effet, les images étant segmentées par la couleur, elles ont toutes des couleurs très proches, ce qui rend peu utile un descripteur de couleur.

5.4.2 Nettoyage par analyse des résultats de la segmentation automatique utilisant les noms des couleurs

Ici, nous proposons d'enlever les images non pertinentes en utilisant la couleur des objets. L'idée est d'analyser à quoi la région obtenue par segmentation ressemble. Dans une situation idéale, l'objet est centré dans l'image, et totalement contenu dans celle-ci, sans être trop petit. Nous avons donc décidé de rejeter une image si la région segmentée :

- occupe moins de 20% de la surface de l'image,
- touche plus de 80% des pixels du bord,
- est telle que la distance de son barycentre (x_R, y_R) au centre de l'image (x_I, y_I) est plus petite que 40% de la distance du coin de l'image au centre :

$$(x_R - x_I)^2 + (y_R - y_I)^2 < 0.4 * (x_i^2 + y_i^2)$$

Objet	**Avant**	**Après**	**Objet**	**Avant**	**Après**
beaver	40,2%	72,7%	green cell phone	17,9%	46,2%
cougar	72,0%	93,0%	red cell phone	22,5%	37,5%
crab	55,8%	70,8%	white cell phone	12,9%	30,0%
crocodile	79,5%	85,7%	black chair	66,7%	83,3%
fire ant	61,0%	50,0%	blue chair	50,9%	58,8%
blue-and-yellow macaw	79,4%	82,6%	green chair	22,2%	50,0%
ladybug	72,8%	73,3%	red chair	70,6%	88,5%
leopard	84,9%	89,7%	white chair	62,0%	83,3%
panda	80,0%	81,5%	black cup	35,2%	66,7%
zebra	76,7%	89,3%	green cup	34,5%	80,0%
black camera	69,1%	79,1%	red cup	36,5%	50,0%
white camera	18,6%	50,0%	white cup	33,9%	66,7%
black cell phone	43,0%	87,0%	red Porsche	85,0%	85,7%
blue cell phone	28,6%	42,4%			
moyenne par classes	**52,0%**	**69,4%**			
moyenne par images	**53,5%**	**72,0%**			

TAB. 5.5 – Précisions avant et après filtrage pour différents objets. La précision en moyenne passe de 53,5% à 72% en gardant 25% des images. Seul *fire ant* a une baisse de précision, car beaucoup d'images de peau irritée par des morsures de fourmis sont gardées avec la couleur rouge.

La précision augmente pour toutes les classes, sauf pour *fire ant*, comme expliqué dans la légende du tableau. Quatre classes ont une précision inférieure à 50% après filtrage (contre 12 avant filtrage) : ce sont les téléphones portables, dû au fait que beaucoup d'images représentent un autre objet, ayant la couleur spécifiée dans la requête. Ces erreurs ne peuvent pas être évitées avec notre méthode mais cela serait possible en rajoutant un groupement par texture.

Ne garder que 25% des images n'est pas gênant quand on étudie les images du web : étant donné leur quantité, un utilisateur typique d'un moteur de recherche d'images sur le web ne regardera pas la totalité des images disponibles, et est donc plus intéressé par la qualité (précision) que par la quantité (rappel). Pour une utilisation en apprentissage d'objets, réduire le bruit est également plus important qu'obtenir beaucoup d'images.

Dans le tableau 5.5, les paramètres ont été choisis pour maximiser la précision tout en gardant au moins 25% des images récupérées sur le web. Nous avons également essayé d'introduire un critère pour rejeter les régions plus grandes qu'une certaine taille, mais le critère sur les pixels du bord a donné de meilleurs résultats. Nous avons étudié l'influence de la précision sur chaque critère indépendamment sur la figure 5.23.

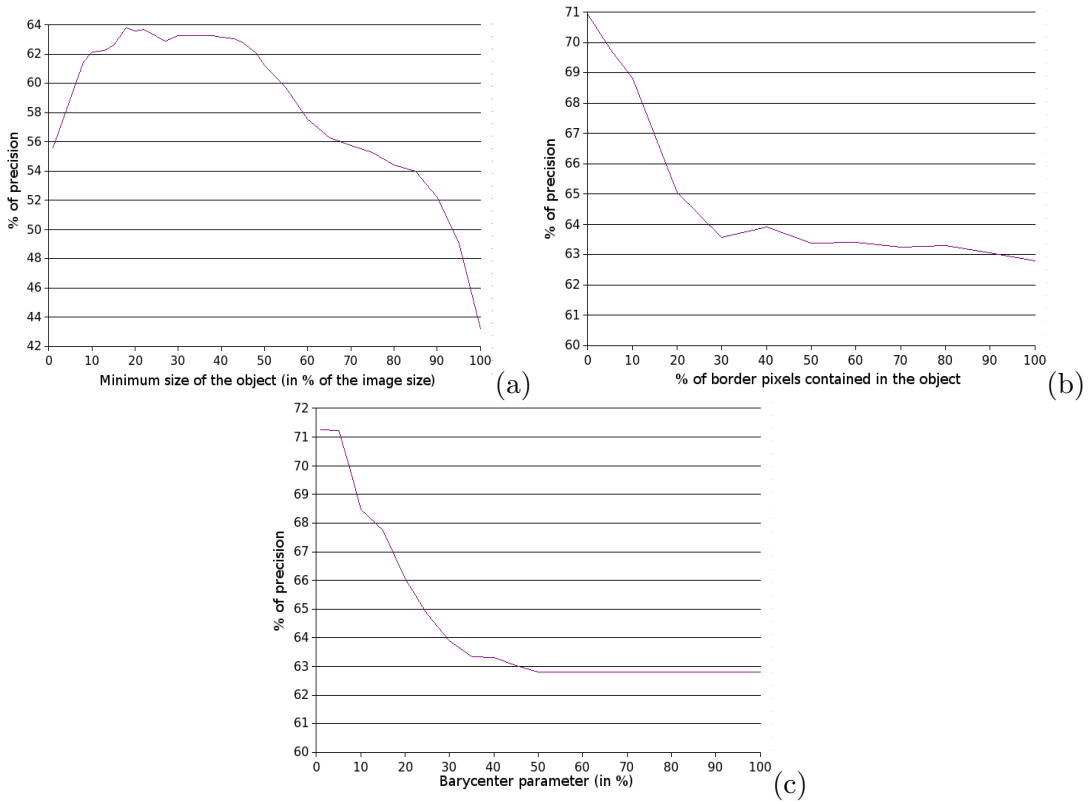


FIG. 5.23 – Influence des paramètres sur la précision après filtrage, (a) : paramètre de la taille ; (b) : paramètre du bord ; (c) : paramètre du barycentre. Les graphiques pour les paramètres du bord et du barycentre sont calculés sur les régions dont la taille est entre 10% et 50% de la taille de l'image, correspondant aux meilleures images pour le critère de la taille (a).

Nous observons d'abord que la plus grande proportion d'images pertinentes sont celles où la taille de l'objet est entre 10% et 50% de la taille de l'image. Les deux autres graphiques ont été calculés sur cet ensemble, et montrent qu'un objet a plus de chances d'être pertinent s'il ne touche pas le bord de l'image et s'il est centré. Il

est possible d'améliorer encore la précision jusqu'à 86% en rendant les paramètres plus contraignants, ce qui réduit le nombre d'images après filtrage jusqu'à seulement 5% du nombre d'images initiales, comme montré sur la figure 5.24.

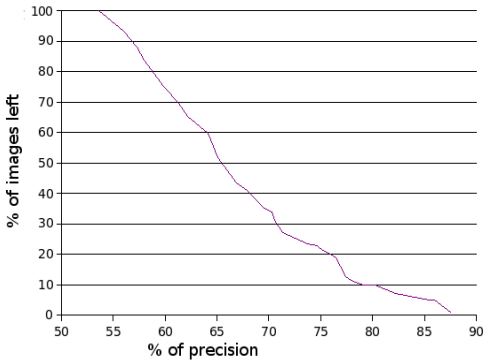


FIG. 5.24 – Relation entre la précision (en abscisse) et le nombre d'images (en ordonnée).

En choisissant de conserver 15% des images comme dans [49], la précision augmente de 24% (de 53% à 77%), ce qui est comparable aux 20% qu'ils annoncent, même si les deux bases de données sont différentes.

Reclasser les images du web

Une autre remarque à propos des images retournées par les moteurs de recherche sur le web est qu'elles ne sont apparemment pas triées visuellement. Nous proposons ici d'utiliser les critères développés ci-dessus pour effectuer le tri. Pour une région segmentée donnée dans une image, nous définissons :

- ν comme la surface de la région objet divisée par la surface de l'image.
- B est le nombre de pixels du bord de l'image inclus dans la région de l'objet divisé par le nombre total de pixels bordant l'image (périmètre). Il vaut 0 si l'objet est totalement inclus dans l'image et augmente si l'objet touche le bord de l'image, signifiant qu'il n'y a peut-être qu'une partie de l'objet dans l'image comme par exemple sur l'image de droite de la figure 5.1 (page 105).

Nous avons vu dans la section précédente que les objets ont plus de chances d'être pertinents si B est proche de 0 et ν est entre 0,2 et 0,4. Nous proposons donc le score Σ suivant :

$$\Sigma = (1 - B) * f(\nu) * g(x_R, y_R)$$

avec

$$f(\nu) = \begin{cases} 1 & si \quad 0,2 \leq \nu \leq 0,4 \\ \frac{\nu}{0,2} & si \quad \nu < 0,2 \\ \frac{1-\nu}{0,6} & si \quad \nu > 0,4 \end{cases}$$

et

$$g(x_R, y_R) = 1 - \frac{(x_R - x_I)^2 + (y_R - y_I)^2}{(x_i^2 + y_i^2)}$$

Les images sont ensuite triées par ordre décroissant de Σ. Des résultats pour les requêtes *beaver* et *green cup* sont montrés respectivement sur les figures 5.25 et 5.26.

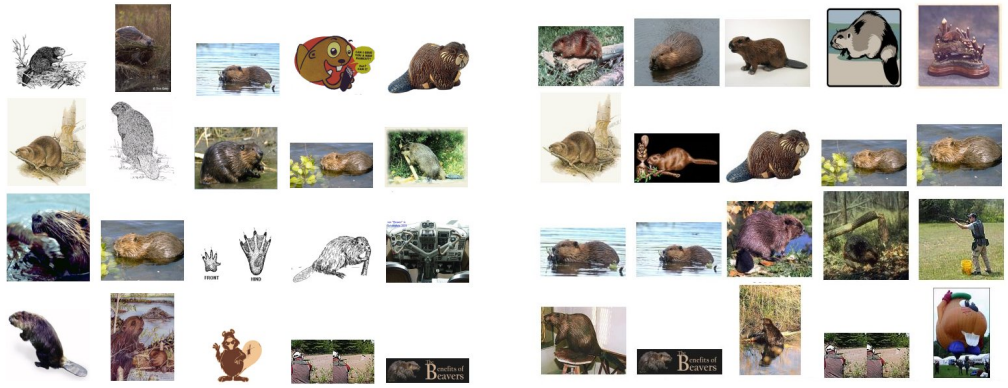


FIG. 5.25 – Reclassement des images pour la requête *beaver*. À gauche : les 20 premières images retournées par ask.com, à droite : les 20 premières après reclassement des 100 premières images de ask.com.

FIG. 5.26 – Reclassement des images pour la requête *green cup*. À gauche : les 20 premières images retournées par ask.com, à droite : les 20 premières après reclassement des 100 premières images de ask.com.

Après reclassement, nous observons principalement des images avec un objet central de la bonne couleur, et dans l'exemple des tasses vertes, l'amélioration en précision bien visible. Dans la cas où la recherche serait simplement *cup*, et comme l'objet n'a pas de couleur spécifique, nous pourrions prévoir de lancer les requêtes du type *color cup* pour toutes les couleurs, de reclasser chaque résultat, puis de réunir le tout, ou d'afficher chaque couleur séparément.

5.4.3 Nettoyage par analyse des résultats de la segmentation automatique combinant les méthodes individuelle et globale

Après avoir développé le dernier algorithme de segmentation présenté précédemment (section 5.3.5), combinant une méthode individuelle et une méthode globale, nous avons mené une deuxième série d'expériences sur le reclassement des images.

D'après les variables ν et B définies ci-dessus, nous calculons le score Σ suivant :

$$\Sigma = (1 - B) \times f(\nu) \text{ avec } f(\nu) = \begin{cases} 1 & \text{si} \quad 0,2 \leq \nu \leq 0,6 \\ \frac{\nu}{0,2} & \text{si} \quad \nu < 0,2 \\ \frac{1-\nu}{0,4} & \text{si} \quad \nu > 0,6 \end{cases}$$

Les images sont triées par ordre décroissant de Σ. Nous ne pouvons plus utiliser le critère précédent qui avantageait également les régions proches du centre, car trouver un objet central fait désormais partie de l'algorithme de segmentation. Un objet avec une grande valeur pour Σ est un objet dont la taille est comprise entre 20% et 60% de la taille de l'image et qui est totalement entouré par du fond, c'est-à-dire qu'il ne touche pas les bords de l'image. Ici, la taille des « meilleurs » a été changée par rapport à la série d'expériences précédentes (le ν optimal était alors entre 20% à 40%) car les résultats obtenus sont meilleurs avec ce nouveau taux, pour l'algorithme de segmentation considéré ici. Ce taux ne peut être choisi qu'empiriquement, mais doit toutefois respecter la proportion typique des objets dans l'image, qui s'observe notamment dans le graphique de la figure 5.32 (voir plus loin : la mesure de la segmentation pour l'image entière correspond en fait au pourcentage de l'image que l'objet occupe)

Pour le critère sur la forme, B, la boîte englobante de la région peut être utilisée au lieu de celle du bord de l'image, donnant de meilleurs résultats, notamment à cause des images contenant un cadre, qui sont en quantité non négligeables sur Internet. Cela peut par contre poser problème pour les objets qui ont des bords verticaux et horizontaux droits, tels que les bâtiments, les moniteurs ou les étagères. Disposer d'un algorithme capable de détecter et d'enlever les cadres serait probablement une meilleure solution.

Cette méthode pour le reclassement des images présente encore un autre problème. Prenons par exemple le cas des images de zèbres. Les couleurs utilisées pour la segmentation sont des couleurs proches du noir et du blanc. Avec le reclassement présenté ci-dessus, une région entièrement noire, une région entièrement blanche et un zèbre de la même forme auront le même score. Afin de donner plus d'importance à une région qui, par exemple, aurait à la fois du blanc et du noir dans des proportions données calculées d'après toutes les images, nous proposons l'amélioration suivante :

1. Construire l'histogramme médian à 125 composantes H_M à partir de toutes les segmentations. Il est obtenu en calculant la valeur médiane de chaque composante de l'histogramme sur toutes les images.
2. Pour chaque segmentation, nous introduisons une similarité de couleur C_s comme étant une intersection d'histogrammes entre l'histogramme H_I de l'image et l'his-

togramme médian H_M :

$$C_s = \sum_{k=1}^{125} \min(H_I(k), H_M(k))$$

Nous définissons alors le score pour reclasser les images par $R_s = C_s \times \Sigma$ et considérons que les images les plus pertinentes sont celles avec le score le plus élevé.

Analyse qualitative des résultats

Il est difficile de faire une évaluation de cet algorithme, étant donné qu'il faut évaluer à la fois la qualité de la segmentation et la précision des meilleures images après le reclassement. Nous avons décidé de discuter d'abord qualitativement à partir des vingt premières images de trois requêtes, affichées sur les figures 5.27 à 5.29. Nous ferons ensuite une évaluation plus quantitative sur 20 requêtes.

FIG. 5.27 – Les 20 premières segmentations pour la requête *common zebra*. La précision est de 100%, et la plupart des segmentations sont bonnes (i.e. permettraient d'apprendre le concept par exemple avec un apprentissage texture/couleur).

Deux objets naturels qui ne sont pas faits d'une couleur uniforme sont montrés ici : *zebra* (figure 5.27) et *Bengal tiger* (figure 5.28). Les algorithmes traditionnels de segmentation ne peuvent pas en général segmenter ces objets à cause des forts gradients causés par les rayures. Un résultat de la segmentation est aussi montré pour un objet fabriqué par l'homme : *yellow Ferrari* (figure 5.29). Les résultats montrés ici, en termes de précision et de segmentation sont très bons pour les animaux. Pour *yellow Ferrari*, la précision des images sélectionnées est bonne, mais seul le chassis de la voiture a été segmenté. Les roues et les vitres sont manquantes car la couleur noire a été principalement observée dans les fonds. Sur cet exemple particulier, les résultats seraient meilleurs en considérant l'enveloppe convexe de ces objets.

Cet algorithme fonctionne mieux en général que celui présenté précédemment : la qualité des segmentations est notamment améliorée car l'algorithme peut mieux isoler

FIG. 5.28 – Les 20 premières segmentations pour la requête *Bengal tiger*. La précision est de 100% : toutes les images représentent le tigre, et nous avons à la fois des images de l'animal en entier et des images de la tête.

les objets sur un fond d'une couleur proche. La précision obtenue après le reclassement est également meilleure.

Cependant, pour les requêtes contenant trop d'images non pertinentes au départ, les performances ne sont pas bonnes. Par exemple, pour la requête *banana fruit*, nous nous attendions à obtenir principalement des objets jaunes. Les 20 « meilleures » images obtenues sont montrées sur la figure 5.30.

En effet, parmi les 100 premières images retournées par Yahoo ! Image Search, et après avoir éliminé les cliparts, il reste seulement 20 images contenant une banane jaune au centre de l'image, ce qui signifie que le bruit est environ de 80%. Les autres images montrent principalement des bananiers (même si nous avions utilisé *fruit* pour désambiguïser la requête), des images de plusieurs fruits ensembles, ou des poduits dérivés de la banane. Dans le processus de la segmentation, jaune a bien été identifié comme une couleur de l'objet, mais vert, rouge et orange également.

Nous avions fait l'hypothèse en concevant cet algorithme que le bruit serait au plus de 50%, et il fonctionne dans ce cas. Par exemple, pour la requête *yellow Ferrari* dont les résultats sont montrés ci-dessus, il y a environ 43% d'images non pertinentes, c'est-à-dire où nous ne pouvons pas reconnaître de Ferrari jaune, sur les 100 premières images retournées par le moteur de recherche. Le bruit pour les requêtes *common zebra* et *Bengal tiger* est bien plus faible et avoisine les 10%. Le choix des mots clés est donc essentiel dans ce procédé de segmentation pour commencer avec un ensemble d'images contenant suffisamment d'images pertinentes.

FIG. 5.29 – Les 20 premières segmentations pour la requête *yellow Ferrari*. La couleur a été ajoutée à la requête, comme expliqué dans la section 5.1. La précision est de 95%, l'image non pertinente est encadrée en rouge.

Évaluation du reclassement

Une image est considérée comme pertinente si l'objet recherché peut être identifié dans l'image. Des objets tels que des jouets, des sculptures, ou des peintures sont également comptées comme pertinentes si l'objet peut être facilement reconnu. Les images où l'objet ne peut pas être reconnu car il est trop loin, trop flou, ou car la partie montrée n'est pas caractéristique de l'objet sont comptées comme non pertinentes. Par exemple, des images de l'intérieur de l'avion ne sont pas considérées comme pertinentes pour la requête *Boeing 777*. Le tableau 5.6 compare les précisions obtenues avec les images du moteur de recherche Yahoo ! sans traitement, et après notre reclassement.

L'amélioration moyenne de la précision est d'environ 5% sur les 20 premières images. La taille des objets dans l'image n'est pas prise en compte dans cette évaluation. En général, les objets retournés par notre méthode sont plus grands que ceux obtenus sans reclassement, et sont donc mieux adaptés pour constituer une base d'apprentissage. Nous évaluons ensuite la qualité des segmentations et donnons quelques statistiques sur la taille de ces objets dans les 20 premières images, et sur la proportion qui est correctement segmentée.

Évaluation de la segmentation

Dans nos travaux, nous avons toujours comme objectif d'utiliser les résultats de la segmentation afin de construire une base de données d'apprentissage permettant de faire de la classification d'objets. Nous n'avons donc pas vraiment l'ambition d'obtenir une segmentation parfaite, mais plutôt une segmentation qui ne contient pas trop de fond, pour ne pas bruiter l'apprentissage, et suffisamment de parties de l'objet (mais pas nécessairement l'objet en entier) pour permettre d'extraire les caractéristiques visuelles. Nous pouvons cependant évaluer notre algorithme comme si l'objectif était d'avoir une segmentation parfaite, afin d'avoir une idée sur la qualité des résultats.

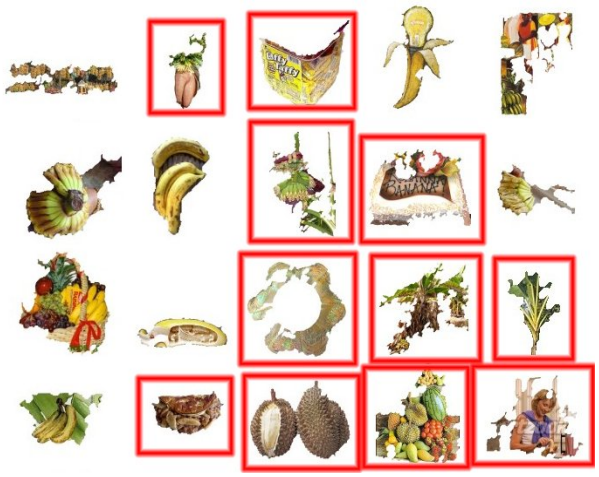


FIG. 5.30 – Les 20 premières segmentations pour la requête *banana fruit*. Les images non pertinentes sont encadrées en rouge. Il n'y avait pas suffisamment de bananes jaunes dans les images pour permettre à l'algorithme d'identifier le jaune comme étant la principale couleur. Cela arrive avec les requêtes qui retournent trop d'images non pertinentes dans les moteurs de recherche d'images sur Internet.

Afin d'évaluer la qualité des segmentations, nous avons segmenté manuellement quelques images et nous avons comparé ces segmentations manuelles avec nos segmentations automatiques. Les segmentations manuelles ont été faites avec SAIST (*Semi-Automatic Image Segmentation Tool*), un logiciel développé par le laboratoire PRIP à Vienne [59]. Il permet de calculer une segmentation par ligne de partage des eaux à partir de marqueurs définis par l'utilisateur.

Pour la vérité terrain, nous avons sélectionné tous les pixels appartenant à l'objet recherché comme étant de l'objet, et les autres comme étant du fond. C'est-à-dire que si deux objets sont présents dans l'image, ils seront tous deux segmentés, même si notre algorithme de segmentation automatique a été conçu pour ne garder idéalement que le plus grand des deux. Si un objet est masqué par un autre, par exemple un cheval dont une partie est masquée par la selle, l'objet masquant est considéré comme du fond, altérant ainsi la forme de l'objet masqué, par rapport à sa forme réelle.

Étant donné qu'il serait trop long de segmenter manuellement les 300 images que nous avons récupérées par requête, nous n'évaluons la segmentation que sur les images pertinentes parmi les 20 meilleures sélectionnées par notre algorithme de reclassification.

Il y a deux mesures à considérer pour évaluer la segmentation d'un objet : la proportion des pixels de l'objet qui sont correctement retrouvés M_1 (rappel de la segmentation), et la proportion des pixels retenus par la segmentation automatique qui sont corrects M_2 (précision de la segmentation). Soit S_A la région identifiée comme étant l'objet par notre algorithme, et S_T la vérité terrain. M_1 et M_2 s'écrivent :

$$M_1 = \frac{surface(S_A \cap S_T)}{surface(S_T)}$$

animals	Yahoo !	Re-ranking	**man-made**	Yahoo !	Re-ranking
bald eagle	100	100	black shirt	90	90
Bengal tiger	100	100	blue mug	60	80
bull	70	75	Boeing 777	55	90
cerastes	75	80	Eiffel tower	100	80
common dolphin	90	90	fire engine	80	85
common zebra	95	100	red bottle	60	65
dromedary	70	80	sun glasses	90	85
ewe	85	100	white Porsche	65	95
German shepherd	95	100	wood table	95	80
monarch butterfly	95	100	yellow Ferrari	90	95
average animals	87.5	92.5	*average man-made*	78.5	84.5

TAB. 5.6 – Comparaison de la précision (en %) calculée sur les 20 premières images retournées par notre algorithme et par Yahoo ! image search, qui nous procure les images originales. Vingt requêtes sont considérées : dix animaux et dix objets artificiels. Nous observons une augmentation de précision d'environ 5% alors que la qualité des images est également améliorée, mais cela n'apparaît pas dans ce tableau.

$$M_2 = \frac{surface(S_A \cap S_T)}{surface(S_A)}$$

Les deux mesures sont fortement liées. Nous pouvons facilement avoir $M_1 = 100\%$ en considérant l'image entière comme résultat de la segmentation, mais dans ce cas, M_2 vaut (seulement) le rapport de la surface de l'objet divisée par la surface de l'image. Ces deux mesures n'ont donc pas de sens séparément et doivent être représentées ensemble. Leurs valeurs pour nos 20 requêtes de tests sont données sur la figure 5.31.

Nous constatons que, à part pour quelques requêtes, nous pouvons nous attendre en moyenne à un taux de pixels corrects dans la segmentation automatique (M_2) compris entre 70% et 90%, tout en retrouvant entre 70% et 90% des objets (M_1).

Décider s'il est préférable d'avoir plutôt une grande valeur de M_1 ou de M_2 dépend de l'application considérée. Si, comme dans notre cas, l'application visée est la consitution de bases d'apprentissage, alors il est préférable de minimiser le bruit et donc de maximiser M_2. Afin d'évaluer les segmentations, nous avons décidé d'utiliser une mesure qui prend en compte à la fois le taux de pixels retrouvés et la proportion de pixels qui sont corrects. Cette mesure, souvent utilisée pour évaluer les algorithmes de segmentation, est la suivante :

$$M_S = \frac{surface(S_A \cap S_T)}{surface(S_A \cup S_T)}$$

Nous la comparons dans la figure 5.32 avec le score que nous obtiendrions si nous gardions l'image entière comme résultat de segmentation, qui représente également la taille des objets dans l'image (en %).

Les résultats obtenus sont bons : la mesure passe d'une moyenne d'environ 30% pour l'image entière à environ 60% avec notre algorithme. Cela marche notamment bien avec

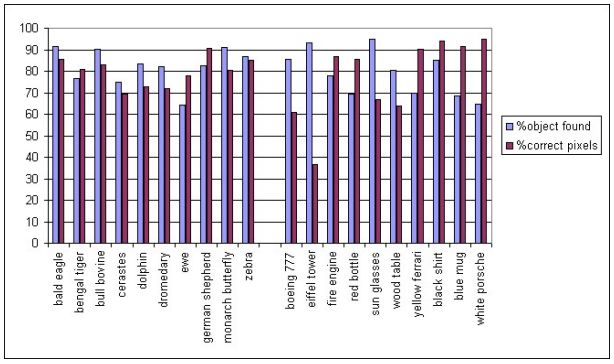


FIG. 5.31 – Évaluation des résultats de la segmentation montrant le pourcentage de pixels d'objets qui sont correctement retrouvés et le pourcentage de pixels qui sont corrects dans les résultats de la segmentation automatique de 20 requêtes : 10 animaux et 10 objets fabriqués par l'homme.

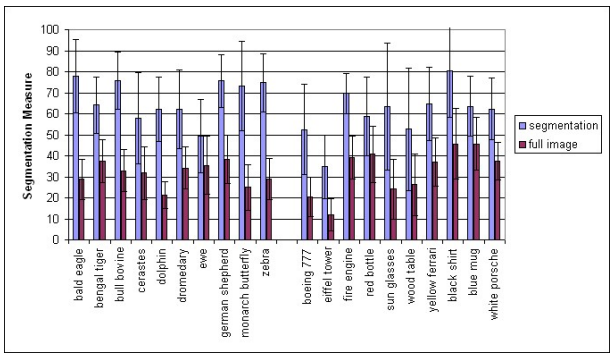


FIG. 5.32 – Évaluation de la précision de la segmentation sur vingt requêtes, avec affichage des écarts types.

des objets à deux couleurs, dans le cas où ce sont des rayures (*zebra*, *Bengal tiger*), des taches (*monarch butterfly*) ou deux couleurs clairement séparées (textitbald eagle, *German shepherd*). La tâche est plus difficile pour des objets ayant la même couleur que leur environnement (*cerastes*, *dolphin*, *dromedary*). Nous obtenons de meilleurs résultats en général pour les animaux que pour les objets artificiels. Cela est dû principalement au fait que les animaux tendent à avoir moins de variations en termes de couleur que les objets fabriqués par l'homme, les rendant ainsi plus facile à identifier en cherchant une similarité entre toutes les images.

La requête *Eiffel tower* est celle obtenant le moins bon résultat. C'est aussi la requête pour laquelle les objets ont la taille la plus petite. Si nous comparons ce résultat avec celui de la figure 5.31, nous comprenons que les résultats de segmentation contiennent toujours la majeure partie de l'objet, mais l'objet n'occupe que 40% de la région segmentée, ce qui veut dire que les 60% restant sont du fond. La principale raison expliquant les mauvaises performances pour cette requête est que notre algorithme de reclassement est conçu pour

favoriser les objets dont la taille est comprise entre 20% and 60% de la taille de l'image, alors que la tour Eiffel est un objet fin occupant en moyenne 10% de l'image selon la figure 5.32.

5.5 Conclusion sur la construction automatique de bases d'apprentissage

Nous avons décrit dans ce chapitre nos efforts pour traiter automatiquement les images du web afin de produire des ensembles d'images « propres » simplement à partir de noms de concepts. Afin de rendre ces ensembles propres, nous avons introduit plusieurs innovations :

- la requête textuelle pour collecter les images a été soigneusement formulée en utilisant notamment WordNet pour augmenter la pertinence des résultats bruts d'images retournés par les moteurs de recherche,
- les cliparts ont été enlevés de ces images pour ne conserver que les photographies,
- les objets ont été automatiquement localisés et segmentés dans les images
- ce qui a permis de reclasser les images dans le but d'obtenir en premier les meilleurs représentants des concepts recherchés.

Divers scénarios ont été envisagés pour la segmentation automatique. Notamment, l'idée d'avoir recours aux noms des couleurs est intéressante car elle permet d'extraire une information dans un corpus textuel qui sera utilisée pour le traitement d'images. Néanmoins, une « simple » quantification de l'espace RVB en 125 couleurs donne de meilleurs résultats en segmentation, principalement à cause de l'augmentation du nombre de couleurs.

Il nous reste à présent à évaluer dans quelle mesure les bases d'images ainsi constituées sont exploitables par un algorithme d'apprentissage afin de faire de la reconnaissance d'objets. Nous comparerons dans le chapitre suivant, avec le même algorithme de reconnaissance, les résultats obtenus avec les bases constituées automatiquement par rapport à une base qui serait construite et segmentée manuellement.

Chapitre 6

Reconnaissance d'un objet dans une région de l'image

Looking out of my window this lovely spring morning I see an azalea in full bloom. No, no! I do not see that; though that is the only way I can describe what I see. That is a proposition, a sentence, a fact; but what I perceive is not proposition, sentence, fact, but only an image which I make intelligible in part by means of a statement of fact. This statement is abstract; but what I see is concrete.
Charles Sanders Pierce, manuscript 692

La reconnaissance d'un objet dans une image peut être vue comme constituée de deux étapes : l'apprentissage d'un modèle pour chaque objet, et la localisation de la région correspondant au mieux à ce modèle dans une nouvelle image. L'étape d'apprentissage utilise des bases d'images d'exemples du ou des objets que l'on souhaite savoir reconnaître. Nous utiliserons à la fois, dans ce chapitre, des bases construites automatiquement, suivant le procédé développé au chapitre précédent, et des bases construites manuellement, c'est-à-dire dont les images sont sélectionnées et éventuellement segmentées à la main.

Nous faisons l'apprentissage des objets avec un procédé similaire à l'apprentissage qui a été détaillé pour les scènes (chapitre 4) : à partir d'images représentant les objets, plusieurs descripteurs sont calculés et mis bout à bout pour permettre un apprentissage multiclasse avec un séparateur à vaste marge. La classification multiclasse est rendue possible avec la technique « un contre un » qui calcule des probabilités d'appartenance multiclasse à partir de l'ensemble des SVMs binaires correspondant à chaque couple de

classes possible. Nous choisissons toujours dans nos expériences un noyau gaussien dont les paramètres sont estimés par validation croisée (4/5 pour l'apprentissage et 1/5 pour le test, répété 5 fois).

Lorsque les images d'apprentissage sont des régions segmentées dont les formes ne sont pas rectangulaires, la plupart des descripteurs sont plus difficile à calculer. Nous avons étendu certains de nos descripteurs pour prendre en compte ces cas, mais le descripteur de Gabor, par exemple, n'a été implémenté que pour des images rectangulaires.

Deux types de segmentations sont utilisés : les segmentations manuelles et les segmentations automatiques. Les segmentations manuelles servent à avoir une vérité terrain, notamment pour évaluer les algorithmes de segmentation automatique, comme nous l'avons fait au chapitre précédent, ou pour comparer l'apprentissage à partir de données segmentées automatiquement par rapport à l'apprentissage fait à partir de données segmentées manuellement. Les segmentations manuelles sont faites grâce à l'outil d'annotation semi-automatique SAIST [59] que nous avons présenté en section 5.4.3.

Pour la segmentation automatique, nous utilisons principalement la segmentation morphologique en cascade [89] qui calcule d'abord une segmentation par ligne de partage des eaux sur l'image, puis fusionne les régions de manière hiérarchique en fonction du gradient séparant ces régions.

6.1 Expérience 1 : reconnaissance de 14 animaux segmentés manuellement

Cette expérience a été mise en œuvre lors d'un séjour de deux semaines au laboratoire PRIP à Vienne. Nous avons développé, dans le cadre du réseau d'excellence MUSCLE, un ensemble de 1292 images segmentées manuellement qui sont un sous-ensemble de la base Corel. Nous voulions effectuer des expériences simples sur la reconnaissance de ces objets pour avoir des résultats de base en classification afin de pouvoir comparer les performances futures.

Deux expériences distinctes ont été faites sur ces données. La première utilise seulement les segmentations manuelles des animaux et les divise en un ensemble d'apprentissage et un ensemble de test, afin d'évaluer la qualité de la reconnaissance indépendamment de la qualité de la segmentation. La deuxième expérience est plus difficile : étant donné une image contenant un ou plusieurs animaux d'une même classe, nous utilisons une segmentation automatique et un algorithme de classification entraîné sur les segmentations manuelles pour localiser et reconnaître le ou les animaux présents dans l'image.

6.1.1 Descripteurs et apprentissage

Les deux descripteurs simples suivants ont été utilisés :

- histogramme couleur : un histogramme RVB à 64 composantes où les plans R, V et B sont quantifiés en 4 valeurs chacun.
- histogramme de texture : les *local binary patterns* (LBP) [106] pour trois rayons différents sont combinés : R=1 (10 composantes), R=2 (18 composantes) and R=3

animal	apprentissage	évaluation	nombre total
cheetah	4300 à 4327	4328 à 4333	34
cougar	4400 à 4479	4480 à 4499	100
coyote	5900 à 5979	5980 à 5999	100
deer	5300 à 5368	5369 à 5385	86
dog	900 à 979 17400 à 17479	980 à 999 17480 à 17499	200
elephant	5600 à 5679	5680 à 5699	100
goat	4600 à 4679	4680 à 4699	100
hippopotamus	6100 à 6132	6133 à 6140	41
horse	6200 à 6279 8700 à 8779	6280 à 6299 8780 à 8799	200
leopard	4334 à 4385	4386 à 4399	66
lion	5400 à 5479	5480 à 5499	100
moose	5386 à 5396	5397 à 5399	14
rhinoceros	6141 à 6187	6188 à 6199	59
tiger	5700 à 5779	5780 à 5799	100

TAB. 6.1 – Images utilisées pour l'apprentissage et l'évaluation

(26 composantes).

L'apprentissage a été fait avec des séparateurs à vaste marge probabilistes, en utilisant la méthode un-contre-un pour pouvoir faire de la classification multiclasse. Les paramètres du noyau gaussien utilisé ont été estimés par validation croisée. Les 14 animaux suivants ont été appris : *cheetah, cougar, coyote, deer, dog, elephant, goat, hippopotamus, horse, leopard, lion, moose, rhinoceros, tiger* ainsi qu'une quinzième classe contenant les fonds, c'est-à-dire toute région qui n'est pas un animal.

Les segmentations manuelles ont été divisées en environ 80% pour l'apprentissage et 20% pour l'évaluation. Les images choisies pour chaque ensemble sont données dans le tableau 6.1.

Étant donné que nous avons séparé, lors de la segmentation manuelle, les animaux apparaissant dans une même image, dans le cas où il y en a plus d'un, le nombre de régions segmentées pour les animaux est différent du nombre d'images (cf. tableau 6.2).

6.1.2 Évaluation des descripteurs pour la reconnaissance

Nous comparons les résultats obtenus en utilisant pour l'apprentissage des SVMs soit l'histogramme de couleur RVB, soit l'histogramme de texture LBP ou bien encore les deux. Les résultats sont montrés sur le tableau 6.3. Étant donné que la reconnaissance des fonds semble être plus facile que la classification des animaux, et que le nombre d'images de fonds est bien supérieur à celui de chaque classe d'animal, nous avons mis la classe de fonds à part dans ce tableau.

	cheetah	cougar	coyote	deer	dog	elephant	goat	hippopotamus	horse	leopard	lion	moose	rhinoceros	tiger	background
Apprentissage	48	81	115	80	203	196	119	58	314	53	87	12	54	80	1032
Évaluation	8	20	20	22	44	29	37	9	78	14	21	3	12	20	260

TAB. 6.2 – Nombre de régions pour chaque classe correspondant à la répartition proposée dans le tableau 6.1.

descripteurs	% de bonnes classifications		
	total	fond	animal
RVB64	63,0%	90%	42,1%
LBP	71,9%	99,2%	50,7%
RVB64+LBP	78,1%	99,2%	61,7%

TAB. 6.3 – Résultats de la classification. La quatrième colonne correspond à l'attribution correcte de la classe d'un animal donné, parmi les 14 classes d'animaux : ce n'est pas une classification binaire fond/animal, mais bien une classification en 14 classes d'animaux et une classe de fonds.

Nous remarquons, comme nous l'avons déjà vu dans le chapitre sur la classification de scènes, que l'utilisation conjointe des descripteurs de couleur et de texture améliore les résultats. Ce qui est en revanche surprenant, c'est que le descripteur de texture seul donne de meilleurs résultats que la couleur, alors que le nombre de composantes est à peu près le même (64 pour la couleur et 54 pour la texture) et que nous pensions que la couleur serait plus discriminante pour le cas particulier des animaux.

La matrice de confusion est représentée sur le tableau 6.4 pour le meilleur classifieur combinant la couleur et la texture.

Nous obtenons un très bon taux de bonne classification de 99,2% pour la classe de fond. Cela signifie que, même si nos descripteurs sont simples, nous pouvons construire un classifieur binaire efficace pour faire la distinction entre animal et fond. Le modèle multiclasse appris ci-dessus et appliqué à cette classification binaire donne un taux moyen de bonne classification de 97,5%.

Quelques erreurs de classification peuvent être expliquées. Les rhinocéros sont presque tous classés dans la classe éléphant : ces deux animaux ont des couleurs et des textures très proches et étant donné qu'il y a plus d'images d'éléphants pour l'apprentissage, l'algorithme de classification donne la priorité à cette classe. Une solution pour améliorer la classification serait d'utiliser une classification à partir des points d'intérêt qui pourrait se concentrer sur les parties locales des animaux qui permettent de faire la différence entre les rhinocéros et les éléphants. Une autre classe obtenant 0% de classification correcte est la classe des élans (*moose*), principalement à cause du manque d'images d'exemples

	Requête														
classé comme ↓	cheetah	cougar	coyote	deer	dog	elephant	goat	hippopotamus	horse	leopard	lion	moose	rhinoceros	tiger	background
cheetah	**6**	0	0	0	0	0	0	0	0	1	0	0	0	0	0
cougar	0	**10**	0	0	3	0	1	0	1	0	0	0	0	1	0
coyote	0	0	**11**	1	3	1	4	1	1	0	0	0	0	0	0
deer	0	0	1	**5**	0	2	4	0	0	0	0	0	0	0	0
dog	0	6	1	2	**30**	1	6	1	4	0	1	1	0	1	1
elephant	0	0	0	2	0	**21**	0	0	3	0	3	1	10	0	0
goat	0	0	4	6	0	1	**16**	0	0	0	0	0	0	1	0
hippopotamus	0	0	0	0	1	1	0	**3**	0	0	0	1	0	0	0
horse	0	0	2	5	3	2	4	3	**64**	0	0	0	1	1	1
leopard	0	0	0	0	0	0	0	0	0	**13**	0	0	0	0	0
lion	1	4	0	0	2	0	0	1	2	0	**16**	0	0	0	0
moose	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0
rhinoceros	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0
tiger	0	0	0	1	0	0	0	0	1	0	0	0	0	**13**	0
background	1	0	1	0	2	0	2	0	2	0	1	0	1	3	**258**

TAB. 6.4 – Matrice de confusion pour la classification des régions.

pour l'apprentissage de cet animal.

Nous remarquons également que beaucoup d'animaux différents sont classifiés dans la classe chien (*dog*). En effet, la classe des chiens contient des images correspondant à différentes espèces de chien ce qui donne à cette classe une grande diversité en termes de couleur et de texture et la rend proche de nombreux autres animaux dans l'espace couleur/texture étudié. Il serait plus judicieux, pour la reconnaissance d'images, de séparer les espèces de chiens en plusieurs classes.

6.1.3 Évaluation de la segmentation automatique

Nous avons évalué des segmentations en cascade (une ligne de partage des eaux hiérarchique) faites par le CMM [89]. Pour chaque image, de 2 à 4 niveaux de segmentation sont disponibles. Nous allons étudier et comparer deux méthodes pour évaluer dans quelle mesure cette segmentation automatique nous permet de localiser et reconnaître un animal dans une image.

La première évaluation utilise l'hypothèse que l'animal recherché est contenu dans une seule région produite par l'algorithme de segmentation, dans l'un des différents niveaux de segmentation. Afin de localiser et reconnaître l'animal dans l'image, nous

considérons tous les niveaux de la segmentation pour cette image. Les descripteurs sont calculés sur chaque région de chaque niveau afin de la classer dans l'une des 15 classes, à l'exception des petites régions (d'une surface inférieure à 1% de celle de l'image) qui sont ignorées. Comme les SVMs appris sont probabilistes, nous pouvons identifier la région pour laquelle la probabilité de reconnaissance est la plus grande et la classe reconnue est un animal (pas un fond). Nous retenons la classe correspondante comme la classe de l'animal contenu dans l'image. Les résultats pour les images utilisées dans l'apprentissage et les autres sont considérés séparément dans le tableau 6.5.

	images d'apprentissage	images de test
cheetah	82,1%	66,7%
cougar	40,0%	35,0%
coyote	51,3%	40,0%
deer	7,2%	0%
dog	62,5%	37,5%
elephant	71,3%	55,0%
goat	33,8%	20,0%
hippopotamus	42,4%	12,5%
horse	85,0%	65,0%
leopard	73,1%	85,7%
lion	27,5%	20,0%
moose	0%	0%
rhinoceros	19,1%	0%
tiger	51,3%	40,0%
tous	52,4%	38,5%

TAB. 6.5 – Pourcentage de bonnes classifications dans le cas où l'animal assigné à une image est celui pour lequel la probabilité de détection est la plus grande, parmi les différentes probabilités obtenues dans chacune des régions de la segmentation automatique, considérant tous les niveaux de segmentations.

Ces pourcentages ne reflètent pas tout à fait les résultats réels, car ils ne tiennent pas compte de la qualité de la localisation de l'animal trouvé. Il peut arriver en effet que l'animal détecté dans l'image soit correct, mais ne soit pas au bon endroit. Il est alors néanmoins compté comme une bonne reconnaissance. Inversement, si l'animal est correctement localisé, mais mal classé, il sera compté comme une erreur. Nous n'évaluons donc pas ici la localisation de l'animal mais plutôt la capacité à reconnaître quel animal est dans l'image. Cependant, bonne localisation et bonne reconnaissance sont fortement liées : dans la plupart des cas, les animaux correctement localisés sont correctement reconnus, et ceux qui sont mal localisés (un animal est vu dans le fond) sont mal reconnus.

La seconde méthode d'évaluation consiste à chercher l'animal reconnu sur la plus grande surface au lieu de l'animal reconnu avec la plus forte probabilité. À chaque région est assignée la classe de plus forte probabilité parmi les 14 animaux ou le fond.

	images d'apprentissage	images de test
cheetah	89,3%	83,3%
cougar	62,3%	36,8%
coyote	51,4%	25,0%
deer	13,6%	0%
dog	67,1%	67,5%
elephant	88,9%	60,0%
goat	59,2%	26,3%
hippopotamus	63,6%	12,5%
horse	90,6%	75,7%
leopard	73,9%	100%
lion	37,2%	15,8%
moose	0%	0%
rhinoceros	32,6%	0%
tiger	51,3%	44,4%
tous	65,8%	45,4%

TAB. 6.6 – Pourcentage de bonnes classifications dans le cas où l'animal assigné à une image est celui pour lequel la surface est la plus grande, en considérant la somme des surfaces de chacune des régions de la segmentation automatique où cet animal est reconnu, pour un seul niveau de segmentation.

Les surfaces des régions pour lesquelles un même animal est reconnu sont additionnées et l'animal obtenant la plus grande surface (en ignorant la classe de fond qui a souvent la surface la plus grande) est conservé et sa localisation est constituée de l'ensemble des régions où cet animal a été reconnu. Dans le cas où toutes les régions sont reconnues comme étant du fond, l'image est considérée comme ne contenant aucun animal.

En comparaison avec la première méthode d'évaluation décrite ci-dessus, celle-ci essaie de fusionner plusieurs régions, et serait par exemple capable de localiser en même temps deux animaux de la même espèce présents dans une même image, ou de reconstituer un animal segmenté en plusieurs parties. En revanche, le fait de fusionner rend beaucoup plus difficile l'utilisation de plusieurs niveaux de segmentation. De plus, la deuxième méthode est dépendante de la taille de l'animal dans l'image : un animal occupant une plus grande surface dans l'image aura plus de chances d'être correctement reconnu, ce qui n'était pas le cas de l'autre méthode. Nous avons choisi en pratique de n'utiliser, pour la deuxième méthode, qu'un seul niveau de segmentation qui est le niveau contenant le moins de régions, parmi ceux contenant au moins 10 régions.

Les résultats sont affichés dans le tableau 6.6.

Nous constatons que, même si nous ne considérons qu'un niveau de segmentation, les résultats sont meilleurs que ceux obtenus avec la première méthode (tableau 6.5).

6.2 Expérience 2 : Comparaison des constructions manuelle et automatique de bases d'apprentissage

Cette expérience a pour but de comparer les résultats obtenus avec une base construite automatiquement, c'est-à-dire où les images sont segmentées et sélectionnées automatiquement, avec ceux obtenus en apprenant à partir d'une base construite manuellement. A priori, l'apprentissage avec une base construite automatiquement est plus difficile qu'avec une constituée manuellement, pour deux raisons :

- les images ne sont pas segmentées exactement : il est possible qu'une large partie du fond soit présent dans la segmentation, et/ou qu'une certaine proportion de l'objet soit manquante,
- certaines images sont des intrus et ne correspondent pas au concept qui est appris.

Pour construire la base d'apprentissage automatique, les images sont collectées depuis Internet, segmentées puis reclassées et les 20 premières sont conservées suivant le procédé décrit dans le chapitre précédent. Afin de pouvoir comparer les résultats, la base manuelle est constituée à partir des mêmes images que la base automatique : parmi les 20 images conservées dans la base automatique pour chaque concept, nous ne gardons que celles qui sont pertinentes, que nous segmentons manuellement à l'aide de l'outil SAIST.

Dans cette expérience, 20 concepts ont été appris et testés : 10 objets fabriqués par l'homme et 10 animaux. Ce sont les concepts qui ont servi à évaluer le filtrage et la segmentation automatique d'images collectées sur Internet au chapitre précédent. Nous avons donc 400 images d'objets pour la base automatique et 361 pour la base manuelle. En complément, nous avons rajouté une classe « fonds » contenant 42 images permettant d'éviter quelques fausses détections.

6.2.1 Détection d'un objet dans une image

Dans le problème traité dans cette expérience, il faut être capable à la fois de reconnaître et de localiser les objets. Nous voulions initialement utiliser le même procédé que dans l'expérience précédente pour localiser les objets : faire d'abord une segmentation morphologique de l'image, puis classer chaque segment selon les modèles SVMs appris. Cependant, la base d'images utilisée dans cette expérience est volontairement plus difficile que la précédente en ce qui concerne la segmentation. Notamment, les animaux à rayures tels que *zebra* ou *Bengal tiger* sont en général séparés en plusieurs segments lors d'une segmentation automatique. *Bald eagle* possède un corps noir et une tête blanche et aura également tendance à être séparé en deux régions. Nous avons donc choisi d'utiliser des fenêtres glissantes de différentes échelles afin de pallier ce type de problème.

Trois tailles de fenêtres glissantes sont considérées :

- (a) toute l'image,
- (b) 50% de l'image en largeur et en hauteur,
- (c) 25% de l'image en largeur et en hauteur.

Plusieurs taux de recouvrements ont été brièvement comparés, et un taux de 50% nous a semblé être un bon compromis : pour un taux plus grand, les performances sont à

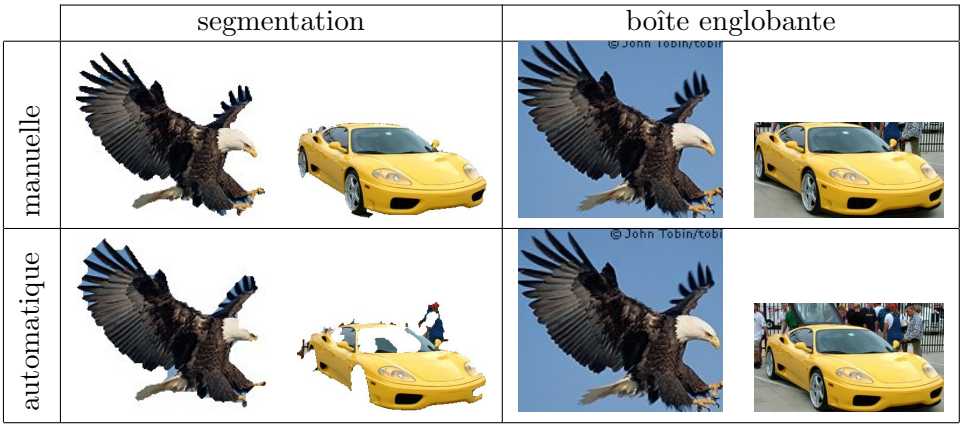


FIG. 6.1 – Exemple des quatre versions possibles de deux images qui appartiennent aux différentes bases d'apprentissage considérées.

peu près les mêmes, mais le temps de calcul augmentait proportionnellement au nombre de fenêtres. Un recouvrement de 50% correspond à 3×3 fenêtres de type (b) et 7×7 fenêtres de type (c).

L'utilisation de fenêtres glissantes rend possible la présence de contexte autour de l'objet dans la fenêtre à classer. Nous avons donc essayé, pour l'apprentissage, d'utiliser uniquement l'objet segmenté comme précédemment, ou d'utiliser la boîte englobante de cet objet, c'est-à-dire incluant une partie du fond (sauf dans le cas exceptionnel où l'objet serait précisément rectangulaire). Les quatres bases différentes ainsi générées sont illustrées sur la figure 6.1.

Pour une échelle donnée λ et une fenêtre $W_{i,\lambda}$ parmi les N_λ fenêtres de cette échelle, les descripteurs sont calculés sur cette fenêtre, et la probabilité $P(O|\lambda, W_{i,\lambda})$ de détecter l'objet O dans la fenêtre $W_{i,\lambda}$ est la probabilité retournée par le SVM. Nous choisissons de donner la même importance aux N_λ fenêtres d'une échelle λ donnée :

$$P(W_{i,\lambda}) = \frac{1}{N_\lambda}$$

ce qui nous permet de calculer la probabilité $P(O|\lambda)$ de détecter l'objet O à l'échelle λ comme :

$$P(O|\lambda) = \sum_{i=1}^{N_\lambda} \frac{P(O|\lambda, W_{i,\lambda})}{N_\lambda}$$

puis la probabilité $P(O)$ de détecter l'objet dans l'image par :

$$P(O) = \sum_{\lambda=1}^{3} \frac{P(O|\lambda)}{3}$$

où nous considérons que chacune des trois échelles a la même importance : $P(\lambda) = 1/3$

Nous avons testé plusieurs combinaisons de nos descripteurs, et avons obtenu les meilleurs résultats avec la combinaison d'un descripteur de couleur (RVB64) et d'un descripteur de texture (LEP). L'ajout d'un autre descripteur dégrade les performances, contrairement à ce qui avait été observé pour les classifications de scènes. Nous pensons que cela est dû au faible nombre d'images d'exemples qui entraînent un sous-apprentissage lorsque le vecteur des descripteurs est trop grand. Notons toutefois qu'il y a moins de descripteurs disponibles lorsque les images considérées ne sont pas de forme rectangulaire.

6.2.2 Résultats

Nous sélectionnons comme images pour la phase d'évaluation les 10 premières images pertinentes de chaque classe parmi les images retournées par Internet, et qui ne sont pas dans notre base d'apprentissage. La base de test est donc constituée de 200 images, non segmentées, et il faut donc à la fois trouver la position de l'objet dans l'image ainsi que le reconnaître.

Afin d'évaluer les différents choix méthodologiques que nous avons présentés ci-dessus, nous donnons quelques résultats dans les tableaux 6.7 et 6.8 en utilisant la base manuelle. Nous considérons la base manuelle comme constituée à la fois les images segmentées, et les boîtes englobantes de ces images (incluant une partie du contexte). L'apprentissage en utilisant ces deux données à la fois est en effet meilleur que l'apprentissage n'utilisant qu'un des deux comme nous le verrons par la suite.

Nous commençons ici par évaluer l'intérêt de la combinaison des descripteurs de texture et de couleur, ainsi que la manière de prendre en compte les différentes informations obtenues pour chaque fenêtre de chaque échelle. Le tableau 6.7 montre les résultats obtenus si l'on considère la probabilité que l'objet O soit contenu dans l'image comme $max_{i,\lambda} P(O|W_{i,\lambda})$. L'objet retenu pour l'image est alors celui de plus forte probabilité.

	(a)	(b)	(c)	(a)+(b)+(c)
RVB64	25,5%	33%	33,5%	**34,5%**
LEP	37%	**42%**	32%	39,5%
LEP+RVB64	42,5%	**54%**	46%	50%

TAB. 6.7 – Pourcentage de bonnes détections pour différentes échelles en considérant $max_{i,\lambda} P(O|W_{i,\lambda})$ comme étant la probabilité de détection d'un objet dans l'image.

Ces résultats sont à comparer à ceux du tableau 6.8 où $P(O)$ est la probabilité définie précédemment qui pondère les contributions de chaque fenêtre. Nous constatons que les résultats sont légèrement meilleurs avec cette pondération, sauf pour RVB64. Cette pondération présente notamment l'avantage de diminuer l'influence d'une fausse détection qui aurait un pourcentage très élevé, ce qui arrive assez souvent sur les petites fenêtres.

Nous voyons tout d'abord que la combinaison des deux descripteurs donne toujours

	(a)	(b)	(c)	(a)+(b)+(c)
RVB64	25,5%	**32,5%**	30%	30,5%
LEP	37%	**44,5%**	37,5%	40,5%
LEP+RVB64	42,5%	**55,5%**	50,5%	50%

TAB. 6.8 – Pourcentage de bonnes détections pour différentes échelles en considérant $P(O|\lambda)$ et $P(O)$ comme décrits précédemment.

de meilleurs résultats que l'utilisation d'un seul. En ce qui concerne les échelles, l'échelle (b) donne les meilleurs résultats, ce qui signifie que la plupart des objets ont une largeur et une hauteur d'environ la moitié de celles de l'image, correspondant à une surface de 25% de celle de l'image. L'échelle (a), qui est équivalente à considérer le problème comme un problème de classification de scène (puisque toute l'image est considérée sans segmentation), donne les moins bonnes performances. La combinaison des trois échelles donne des performances inférieures à l'échelle (b) uniquement, mais supérieures aux deux autres échelles. Nous pourrions envisager de donner un poids plus important à l'échelle (b) lors de la combinaison en adaptant $P(\lambda)$, ce qui supposerait que nous disposions d'une connaissance a priori de la taille de l'objet dans les images. Nous n'avons pas souhaité faire cette hypothèse, car il devient alors difficile de trouver une justification théorique aux différents poids donnés à chaque échelle.

En conservant la méthode qui donne les meilleurs résultats, nous étudions maintenant l'effet de la base d'apprentissage utilisée sur la qualité de la détection. Deux paramètres peuvent changer pour cette base. Premièrement, elle peut être créée automatiquement ou manuellement. La base créée automatiquement peut contenir des images non pertinentes et les segmentations des objets dans ces images sont automatiques. La base créée manuellement utilise les mêmes images, où les images non pertinentes sont filtrées à la main, et la segmentation des objets est faite à la main. Deuxièmement, nous pouvons considérer pour l'apprentissage les images segmentées, c'est-à-dire les objets détourés ou la boîte englobante de ces objets, incluant une partie du contexte. Il est également envisageable de mélanger ces deux types d'images. L'idée d'utiliser la boîte englobante vient de ce que la détection est faite avec des fenêtres glissantes rectangulaires qui ne cherchent pas à détourer l'objet, et qui auront donc tendance à contenir une partie du contexte. Les résultats pour ces 6 différentes bases d'apprentissage sont donnés dans le tableau 6.9.

LEP+RVB64	manuelle	automatique
images segmentées	43%	34%
boîtes englobantes	49,5%	50%
les deux ensembles	55,5%	51,5%

TAB. 6.9 – Comparaison des différents taux de bonnes détections obtenus en fonction de la base d'apprentissage utilisée.

Nous remarquons d'abord que les résultats avec la base automatique sont, comme attendus, inférieurs à ceux obtenus avec la base manuelle (légèrement supérieurs dans le cas des boîtes englobantes). Ils sont cependant proches dans les deux derniers cas du tableau 6.9, ce qui montre que la segmentation et le filtrage automatiques des images est efficace et permet de construire une base dont la qualité approche celle d'une base manuellement construite.

Comme nous nous y attendions, les résultats sont meilleurs avec les boîtes englobantes qu'avec les images segmentées. En effet, comme nous l'avons expliqué précédemment, les fenêtre glissantes utilisées pour localiser l'objet auront tendance, comme les boîtes englobantes, à contenir du contexte autour de l'objet. Les résultats sont encore meilleurs en combinant les deux types d'images. Nous pensons que cela est dû notamment aux cas où la fenêtre glissante est entièrement comprise à l'intérieur d'un objet. Une telle sous-image sera mieux classée si des images sans contexte sont présentes dans la base d'apprentissage, ce qui est le cas des images segmentées.

Enfin, une note sur le temps de calcul : typiquement, pour une image d'une taille de l'ordre de 250×250, il faut environ 2 secondes pour parcourir les 59 fenêtres aux 3 échelles.

6.3 Conclusion sur la reconnaissance d'objets

Nous avons présenté et évalué dans ce chapitre les différentes méthodes que nous avons mises en œuvre au cours de cette thèse pour reconnaître un objet dans une image. Nous avons montré que les bases d'apprentissage automatiquement créées suivant le principe développé au chapitre précédent donnent des résultats en classification très proches de ceux obtenus avec des bases constituées manuellement. En ce qui concerne les résultats en eux-mêmes, nous avons obtenu au mieux 55,5% pour la reconnaissance de 20 concepts. Cela est faible par rapport aux performances humaines. Notons toutefois qu'il s'agit là d'une application difficile où d'une part l'objet doit à la fois être reconnu et localisé dans l'image, et d'autre part la plupart de ces 20 concepts sont composés de différentes couleurs et/ou textures, ce qui les rend difficiles à segmenter avec les algorithmes traditionnels de segmentation.

Nous sommes conscients que nous n'utilisons pas les meilleurs algorithmes de la littérature. Nous n'utilisons que deux descripteurs, et nous n'avons notamment pas eu le temps de faire des tests avec les méthodes de sacs de mots s'appuyant sur les points d'intérêts et qui sont actuellement parmi les meilleures méthodes en reconnaissance d'objets. Cependant, même si nous avons passé une bonne partie du temps à améliorer ces algorithmes, les travaux originaux de cette thèse se situent plutôt en amont et en aval de cette partie, c'est-à-dire en ce qui concerne la création automatique de bases d'apprentissage, la classification hiérarchique de scènes et les différentes possibilités de désambiguïsation que nous détaillerons dans le chapitre suivant.

Chapitre 7

Désambiguïsation de régions

La conscience n'est jamais assurée de surmonter l'ambiguïté et l'incertitude.
Edgar Morin, Le paradigme perdu

La désambiguïsation est un champ d'étude très actif en analyse de textes. Il s'agit, pour les mots polysémiques, de déterminer quel sens de ce mot est utilisé dans une phrase donnée, parmi tous les sens possibles du mot. Par exemple, le mot « plage » n'a pas le même sens dans « plage de sable » et « plage de valeur ». Cette désambiguïsation est faite en considérant le contexte du mot, c'est-à-dire les mots qui se situent dans la même phrase ou dans le même document, la catégorie du document (scientifique, politique), etc. Cette désambiguïsation est très importante par exemple pour être capable de comprendre le sens du texte, ainsi que pour faire de la traduction automatique.

Cette ambiguïté existe également en analyse d'images, même si elle est beaucoup moins étudiée. Dans beaucoup de publications, il est considéré comme une hypothèse de base que chaque objet dans les images a une représentation unique en termes de couleur, forme et texture, et qu'un apprentissage multiclasse sur les régions peut donc permettre de classer ces objets idéalement sans erreur. En réalité, certains objets peuvent apparaître identiques en couleur, texture et forme si l'on considère uniquement la région segmentée dans l'image, mais être perçus différemment lorsque l'on regarde l'image dans sa globalité, c'est-à-dire la région dans son contexte.

Par exemple, la mer et le ciel peuvent être tous deux représentés par des régions bleues uniformes, mais dans la plupart des cas, nous sommes capables de faire la différence, notamment grâce à la position des régions concernées par rapport aux autres régions de l'image. Un papier peint bleu dans une chambre pourrait également être confondu avec un ciel, mais le fait d'avoir déterminé si la scène est une scène d'intérieur ou une scène d'extérieur peut permettre de faire la différence.

Nous sommes d'autant plus confrontés à cette ambiguïté des régions que le nombre d'objets que l'on cherche à reconnaître est grand, et donc, même si le problème ne se pose que rarement quand il s'agit de reconnaître une dizaine ou une centaine d'objets, il convient de s'y intéresser pour le jour où l'on envisagera de dépasser le millier d'objets.

Nous étudions dans ce chapitre deux types de désambiguïsation : l'une utilisant les relations spatiales entre les régions pour aider la reconnaissance de ces régions (ici : les fonds) et obtenir une description spatialement cohérente de l'image ; l'autre améliorant la reconnaissance des objets à l'aide de la classe des scènes dans lesquelles sont ces objets.

7.1 Désambiguïsation par relations spatiales

Afin de comprendre ce qui a motivé notre travail sur l'utilisation des relations spatiales dans le cadre de la désambiguïsation, étudions les deux images de la figure 7.1.

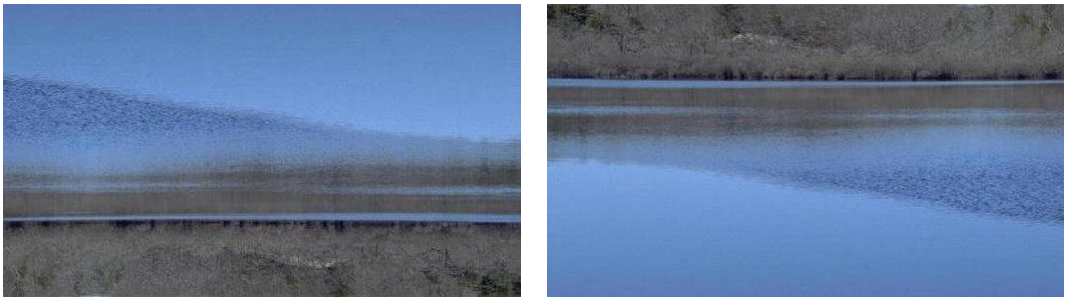


FIG. 7.1 – Ambiguïté entre le ciel et l'eau en termes de couleur et de texture. L'image de gauche correspond à l'image de droite renversée.

Dans cette image, nous reconnaissons rapidement la présence du ciel en haut de l'image de gauche, et celle de l'eau en bas de l'image de droite. Il s'agit en fait de la même image renversée, l'image de droite étant l'originale. Par conséquent, les propriétés de couleur et de texture de ces deux régions que nous reconnaissons tantôt comme du ciel et tantôt comme de l'eau sont les mêmes, et ce n'est pas cela qui nous a permis de faire la différence. Nous pensons que les relations spatiales constituent l'élément décisif dans ce cas, et proposons dans cette partie de mettre en œuvre un tel raisonnement spatial dans un système de reconnaissance d'objets.

Dans le domaine de la reconnaissance de partitions musicales, Rossant et al. [118] ont déjà montré que l'intégration des relations spatiales relatives entre les symboles et l'utilisation des règles musicales permettait d'améliorer grandement les systèmes de reconnaissance, donnant de meilleurs résultats que les logiciels commerciaux. L'idée est que pour chaque symbole, trois propositions sont faites, chacune avec un degré de possibilité indépendant des autres symboles. Puis, les relations entre symboles, à la fois relations spatiales et le fait que les règles musicales doivent être respectées, sont utilisées afin de choisir le meilleur ensemble de symboles pour la partition.

Notre approche est similaire à celle-ci en ce sens que nous souhaitons tout d'abord

générer un ensemble d'objets probables pour une région donnée, puis utiliser les règles de relations spatiales pour décider de l'ensemble spatialement cohérent d'objets à conserver.

7.1.1 Apprentissage de régions ambiguës

Avant l'étape de reconnaissance d'objets, une segmentation de l'image est effectuée avec l'algorithme de cascade en morphologie mathématique développé par Marcotegui et Beucher [89] que nous avons implémenté, qui est à la fois rapide et efficace pour les images en couleur. Nous l'avons paramétré afin d'obtenir une segmentation en au plus vingt régions. Pour chaque région, les descripteurs RVB-64 et LEP (décrits au chapitre 3) sont calculés et utilisés pour entraîner un SVM binaire probabiliste [21] par classe. Pour chaque classe, l'objet à apprendre est utilisé comme exemple positif, et tous les autres objets sont considérés comme exemples négatifs.

Nous nous intéressons ici plus particulièrement à l'apprentissage de classes ambiguës qui sont indifférenciables avec les descripteurs LEP et RVB-64. Considérons le cas où nous n'avons que deux objets A et B à apprendre qui sont parfois ambigus, tels que le ciel et l'eau. Si nous donnons le même poids aux exemples positifs et négatifs lors de l'apprentissage, alors l'apprentissage de SVM-A (A en positif, B en négatif), donnera la même frontière entre A et B que l'apprentissage de SVM-B (B en positif, A en négatif). Si le premier SVM reconnaît une région comme étant A à 80% et B à 20% (la somme faisant 100%), alors le deuxième SVM aura les mêmes probabilités. Nous voudrions que dans le cas où la région à reconnaître est ambiguë, la probabilité de A donnée par SVM-A soit forte, et la probabilité de B donnée par SVM-B également, la somme étant alors supérieure à 100%.

La solution que nous proposons consiste à donner un poids plus important à la classe positive. De cette manière, un objet pourra être classé comme A par le SVM-A et B par le SVM-B. La librairie libSVM [21] que nous utilisons permet simplement d'utiliser des SVMs binaires probabilistes, et de donner des poids différents pour chaque classe. L'effet des poids sur l'apprentissage est schématisé sur la figure 7.2.

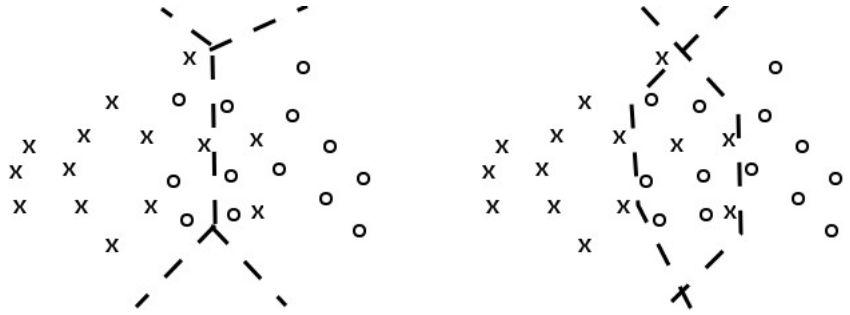


FIG. 7.2 – Exemple de l'effet des poids sur la classification. L'image de gauche : même poids pour les exemples positifs et négatifs. Image de droite : un poids de 3 est appliqué aux exemples positifs et un poids de 1 aux exemples négatifs, permettant de mieux gérer l'ambiguïté intrinsèque des données dans la zone de chevauchement.

Nous considérons deux classes 'x' et 'o' se superposant (par exemple, 'x' peut être la classe « ciel » et 'o' la classe « eau ».) Si un poids égal est donné aux deux classes pour l'apprentissage, alors la frontière apprise entre les deux classes ne dépend pas de quelle classe est positive, et laquelle est négative, puisque le problème étudié est alors symétrique. En revanche, si nous donnons un poids plus grand aux exemples positifs lors de l'apprentissage, alors un objet de la zone commune aux deux classes sera appris comme 'x' quand 'x' est la classe positive, et 'o' quand 'o' est la classe positive. Nous avons utilisé en pratique un poids de 3, ce qui signifie qu'une erreur de classification sur un exemple positif coûte autant qu'une erreur de classification sur trois exemples négatifs. Nous avons déjà vu ce qui se passe pour un poids plus petit (de 1). Un poids plus grand (par exemple 10), à l'inverse, a tendance à trop avantager la classe positive. Dans ce cas, le classificateur obtenu tend à classer tout nouvel élément dans la classe positive avec une forte probabilité, ce qui n'est pas discriminant au final.

Pour chaque région, nous conservons finalement les hypothèses pour lesquelles la probabilité retournée est supérieure à 30%, et rajoutons l'hypothèse que l'objet est inconnu avec une probabilité de 30%.

7.1.2 Calcul des relations spatiales

Nous calculons quatre relations spatiales : « au-dessus » , « en-dessous », « à gauche » et « à droite », en utilisant la méthode de l'histogramme des angles [100] décrite dans la section 2.2.1.

Afin d'accélérer le calcul de ces relations, seuls 500 pixels au plus sont utilisés pour chaque région dans le calcul de cet histogramme. En choisissant les 500 pixels, il faut faire attention à ce qu'ils soient représentatifs de la forme de la région. Nous achevons cela en triant les pixels dans une liste, dans l'ordre de la lecture – le premier pixel est celui d'en haut à gauche, et le dernier est celui d'en bas à droite – puis en sélectionnant un échantillon de pixels de cette liste de manière régulière du type 1 sélectionné tous les N pixels. Nous avons vérifié que le calcul des relations spatiales sur ces 500 pixels, ce qui représente tout de même 250 000 couples, est une très bonne approximation de ce que l'on obtient en utilisant tous les pixels. Cette méthode a l'avantage d'être rapide à implémenter et à calculer.

7.1.3 Fonction de cohérence

Le but de la fonction de cohérence est d'évaluer quelles hypothèses sont les meilleures, étant donné la connaissance sur l'agencement spatial des objets, les probabilités de reconnaissance des différents objets retournées par les SVMs pour chaque région et les relations spatiales entre les régions de l'image.

Soient N régions R_i dans l'image, la fonction de cohérence est évaluée pour chaque combinaison possible. Soit O_i un objet attribué à la région R_i. Une hypothèse peut être par exemple : $O_1 = ciel, O_2 = inconnu, O_3 = eau$, signifiant que « la région 1 est du ciel, la région 2 n'est pas un objet connu et la région 3 est de l'eau ». Nous proposons la formule suivante :

$$C(Image) = \sum_{\substack{i,j=1 \\ i \neq j}}^{N} \sum_{\Re} C_{\Re}(R_i(O_i), R_j(O_j))$$

où

$$C_{\Re}(R_i(O_i), R_j(O_j)) = P(O_i) * P(O_j) * (R_i \Re R_j) * Eval(O_i, \Re, O_j)$$

avec les notations suivantes :

- $P(O_i)$ est la probabilité retournée par le SVM de détection de l'objet O_i dans la région R_i ;
- $\Re$ est une relation spatiale entre deux objets, prenant ses valeurs parmi : « à gauche », « à droite », « au-dessus », « en-dessous » ;
- $(R_i \Re R_j)$ est le degré avec lequel la relation spatiale relative $\Re$ entre les régions R_i et R_j est vérifiée. Par exemple, si la région R_i est à 80% au-dessus de la région R_j, 10% en-dessous et 10% à droite, alors « R_i au-dessus de $R_j = 0.8$ » ;
- $Eval(O_i, \Re, O_j)$ est une fonction utilisant la connaissance. Elle retourne une valeur indiquant si la relation $(O_i, \Re, O_j)$ entre deux objets est en accord avec les règles (+1), en contradiction avec les règles (-1), ou n'est pas mentionnée dans les règles (0).

Ces notations seront illustrées sur un exemple dans la section suivante. Cette fonction ne peut être appliquée que si l'on a deux ou plusieurs régions reconnues comme des objets : avec une seule région, on ne peut pas calculer de relation spatiale.

Le maximum de cette fonction est trouvé simplement en essayant toutes les combinaisons. Dans les tests que nous avons faits, nous n'avons typiquement pas plus de 4 régions par image et 3 hypothèses pour chaque région, ce qui fait 81 combinaisons, et est très rapide à calculer. Dans le cas où il y aurait trop de combinaisons à étudier, des algorithmes tels que le recuit simulé, ou toute autre méthode d'optimisation pourraient être envisagés.

Afin d'analyser cette fonction, imaginons une image avec trois objets. Plusieurs cas sont possibles :

1. les trois couples d'objets, (O_1, O_2), (O_1, O_3) et (O_2, O_3) vérifient $Eval(O_i, \Re, O_j) > 0$. Alors la contribution de chaque couple est positive, la fonction de cohérence est aussi toujours positive, et le meilleur score est obtenu en gardant tous les objets les plus probables pour chaque région ;
2. deux objets O_1 et O_2 sont incompatibles : $Eval(O_1, \Re, O_2) < 0$, mais sont compatibles avec O_3. Dans ce cas, le score dépend principalement de la cohérence de chaque couple de fonds. Si $C(R_1(O_1), R_2(O_2)) + C(R_1(O_1), R_3(O_3)) > 0$ et $C(R_1(O_1), R_2(O_2)) + C(R_2(O_2), R_3(O_3)) > 0$ alors les trois objets sont conservés, sinon, le score sera plus grand si l'objet O_1 ou O_2 dont la cohérence avec O_3 est la plus petite est changé en « inconnu ». Quand les trois objets sont conservés, la description finale de l'image reste incohérente, et il faudrait plus de connaissance, comme par exemple un quatrième objet, pour décider ;

3. un objet O_1 est incohérent avec les deux autres objets O_2 et O_3, mais O_2 et O_3 sont cohérents. Alors la contribution de O_1 est négative dans les deux couples, et le changer en « inconnu » améliorera le score. La combinaison (O_2, O_3) est meilleure car elle donne un score positif tandis que O_1 seul donne un score de zéro ;
4. les trois couples d'objets sont incohérents. Alors, chaque couple a une contribution négative, et la fonction de cohérence est toujours négative. Le meilleur score est 0, et nous gardons l'objet qui a la probabilité de détection la plus haute, les deux autres objets étant alors changés en « inconnu ».

D'une manière générale, si tous les couples de régions sont incohérents, alors le meilleur score obtenu sera 0, pour toutes les combinaisons où l'on ne garde qu'une seule ou aucune région, si bien qu'il existe plusieurs maxima globaux. Dans ce cas, nous gardons toujours la région dont la probabilité de reconnaissance individuelle est la plus haute. Si nous pouvons trouver au moins deux régions qui ne sont pas en contradiction, alors nous sommes assurés que le maximum global sera supérieur à 0 et que la meilleure combinaison contiendra au moins deux régions.

En comparant les résultats de détection avant et après avoir appliqué le raisonnement spatial, nous avons trois types de modifications pour une région donnée : l'objet reconnu pour la région est conservé, l'objet reconnu est changé en un autre objet, ou la région est considérée comme n'étant pas un objet connu.

7.1.4 Exemple

Nous appliquons dans cette partie la fonction de cohérence définie ci-dessus à un exemple où nous cherchons à reconnaître les fonds dans les images de type photographique. Huit fonds sont considérés : ciel, eau, neige, arbre (feuillage), herbe, sable, terre et bâtiment. Ces fonds peuvent être séparés en trois groupes selon leur position relativement à la ligne d'horizon. Le premier groupe (groupeA) contient les fonds qui sont toujours au-dessus de la ligne d'horizon, le second groupe (groupeB) ce qui peuvent être au-dessus ou en-dessous, et le troisième groupe (groupeC) ceux qui sont toujours en-dessous. Les groupes suivants sont formés :

$$\begin{cases} \text{groupeA} = \{ \text{ ciel } \} \\ \text{groupeB} = \{ \text{ arbre, bâtiment } \} \\ \text{groupeC} = \{ \text{ eau, herbe, neige, sable, terre } \} \end{cases}$$

Nous n'utilisons pas de détecteur de ligne d'horizon, mais ces groupes nous permettent d'établir des règles simples. La fonction *Eval* explicitant ces règles est donnée dans le tableau 7.1.

En considérant ce tableau, nous nous rendons compte que la relation « en haut à droite » engendre des erreurs dans la détection finale. Prenons comme exemple une région reconnue comme pouvant être soit « arbre », soit « herbe » et située en haut à droite d'une région « eau ». Le calcul des relations spatiales floues de ces deux régions donne 50% pour « à droite » et 50% pour « au-dessus ». Pour le couple *(trees, water)*, seule la relation

$(A, \Re, B)$	$Eval(A, \Re, B)$
$(groupeA, au-dessus, groupeB)$	$+1$
$(groupeA, en-dessous, groupeB)$	-1
$(groupeA, au-dessus, groupeC)$	$+1$
$(groupeA, en-dessous, groupeC)$	-1
$(groupeB, au-dessus, groupeA)$	-1
$(groupeB, en-dessous, groupeA)$	$+1$
$(groupeB, au-dessus, groupeC)$	$+1$
$(groupeB, en-dessous, groupeC)$	-1
$(groupeC, au-dessus, groupeA)$	-1
$(groupeC, en-dessous, groupeA)$	$+1$
$(groupeC, au-dessus, groupeB)$	-1
$(groupeC, en-dessous, groupeB)$	$+1$
$(groupeA, -, groupeA)$	$+1$
$(groupeB, -, groupeB)$	$+1$
$(groupeC, -, groupeC)$	$+1$
autres cas	0

TAB. 7.1 – Description de la fonction $Eval$. « - » est mis pour remplacer n'importe laquelle des quatre relations : au-dessus, en-dessous, à droite, à gauche.

« au-dessus » est prise en compte dans la règle $(groupeB, au-dessus, groupeC) = 1$. La cohérence vaut alors :

$$C = P(arbre) * P(eau) * (0.5) * 1$$

Tandis que pour le couple (*herbe, eau*) les deux relations sont prises en compte dans la règle $(groupeC, -, groupeC) = 1$, ce qui donne :

$$C = P(herbe) * P(eau) * (0.5 + 0.5) * 1$$

Ainsi, l'hypothèse *herbe* est clairement avantagée par rapport à *arbre*. Pour pallier ce déséquilibre, les relations sont modifiées lorsque sont considérés deux éléments qui ne sont pas du même groupe : les relations « au-dessus » et « en-dessous » sont multipliées par un même facteur de telle sorte que leur somme vaille 100%. Les deux fonctions de cohérence ci-dessus deviennent alors comparables.

Prenons un exemple où un fond mal reconnu devient correct après l'utilisation de ces règles spatiales. Dans l'image de la figure 7.3, le ciel (région 1) est détecté comme pouvant être : « neige » (44%), « ciel » (43%) et « inconnu » (30%). La région 2 est soit « bâtiments » (36%), soit « inconnu » (30%), la région 3 n'est pas reconnue à cause d'une segmentation imprécise, et la région 4 est « terre » (42%) ou « inconnu » (30%).

Les scores retournés par la fonction de cohérence pour chaque hypothèse sont donnés dans le tableau 7.2.

Une détection de régions prises individuellement donne *1=neige, 2=bâtiments, 4=terre* comme le meilleur ensemble, alors que la fonction de cohérence atteint son maximum

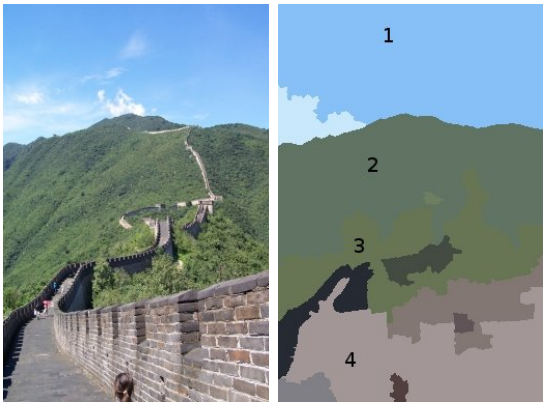
FIG. 7.3 – Image d'exemple et sa segmentation. Les quatre régions numérotées sont celles qui sont suffisamment grandes pour être considérées comme étant potentiellement des fonds.

région 1	région 2	région 4	score
inconnu	inconnu	inconnu	0
ciel	inconnu	inconnu	0
neige	inconnu	inconnu	0
inconnu	bâtiments	inconnu	0
ciel	bâtiments	inconnu	0.31
neige	bâtiments	inconnu	-0.32
inconnu	inconnu	terre	0
ciel	inconnu	terre	0.36
neige	inconnu	terre	0.37
inconnu	bâtiments	terre	0.28
ciel	bâtiments	terre	0.96
neige	bâtiments	terre	0.34

TAB. 7.2 – Scores pour la fonction de cohérence appliquée à l'image de la figure 7.3.

pour *1=ciel, 2=bâtiments, 4=terre*. La deuxième meilleure hypothèse est *1=neige, 2=inconnu, 4=terre* qui est également cohérente et a un meilleur score que *1=neige, 2=bâtiments, 4=terre* qui ne l'est pas.

Plus d'exemples sont donnés sur les figures 7.4 à 7.7.

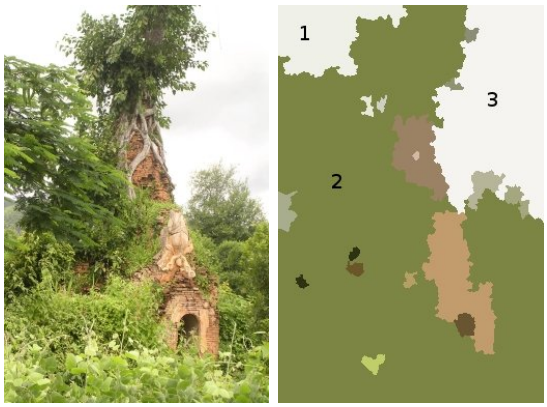


Région	Sans RS	Avec RS
1	ciel	ciel
2	herbe	arbre
3	arbre	arbre
4	bâtiment	bâtiment

FIG. 7.4 – Comparaison de la détection de fonds avec raisonnement spatial (RS) et sans raisonnement spatial. Dans cet exemple, la reconnaissance incorrecte de l'herbe a été changée en arbre, car elle générait un conflit avec la détection de bâtiments et d'arbres en-dessous de cette région.

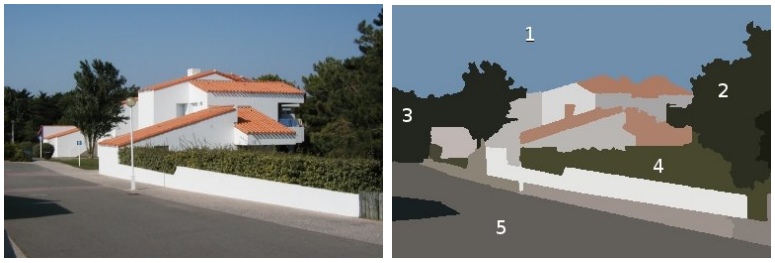
Notre algorithme a été évalué sur une base de données de 10000 images annotées manuellement, dont 4076 proviennent de la base Corel [149] et 5924 du noyau de la base CLIC [102]. La base d'apprentissage pour les fonds est constituée d'environ 300 régions segmentées automatiquement, mais sélectionnées manuellement, extraites de la base Corel uniquement, et donc indépendantes de la base CLIC en termes de qualité d'images.

Le processus d'évaluation est le suivant : chaque image est d'abord segmentée, puis chaque région dont la taille est supérieure à 5% de la surface de l'image est classée par des séparateurs à vaste marge correspondants à chacun des huit fonds, qui retournent une liste des fonds candidats avec leurs probabilités. La combinaison qui garde chacun des fonds dont la probabilité est la plus grande sans utiliser le raisonnement spatial est ce que l'on appelle le résultat *avant RS*. Celle qui maximise la fonction de cohérence décrite ci-dessus est le résultat *après RS*. Les annotations en double pour une image sont ensuite retirées. Par exemple, si une région de ciel est segmentée en deux régions, et que chacune est correctement reconnue, on ne gardera qu'une fois le mot-clé « ciel ».



Région	Sans RS	Avec RS
1	neige	ciel
2	arbre	arbre
3	ciel	ciel

FIG. 7.5 – La détection d'arbres et du ciel invalide l'hypothèse de la présence de neige et est en faveur de son remplacement par du ciel.



Région	Sans RS	Avec RS
1	eau	ciel
2	arbre	arbre
3	herbe	arbre
4	arbre	arbre
5	ciel	inconnu

FIG. 7.6 – Sur la région correspondant à la route, la détection de ciel est changée en « inconnu ». La présence d'arbres dans les régions 2 et 4 permet aussi de résoudre l'ambiguïté eau/ciel de la région 1, et l'ambiguïté herbe/arbre de la région 3.

Région	Sans RS	Avec RS
1	ciel	ciel
2	arbre	arbre
3	arbre	arbre
4	arbre	arbre
5	ciel	neige

FIG. 7.7 – Exemple en prenant une image issue d'Internet. L'hypothèse du ciel a été écartée pour la région 5 et a été remplacée par neige (au lieu de eau), car le reflet des nuages donne une texture et une couleur plus proches de la neige que de l'eau. C'est un exemple des limitations de notre algorithme qui est incapable de résoudre la confusion neige/eau.

Un mot-clé présent à la fois dans le résultat de l'annotation automatique et dans l'annotation manuelle est ce que l'on appelle une classification correcte. Un mot-clé trouvé par l'annotation automatique mais qui n'est pas dans l'annotation manuelle est un faux positif.

Par exemple, supposons que l'on ait une image dont l'annotation manuelle contient « ciel, arbre, eau », et qu'elle soit reconnue de la manière suivante : le ciel est segmenté en deux régions, chacune reconnue comme du ciel, les arbres sont reconnus correctement et l'eau est reconnue comme de la neige. L'annotation automatique donne alors « ciel, ciel, arbre, neige », ou, en éliminant les doublons, « ciel, arbre, neige ». Dans cet exemple, « ciel » et « arbre » sont deux classifications correctes, « eau » est un fond non détecté, que nous ne prenons pas en compte, et « neige » est un faux positif.

Sur cette base de données de 10000 images, 4124 ont été annotées automatiquement comme contenant au moins deux fonds après reconnaissance des fonds, sans appliquer de raisonnement spatial. Étant donné que la désambiguïsation spatiale n'a pas d'effet sur les images avec un ou aucun fond, nous l'avons appliqué uniquement sur ces 4124 images. Comme cet algorithme ne peut que changer un fond reconnu en un autre fond, ou en « région inconnue », il tend à réduire le nombre de fonds détectés dans une image. Néanmoins, nous ne perdons pas trop de fonds, et l'algorithme a même préservé quelques images avec au moins 5 fonds, comme nous pouvons le voir sur le tableau 7.3.

Sur le tableau 7.4 sont affichés le « rapport du taux de classifications correctes » et le « rapport du taux de faux positifs » obtenus avec l'ajout du raisonnement spatial. Le « rapport du taux de classifications correctes » (resp. « rapport du taux de faux positifs »)

	Nombre d'images	
Nb	avant RS	après RS
0	0	0
1	0	116
2	1419	1518
3	1390	1336
4	803	723
5	371	314
6	104	88
7	27	20
8	9	8
9	1	1

TAB. 7.3 – Nombre d'images où Nb fonds sont détectés avant et après l'analyse de relations spatiales.

est défini comme le taux de classifications correctes (resp. taux de faux positifs) avec raisonnement spatial divisé par le même taux obtenu sans appliquer le raisonnement spatial. Le rapport idéal pour les classifications correctes est de 100% ou plus. Un rapport de 100% est obtenu si aucun fond correct n'est éliminé. Un rapport supérieur à 100% est observé si un fond incorrectement détecté est changé en un fond correct. C'est le cas notamment pour « terre » . Nous remarquons également que le rapport de faux positifs est toujours inférieur au rapport de classifications correctes, signifiant que les fonds éliminés grâce au raisonnement spatial sont principalement des fausses détections.

Fond	Rapport de classifications correctes	Rapport de faux positifs
ciel	98,9%	81,1%
eau	97,5%	87,1%
arbre	98,9%	91,0%
bâtiment	98,3%	94,2%
herbe	92,1%	83,4%
neige	80,0%	45,9%
terre	105,0%	88,2%
sable	88,9%	85,7%
moyenne sur les classes	94,9%	81,7%
moyenne sur les images	98,1%	86,8%

TAB. 7.4 – Rapport de classifications correctes et rapport de faux positifs en utilisant le raisonnement spatial.

Une des erreurs les plus fréquentes est de confondre la neige avec le ciel. L'algorithme que nous avons proposé ici corrige bien ce cas, comme nous pouvons le constater sur le tableau 7.4 : 54,1% des faux positifs initialement détectés comme « neige » sont mo-

difiés (changés en autre chose, ou supprimés) grâce au raisonnement spatial. D'autres confusions fréquentes sont ciel/eau et herbe/arbre.

Parmi les 4611 images contenant au moins deux fonds, nous avons reconnu 13410 fonds parmi lesquels 760 (5,7%) ont été modifiés par l'analyse de cohérence spatiale. Ces modifications sont de 5 types :

1. une mauvaise détection est modifiée en bonne détection (+)
2. une bonne détection est modifiée en une autre bonne détection (=). Cela peut arriver par exemple dans les images contenant à la fois des arbres et de l'herbe. Si la région segmentée correspondante contient à la fois les deux éléments, alors changer la reconnaissance « arbre » en « herbe » ne change pas le nombre de fonds correctement identifiés dans l'image.
3. une bonne détection est changée en mauvaise détection (-).
4. une bonne détection est remplacée par « inconnu » (-).
5. une mauvaise détection est remplacée par « inconnu » (+)

Le nombre de ces modifications est reporté dans le tableau 7.5.

Type de modification	Nombre d'images
1	14 (1,8%)
2	2 (0,3%)
3	41 (5,4%)
4	70 (9,2%)
5	633 (83,3%)

TAB. 7.5 – Nombre d'images concernées par chaque modification.

La principale modification est celle d'éliminer une mauvaise reconnaissance (83,3%), ce qui était notre objectif principal. En ce qui concerne les modifications d'un fond en un autre, 5,4% détériorent la classification, alors que seulement 1,8% l'améliorent. Le résultat de classification serait donc meilleur si nous n'autorisions pas le changement d'un fond en un autre (types de modifications 1, 2 et 3), en conservant seulement la possibilité de changer une région reconnue comme un fond en « inconnu ».

Le principal facteur causant le retrait d'un fond correctement détecté est la présence d'un fond incorrectement détecté autre part dans l'image, et qui a une probabilité de détection plus forte. Cela arrive souvent pour des régions qui ne sont pas des fonds. Par exemple, les éléphants sont souvent classés comme étant des bâtiments, et les rues comme étant de l'eau ou du ciel, ce qui perturbe l'analyse spatiale.

Cela ne peut pas être résolu si nous considérons que l'ensemble des huit fonds est un monde clos. Une solution consisterait à créer une neuvième classe contenant des exemples de non-fonds, où nous pourrions mettre par exemple les animaux qui posent problème. Nous avons récemment essayé d'ajouter quelques images à cette neuvième classe, et cela réduit de beaucoup le taux de faux positifs. Cependant, le danger à éviter est de mettre trop d'exemples négatifs, ce qui pourrait réduire également le taux de bonnes détections.

7.2 Contexte

Nous avons mené à titre exploratoire, avec deux autres personnes du laboratoire, quelques expériences sur la désambiguïsation par le contexte. Il s'agit plus particulièrement, dans nos expériences, de la tentative d'améliorer la reconnaissance des animaux en classant également la scène, et en fusionnant ces deux informations. Nous avons considéré 6 types de scènes et 5 animaux dans chaque scène :

- **champ** : vache (*cow*), lièvre (*hare*), cheval (*horse*), kangourou (*kangaroo*), mouton (*sheep*)
- **désert** : cerastes (*cerastes*, une espèce de serpent), dromadaire (*dromedary*), chacal (*jackal*), oryx (*oryx*), scorpion (*scorpion*)
- **eau** : poisson clown (*clownfish*), dauphin (*dolphin*), méduse (*jellyfish*), oursin de mer (*sea urchin*), baleine (*whale*)
- **forêt** : daim (*deer*), gorille (*gorilla*), sanglier (*boar*), écureuil (*squirrel*), pic-vert (*woodpecker*)
- **polaire** : husky (*husky*), pingouin (*penguin*), ours polaire (*polar bear*), phoque (*seal*), morse (*walrus*)
- **savane** : éléphant africain (*African elephant*), girafe (*giraffe*), lion (*lion*), rhinocéros (*rhino*), zèbre (*zebra*)

Le nom en italique est la traduction en anglais qui a été utilisée pour rechercher les images sur Internet. Pour chaque animal, 50 images pertinentes ont été collectées depuis Internet, ce qui représente 1500 images qui ont ensuite été segmentées manuellement en deux régions (animal / scène) grâce à l'outil SAIST [59] que nous avons déjà présenté, section 5.4.3 (nous faisons ici une segmentation manuelle plutôt qu'automatique afin de pouvoir évaluer les résultats de la désambigüisation indépendamment de ceux de la segmentation automatique). Nous avons donc 30 classes d'animaux avec 50 régions segmentées par classe, et 6 classes de scènes avec 250 régions segmentées par classe. Un exemple de chacune des 30 classes d'animaux est visible sur la figure 7.8.

Pour chaque animal, 20 images sont sélectionnées aléatoirement pour constituer la base d'apprentissage, et 30 images pour la base de test, représentant un taux de 40% pour l'apprentissage, 60% pour le test. Les régions des scènes correspondantes sont séparées de la même manière entre apprentissage et test. Cette sélection aléatoire des images est faite 10 fois, et les résultats montrés seront toujours une moyenne sur ces 10 lots de données. L'apprentissage et la reconnaissance sont faits uniquement sur les segmentations manuelles, afin d'évaluer la performance en reconnaissance indépendamment de la segmentation.

Les résultats obtenus en apprenant à partir des descripteurs LEP et RVB-64 sont donnés dans le tableau 7.6.

	Bonnes reconnaissance
Animaux	44,30%
Scènes	50,74%

TAB. 7.6 – Pourcentages de bonnes reconnaissances sans désambiguïsation.

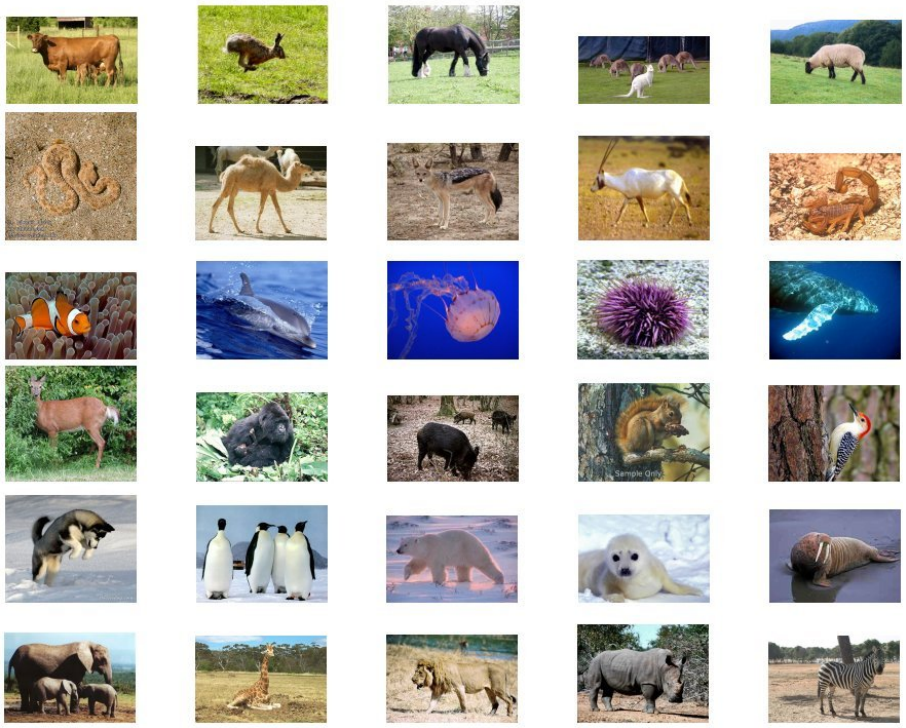


FIG. 7.8 – Exemple d'images représentant les 30 classes d'animaux. Chaque ligne correspond à l'un des 6 types de scène, dans l'ordre énoncé précédemment.

Le taux obtenu pour les animaux n'est pas étonnant par rapport à ce que nous obtenons habituellement pour un tel nombre de classes (30), le nombre d'images d'exemple par classe étant faible (20). En revanche, le taux paraît faible pour les scènes, qui ne sont séparés qu'en 6 classes. La difficulté de la séparation des scènes vient de ce que les noms que nous avons choisis pour les classes de scènes ne correspondent pas tout à fait à leurs contenus. Par exemple, les images que nous avons placées dans la catégorie « champ » seraient plus exactement définies comme étant des « images de scènes d'animaux vivants généralement dans les champs ». C'est-à-dire que si une image représente une vache dans la forêt, alors la région de forêt correspondante sera assignée à la classe champ (cf. figure 7.9). Il y a donc un certain recouvrement entre ces classes, ce qui rend la classification plus difficile. Comme nous l'avons précisé, ces travaux sont exploratoires, et d'autres expériences similaires vont être menées prochainement qui prendront en compte ce recouvrement.

Les premières expériences menées consistent à améliorer la reconnaissance des 30 animaux et des 6 scènes en utilisant :

- les probabilités $P(animal|image)$ de reconnaissance des 30 animaux pour une région donnée, obtenues par l'apprentissage d'un SVM sur les images d'animaux
- les probabilités $P(scène|image)$ des 6 scènes pour la région de scène correspon-

FIG. 7.9 – Exemples d'images de scènes, dont les animaux appartiennent à la catégorie « champ », mais dont les scènes appartiennent à une autre classe : une vache dans la forêt et un lièvre dans la neige.

dante, également obtenues par l'apprentissage d'un SVM sur les images de scènes
– les probabilités conditionnelles $P(animal|scène)$ qui peuvent être calculées de différentes façons que nous décrivons dans la suite.

Nous conservons comme reconnaissance finale pour l'image le couple (animal, scène) qui maximise $P(animal|image) * P(scène|image) * P(animal|scène)$. Cela permet notamment de prendre en compte les probabilités de détection de chaque scène pour corriger la reconnaissance de l'animal, et réciproquement, utilise les probabilités de détection de chaque animal pour améliorer la reconnaissance de la scène.

En ce qui concerne $P(animal|scène)$, nous avons envisagé d'extraire cette valeur uniquement à partir des images, ou uniquement à partir d'un corpus textuel. L'extraction à partir des images calcule $P(animal|scène)$ comme étant égal au « nombre d'images contenant à la fois l'animal et la scène » divisé par le « nombre d'images contenant la scène ». En pratique, étant donné qu'il y a 5 classes d'animaux par type de scène et autant d'images dans chaque classe d'animaux, on a :

$$P(animal|scène) = \begin{cases} \frac{1}{5} & \text{si l'animal appartient à ce type de scène} \\ 0 & \text{sinon} \end{cases}$$

Nous avons également tenté d'extraire les relations entre les scènes et les animaux à partir de corpus textuels et d'appliquer cela à cette désambiguïsation [112]. Deux ressources textuelles sur Internet ont été envisagées :
– le web dans sa globalité, en recherchant avec le moteur Exalead,
– Flickr, un site web de partage de photographies.

Une liste des mots des animaux et des scènes qui nous intéressent est établie (ce sont les mêmes mots que ci-dessus) puis nous cherchons les co-occurrences de ces mots que l'on normalise par le nombre d'occurrences des scènes pour calculer la probabilité conditionnelle suivante :

$$P(animal|scène) = \frac{occurrence(animal \cap scène)}{occurrence(scène)}$$

Quatre méthodes au total ont été testées pour calculer les co-occurrences de ces deux termes :

1. *Exalead* : nombre de pages web contenant à la fois le mot de l'animal et le mot de la scène,
2. *Exalead NEAR* : nombre de pages web contenant à la fois le mot de l'animal et le mot de la scène, ces deux mots étant contenus dans une fenêtre de 16 mots. Cela est obtenu avec l'opérateur NEAR fourni par Exalead,
3. *Annotations Flickr* : nombre d'images contenant les deux mots dans leurs annotations, les annotations étant une liste de mots,
4. *Textes Flickr* : nombre d'images contenant les deux mots dans leurs descriptions, les descriptions étant du texte libre.

Les résultats sont donnés sur le tableau 7.7.

Méthode de fusion	Bonnes reconnaissances
Aucune	44,3%
Image	**48,6%**
Exalead	46,9%
Exalead NEAR	48,1%
Annotations Flickr	45,6%
Textes Flickr	47,5%

TAB. 7.7 – Désambiguïsation de la reconnaissance des animaux en utilisant des informations extraites automatiquement à partir des images ou de corpus textuels.

Le gain obtenu varie de 2,6% à 4,3% selon la méthode ayant servi pour le calcul de la probabilité conditionnelle $P(animal|scène)$. Nous remarquons tout d'abord que la fusion à partir des co-occurrences des animaux et des scènes dans les images donne des meilleurs résultats que la fusion à partir de données textuelles. Toutefois, on pourrait très bien imaginer un système où, pour l'apprentissage, nous aurions des images de scènes d'une part, sans animaux dans ces images, et des images d'animaux ne contenant pas ou peu d'information sur la scène, par exemple des images détourées. Alors la probabilité conditionnelle $P(animal|scène)$ ne pourrait pas être apprise sur les images, et le recours à une méthode textuelle se révélerait avantageuse pour améliorer la reconnaissance dans des images de test du type de celles que nous avons.

En ce qui concerne les performances des méthodes textuelles, l'utilisation du web textuel avec le moteur Exalead offre les meilleurs résultats, qui ne semblent pas vraiment dépendre du recours à l'opérateur NEAR, tenant compte de la proximité de deux termes pour les compter comme co-occurrents, même si ce dernier offre un léger gain. Les résultats sont moins bons avec les textes de Flickr, qui est pourtant un site dédié aux photographies, notamment avec les annotations. Cela pourrait signifier que les utilisateurs mettent des annotations relatives à l'animal, mais peu relatives au contexte, de même que dans les descriptions des images en langage naturel dans une moindre mesure. En revanche, des co-occurrences entre les mots décrivant la scène, et les noms des animaux pourront se trouver notamment dans des articles de types encyclopédiques, auxquels Exalead a accès.

Enfin, une fusion des informations a également été effectuée avec un SVM, sans avoir recours aux données textuelles. Ce SVM prend en entrée les probabilités $P(animal|image)$ de reconnaissance des 30 animaux pour une région donnée, ainsi que les probabilités de reconnaissance des 6 scènes pour la région de scène correspondante, qui sont tous deux issus de l'apprentissage de deux autres SVMs : un sur les animaux et un sur les fonds. Cela permet de monter jusqu'à 53,7% de bonnes reconnaissances pour les animaux, surpassant les résultats obtenus en maximisant $P(animal|image) * P(scène|image) * P(animal|scène)$. Notons qu'il est également possible d'améliorer la reconnaissance des scènes par le même procédé, ce qui n'était pas l'objet de notre étude.

7.3 Conclusion

Nous avons démontré dans ce chapitre que la performance des algorithmes de reconnaissance, dépendant des descripteurs de bas niveau extraits et des algorithmes d'apprentissage utilisés, peut-être améliorée en aval notamment en introduisant des connaissances sémantique pour résoudre certaines ambiguïtés. Nous avons envisagé deux types de connaissances sémantiques : une connaissance sur l'agencement spatial des fonds dans les images pour améliorer la reconnaissance des fonds, et une connaissance sur la co-occurrence des animaux et des types de scènes, améliorant ainsi la reconnaissance des animaux.

Il pourrait être intéressant d'étudier s'il est possible d'étendre ces désambiguïsations à d'autres types d'objets. Par exemple, peut-on dresser une liste de relations spatiales relatives entre des fonds et des animaux : « l'oiseau est dans le ciel » ? entre deux objets : « la tasse est sur la table » ? Peut-on déterminer d'une manière fiable la probabilité de la co-occurrence de deux objets ? Les quelques travaux exploratoires que nous avons menés dans ces directions pour extraire ces informations automatiquement n'ont pas donné de résultats convainquants, mais nous n'avons scruté cette voie que très brièvement.

Chapitre 8

Conclusion et perspectives

Cette nuit, en regardant le ciel, je suis arrivé à la conclusion qu'il y a beaucoup plus d'étoiles qu'on en a besoin.
Quino, Mafalda

8.1 Conclusion

Le but de cette thèse était de réaliser un système complètement automatique pour l'annotation d'images, en tentant d'y incorporer des informations sémantiques. Par complètement automatique, nous voulons dire que le système, lors de la phase d'apprentissage, prend en entrée une liste des noms des concepts à apprendre et génère en sortie des modèles permettant de détecter ces objets. Lors de la phase de test, le système utilise ces modèles et des connaissances sémantiques pour annoter l'image de manière cohérente.

Les travaux que nous avons menés pour arriver à ce but peuvent se décomposer en différentes étapes plus ou moins indépendantes :

- la création automatique d'une base d'apprentissage, à partir d'un mot correspondant à un concept que l'on souhaite apprendre,
- la classification hiérarchique des scènes, car différentes scènes auront besoin de différents traitements et ne contiendront pas les mêmes objets,
- la détection d'un objet dans une image, ainsi que la détection de certains fonds,
- la désambiguïsation permettant d'une part l'amélioration de la reconnaissance d'objets en utilisant les relations spatiales (pour les fonds) et le contexte (pour les objets), d'autre part la génération d'une annotation sémantiquement cohérente.

8.1.1 Contributions méthodologiques

Pour la classification hiérarchique de scènes, nous nous sommes efforcés de définir un ensemble de classes permettant de séparer grossièrement les images que l'on rencontre typiquement sur Internet en fonction de leur type et de leur contenu, dans le

cas de photographies. Nous avons pour cela regroupé dans une structure hiérarchique plusieurs classifications qui existaient dans la littérature en tant que classifications binaires indépendantes les unes des autres : « intérieur / extérieur », « clipart / photo », « nature / urbain ». Nous avons également introduit de nouvelles classes qui ont été ensuite utilisées dans la campagne ImagEval : peinture, carte, « noir et blanc / noir et blanc colorisé / couleur », jour / nuit. La classification des cliparts a été faite avec une méthode adaptée, à la fois très rapide et très efficace, par rapport aux méthodes de la littérature utilisant des séparateurs à vaste marge ou des plus-proches-voisins. Les autres classifications sont faites avec des SVM appris sur des combinaisons de descripteurs de textures et de couleurs.

La création automatique d'une base d'apprentissage à partir des images d'Internet a été très peu étudiée dans la littérature et ces études concernaient seulement la possibilité d'augmenter la pertinence des N premières images en reclassant les images et d'apprendre à partir de ces images reclassées. Nous avons exploré ce domaine en montrant d'abord que le choix des mots clés utilisés pour l'interrogation compte également beaucoup pour la pertinence des images obtenues, et que ce choix peut être fait à l'aide de WordNet. Nous avons également introduit la segmentation automatique de telles images, séparant l'objet du fond. Cette segmentation a pour nous deux intérêts : la séparation de l'objet et du fond est utile pour l'apprentissage, et elle nous permet de faire un reclassement des images en fonction de l'objet segmenté (principalement sa taille et sa couleur).

La détection des fonds (ciel, arbre, bâtiment, herbe, etc.) dans les images est quelque chose qui a déjà été étudié, mais l'utilisation des relations spatiales pour permettre de désambiguïser les fonds reconnus afin d'avoir une description spatialement cohérente de l'image est quelque chose de nouveau que nous avons expérimenté au cours de cette thèse. Cela montre que des connaissances sémantiques sur les relations spatiales relatives entre les différents fonds (le ciel est au-dessus des arbres, les bâtiments sont au-dessus de l'herbe, etc.) peuvent être traduites en termes de traitement d'images afin d'améliorer les résultats de la reconnaissance.

Enfin, nous avons confirmé que la détection d'un objet est améliorée si l'on considère le contexte. Pour l'apprentissage des relations contexte/objets, nous avons comparé deux approches : ce qui est fait dans la littérature, c'est-à-dire l'apprentissage à partir de bases d'images, et une nouvelle approche qui cherche à induire ces relations à partir de corpus textuels.

8.1.2 Résultats

Chacune des quatre parties énumérées en début de section ont été évaluées au cours de la thèse.

Nous avons montré notamment que la base d'apprentissage automatiquement créée à partir des images d'Internet est de bonne qualité : nous sommes capables d'une part de sélectionner les images pertinentes afin d'améliorer la pertinence par rapport aux résultats bruts d'une recherche sur Internet, et d'autre part de faire une segmentation automatique de ces images de bonne qualité. Enfin, nous avons montré que les mêmes performances en détection d'objets sont obtenues en utilisant pour l'apprentissage la

base de données construite automatiquement ou en utilisant la base manuelle équivalente construite à partir des mêmes images où la sélection et la segmentation des images sont faites à la main.

La classification hiérarchique de scènes a pu être évaluée par la participation à la tâche 5 de la campagne d'évaluation ImagEVAL, considérant 10 types de scènes. Nous sommes arrivés deuxième sur cinq participants, avec des résultats (une MAP de 0,62) proches de ceux des premiers. D'une manière générale, les taux de bonnes classification obtenus en classification automatique de scène sont très bons (en l'occurrence, généralement supérieurs à 95%), que ce soit dans les résultats obtenus au cours de cette thèse, ou dans l'état de l'art.

La détection des fonds dans une image donne également de bons résultats, mais la détection d'objets, en revanche, ne donne que de l'ordre de 50% de bonnes détections, sur les 20 classes d'objets que nous avons utilisées pour l'évaluer. En effet, ce problème est plus difficile que la classification de scènes, étant donné que le problème de la classification est ici lié à un problème de localisation (de segmentation). Dans le cas des fonds, une segmentation automatique par couleurs localise assez efficacement les régions qui sont potentiellement des fonds, étant donné que les fonds sont en général uniformes en couleur. La localisation des objets est plus difficile, car ils peuvent posséder plusieurs couleurs avec de forts gradients entre ces couleurs auquel cas une segmentation automatique séparera l'objet en plusieurs régions. Ils peuvent aussi être de la même couleur que leur environnement, et dans ce cas au contraire, la segmentation automatique inclura une grande proportion de l'environnement dans la région contenant l'objet. Nous avons donc eu recours à des fenêtres glissantes et avons développé un algorithme optimisant la forme de cette fenêtre en fonction des modèles appris, ce qui permet également d'améliorer légèrement le taux de bonnes reconnaissances.

Enfin, nos résultats montrent que la désambiguïsation est dans la plupart des cas une étape donnant de meilleurs résultats, comparés à une annotation d'images sans désambiguïsation. Notamment, la désambiguïsation par relations spatiales, qui oblige les fonds détectés dans une même image à être spatialement cohérents, permet d'éliminer de nombreuses fausses détections tout en conservant la plupart des bonnes détections, et permet également de déterminer quel fond choisir, lorsque deux fonds visuellement proches sont reconnus sur une même région. La désambiguïsation par contexte permet, comme nous l'avons vu, d'améliorer conjointement la qualité de la reconnaissance des objets et celle des scènes.

8.2 Perspectives

8.2.1 Amélioration des algorithmes d'apprentissage

La principale limite se posant à notre système automatique d'annotation d'images est la performance en reconnaissance d'objets, qui reste plus faible qu'espérée. Nous nous sommes concentrés en effet au cours de cette thèse plutôt sur deux des aspects énumérés précédemment qui sont la constitution automatique d'une base de données

d'apprentissage et les moyens d'introduire de la sémantique dans l'annotation, à l'aide de relations spatiales et en prenant en compte le contexte.

En ce qui concerne la reconnaissance d'objets proprement dite, nous avons utilisé des descripteurs de l'état de l'art, sans chercher à en inventer de nouveaux. Nous nous sommes également fixé de n'utiliser que les séparateurs à vaste marge avec un noyau gaussien et la méthode un-contre-un pour étendre ce classifieur binaire en un classifieur multiclasse. Il semble cependant que les algorithmes d'apprentissage obtenant les meilleurs résultats dans l'état de l'art sont adaptés au problème. Notamment, pour la base Caltech, les meilleurs résultats sont obtenus par Lazebnik et al. [73] qui définissent leur propre noyau (un noyau pyramidal) pour entraîner des séparateurs à vaste marge.

Nous pensons que la faible performance pour la reconnaissance d'objets est due d'une part à la difficulté de la tâche (par rapport à la reconnaissance de fonds), et d'autre part au faible nombre d'images par rapport à la taille, bien plus grande, des descripteurs. Dans ce cas, un phénomène de sous-apprentissage se produit. Il pourrait être intéressant d'explorer les approches proposées dans la littérature pour apprendre des concepts à partir d'un seul exemple.

8.2.2 Sacs de mots

J'ai fais le choix, dans cette thèse, d'utiliser les descripteurs qui existaient déjà au sein du laboratoire afin de ne pas passer trop de temps au codage d'autres descripteurs. En particulier, l'approche par sacs de mots n'était pas encore disponible au laboratoire au moment où cette thèse a été menée. Cette approche, créant un vocabulaire visuel à partir de descripteurs SIFTs autour de points d'intérêt est l'une des méthodes en vogue en ce moment, offrant dans l'état de l'art de très bonnes performances.

Nous n'avons pas eu le temps d'explorer cette piste, mais nous pensons que de meilleurs résultats pourraient être obtenus en intégrant ces descripteurs locaux, qui seraient complémentaires aux descripteurs plutôt globaux que nous utilisons.

8.2.3 Utilisation de la sémantique

Les recherches sur l'utilisation du contexte pour améliorer la reconnaissance d'objets se poursuivent au sein de notre laboratoire. Elle sera évaluée sur une base de données plus grande, et deux approches seront comparées pour connaître les relations contexte-objet : une extraction de ces relations directement dans les images, et une extraction de ces relations dans des corpus textuels. Deux corpus textuels seront explorés : le web et les annotations de Flickr.

L'utilisation des relations spatiales pour la désambiguïsation n'est faite actuellement que pour les fonds, et il pourrait être intéressant d'étendre cela aux relations spatiales entre les objets et les fonds, ou aux relations inter-objets. Il devrait notamment être possible d'extraire de telles relations à partir de corpus textuels, et de les traduire en termes de traitement d'images, comme par exemple : « l'oiseau est dans le ciel ; sur la branche », « la tasse est sur la table ».

8.2.4 Vers le millier de concepts

Rappelons enfin que le nombre de concepts que l'homme peut reconnaître est estimé à 30000. Nous avons pu voir que nous obtenons de très bons résultats pour la classification de scènes dans la campagne imagEval : il s'agit de 10 classes. Pour la classification d'objets, nécessitant également la localisation, nous avons des résultats d'environ 50% pour un ensemble de 20 classes d'objets complexes (pas uniformes en couleur). La plupart des publications, encore aujourd'hui, ne considèrent pas plus de 20 concepts simultanément.

Le domaine de la centaine de classes a été un peu exploré avec la base de données Caltech 101, contenant 101 objets et une classe de fonds. Ce problème est plus facile que notre expérience sur 20 classes, car les objets sont déjà détourés, ce qui nous libère du problème de la segmentation et de la localisation. Sur cette base, les meilleurs résultats que nous avons obtenus sont de 44% de bonnes classifications, et ceux de l'état de l'art atteignent au plus 66%, alors que le cerveau humain classifie cet ensemble aisément sans erreur. Sur la base Caltech 256, le seul résultat publié reporte 34% de bonnes détections. Pour ces deux bases, la désambiguïsation par le contexte ne peut pas s'appliquer, car il n'y a pas ou peu de contexte.

Nous pouvons, avec ce que nous avons développé, créer sans trop de difficultés une base d'apprentissage pour 1000 concepts, mais les algorithmes d'apprentissage ne sont pas assez performants pour les classer avec un faible taux d'erreur. Les approches qui tentent de modéliser le système visuel humain nous paraissent intéressantes à explorer.

Annexe A

Liste des publications

Voici une liste des publications co-écrites au cours de cette thèse.

A.1 Conférences

A.1.1 Conférences internationales

Présentations orales

- Christophe Millet, Isabelle Bloch, Adrian Popescu, *Using the Knowledge of Object Colors to Segment Images and Improve Web Image Search*, RIAO 2007, 30 mai - 1 juin, 2007, Pittsburg, USA.
- Adrian Popescu, Christophe Millet, Pierre-Alain Moëllic, Patrick Hède, Gregory Grefenstette, *Automatic Construction of a Grounded Multimedia Ontology of Objects to Illustrate Concepts in a Learning Process*, Proceedings of the 10th NETTIES Conference, 6-9 septembre, 2006, Timisoara, Roumanie.
- Christophe Millet, Gregory Grefenstette, Isabelle Bloch, Pierre-Alain Moëllic, Patrick Hède, *Automatically populating an image ontology and semantic color filtering*, International Workshop Ontoimage'2006 Language Resources for Content-Based Image Retrieval, Gènes, Italie, pp. 34-39
- Svetlana Zinger, Christophe Millet, Benoit Mathieu, Gregory Grefenstette, Patrick Hède, Pierre-Alain Moëllic, *Clustering and semantically filtering web images to create a large-scale image ontology*, Proceedings of the IS&T/SPIE 18th Symposium Electronic Imaging 2006, 15-19 janvier, 2006, San Jose Californie, USA.
- Christophe Millet, Isabelle Bloch, Patrick Hède, et Pierre-Alain Moëllic, *Using Relative Spatial Relationships to Improve Individual Region Recognition*, Proceedings of the Second European Workshop on the Integration of Knowledge, Semantics and Digital Media Technologies, EWIMT'05, 30 novembre - 1 décembre, 2005, Londres, Royaume-Uni, pp. 117-126.
- Pierre-Alain Moëllic, Patrick Hède, Gregory Grefenstette, Christophe Millet, *Evaluating Content Based Image Retrieval Techniques with the One Million Images CLIC TestBed*, Proceedings of the Second World Enformatika Congress, WEC'05,

February 25-27, 2005, Istanbul, Turkey, pp 171-174.

Posters et démos

- Adrian Popescu, Christophe Millet, Pierre-Alain Moëllic, *Ontology Driven Content Based Image Retrieval*, CIVR 2007 - posters session, 9-11 juillet, 2007, Amsterdam, Pays-Bas.
- Adrian Popescu, Pierre-Alain Moëllic, Christophe Millet, *SemRetriev – an Ontology Driven Image Retrieval System*, CIVR 2007 - demo session, 9-11 juillet, 2007, Amsterdam, Pays-Bas.
- Adrian Popescu, Christophe Millet, Gregory Grefenstette, Pierre-Alain Moëllic, Patrick Hède, *Imaging Word - Wording Images*, SAMT 2006 - poster session, 6-9 décembre, 2006, Athènes, Grèce.

Campagnes d'évaluation

- Svetlana Zinger, Christophe Millet, Benoit Mathieu, Gregory Grefenstette, Patrick Hède, Pierre-Alain Moëllic, *Extracting an ontology of portrayable objects from WordNet*, Proceedings of the MUSCLE/ImageCLEF Workshop on Image and Video Retrieval Evaluation, pp 17-23, September, 2006, Vienna, Austria.
- Romaric Besançon, Christophe Millet, *Merging results from different media : experiments at ImageCLEF 2005*, Working Notes for the ImageCLEF 2005 Workshop, 21-23 septembre, 2005, Vienne, Autriche.

A.1.2 Conférences nationales

- Christophe Millet, *Connaître la Couleur des Objets pour Segmenter les Images et Améliorer la Recherche d'Images sur le Web*, Actes de la quatrième conférence en recherche d'information et applications CORIA, 28-30 mars, 2007, Saint-Étienne, France, pp. 401-412.

Annexe B

Glossaire

Un certain nombre de termes utilisés dans ce document ont des définitions variables suivant les personnes, ou ont été traduits de l'anglais et ne sont pas forcément répandus dans la communauté scientifique. Le présent glossaire vise à clarifier tout cela.

caractéristique : d'une image, voir « descripteur ».

classer / classifier : classer signifie ranger dans une classe et correspond à l'action de classement. Classer une image donnée, c'est lui attribuer une (ou plusieurs) classe à laquelle elle appartient. Classifier signifie définir des classes, établir des critères de rangement. Ce verbe correspond à l'action de classification. Par exemple, classifier une base d'images revient à séparer la base en plusieurs classes, et établir des critères définissant ces classes pour pouvoir ensuite y classer une nouvelle image.

clustering : traduit en français pas « groupement » ou « regroupement ».

descripteur : d'une image ou « caractéristique », correspond au mot anglais *feature*.

feature : traduit en français par « descripteur » ou « caractéristique ».

groupement ou « regroupement » : traduction de l'anglais *clustering*. Les techniques de groupement consistent à constituer des groupes (*clusters*) plus ou moins homogènes dans une base de données a priori non classifiée.

indexeur : en anglais, *indexer*. C'est un algorithme permettant de calculer des descripteurs d'une image.

k-plus proches voisins : de l'anglais *k-nearest neighbour*. L'abréviation est k-PPV en français, et k-NN en anglais.

machine à vecteur support : en anglais, *support vector machine*, parfois abrégé SVM, technique de classification mise au point par Vapnik.

regroupement : cf. « groupement ».

séparateur à vaste marge : une autre traduction possible de l'anglais *support vector machine* dont l'intérêt est de pouvoir utiliser le même acronyme.

sac de mot : en anglais, *bag of words*.

Annexe C

Comparaison des descripteurs

Afin de comparer les descripteurs présentés dans le chapitre 3, nous avons lancé de nombreux tests de classification sur la base Caltech-101. Il s'agit de comparer les taux de bonnes classifications obtenus en entraînant un séparateur à vaste marge, lorsque l'on prend chaque descripteur séparément, ou lorsqu'on les combine. Cela permet d'avoir une idée de plusieurs aspects, tout d'abord de la capacité de chaque descripteur, séparément, à permettre de classifier cette base d'image. On peut également y observer quels sont les descripteurs dont la combinaison n'apporte rien, et au contraire ceux qui semblent être complémentaires. Enfin, en considérant la taille des vecteurs produits par chaque indexeur, nous pouvons voir ceux qui sont les plus compacts, et contiennent le plus d'informations intéressantes.

La méthodologie utilisée pour évaluer la classification est celle proposée par Fei-Fei et al. [41]. Elle est la suivante : pour chacune des 101 classes de la base Caltech-101, 30 images sont sélectionnées aléatoirement pour l'apprentissage et le reste est utilisé pour le test. La moyenne du taux de bonnes reconnaissances de chaque classe est alors calculée. Ce processus est effectué dix fois, pour varier les 30 images d'apprentissage sélectionnées aléatoirement, et on calcule à chaque fois cette moyenne. La moyenne sur les dix expériences de cette moyenne est le taux affiché dans le tableau C.1.

La première chose remarquable dans ce tableau est qu'il montre que la combinaison de plusieurs descripteurs est bénéfique. Le meilleur descripteur seul est LEP qui a 29,51% de bonnes classifications, et nous somme parvenus jusqu'à 44,23% en associant 6 descripteurs.

Nous observons également que les résultats en bonnes classifications ne dépendent pas directement de la taille du vecteur des descripteurs, mais que les meilleurs résultats sont obtenus quand cette taille est maximale. Cela signifie que chaque descripteur ajouté parmi les six apporte de l'information. Cependant, certains descripteurs semblent porter des informations très proches. Ainsi, l'ajout du descripteur TSVal à *BIC + LEP + Pj + RVB-64* ne permet de gagner que 0,4%, notamment car TSVal et RVB-64 portent une information semblable, relative aux couleurs dans l'image. En revanche, l'association de RVB-64 et de LEP est très avantageuse par rapport à l'utilisation d'un seul, et ce sont en effet deux descripteurs bien différents, l'un décrivant des caractéristiques de couleurs

Descripteurs	Taille	Score
BIC + Gab + TSVal + LEP + Pj + RVB-64-9	2130	44,23%
Gab + TSVal + LEP + Pj + RVB-64 + RVB-64-9 (ACP500)	1754	44,20%
Gab + TSVal + LEP + Pj + RVB-64 + RVB-64-9	1754	43,48%
BIC + Gab + TSVal + LEP + Pj + RVB-64	1618	40,91%
BIC + LEP + Pj + RVB-64 + RVB-64-9	1984	40,88%
LEP + Pj + RVB-64 + RVB-64-9	1552	40,65%
BIC + TSVal + LEP + Pj + RVB-64	1570	40,49%
BIC + LEP + Pj + RVB-64	1408	40,11%
Gab + LEP + Pj + RVB-64	1024	39,49%
TSVal + LEP + Pj + RVB-64	1138	38,46%
LEP + Pj + RVB-64	976	37,15%
BIC + Gab + LEP + RVB-64	1056	36,87%
LEP + RVB-64 + RVB-64-9	1152	36,38%
Gab + TSVal + LEP + RVB-64	778	35,50%
BIC + TSVal + LEP + RVB-64	1170	34,61%
BIC + TSVal + Pj	994	28,41%
LEP + RVB-64	576	33,34%
LEP	512	29,51%
RVB-64-9	576	25,12%
RVB-27-9	243	23,58%
Pj	400	22,68%
BIC	432	15,06%
Gab	48	13,34%
RVB-125	125	11,51%
RVB-64	64	10,34%
TSVal	162	9,40%
RVB-27	27	8,79%

TAB. C.1 – Résultats de classifications sur la base Caltech-101 avec des separateurs à vaste marge utilisant différentes combinaisons de descripteurs.

et l'autre de textures de l'image.

Enfin, nous voyons, avec les lignes 2 et 3 que l'ACP a très peu d'effets : nous gagnons 0,7% en classification ce qui n'est pas intéressant compte tenu du temps de calcul de l'ACP (quelques heures).

À titre de comparaison, les meilleurs résultats de classifications sur Caltech-101 sont d'environ 66% (cf. section 2.3.6), ce qui prouve que nous pouvons encore faire quelques progrès du côté des descripteurs et/ou des algorithmes de classification.

Bibliographie

[1] Z. Aghbari, A. Makinouchi. "Semantic approach to image database classification and retrieval", *NII Journal*, **7**, pp. 1–8, (septembre 2003).

[2] S. Aksoy, K. Koperski, C. Tusk, G. Marchisio, J. Tilton. "Learning bayesian classifiers for scene classification with a visual grammar", *IEEE Transactions on Geoscience and Remote Sensing*, **43**(3), pp. 581–589, (mars 2005).

[3] R. Anand, K. Mehrotra, C. Mohan, S. Ranka. "Efficient classification for multiclass problems using modular neural networks", *IEEE Transactions on Neural Networks*, **6**(1), pp. 117–124, (1995).

[4] L. H. Armitage, P. G. Enser. "Analysis of user need in image archives", *Journal of information science*, **23**(4), pp. 287–299, (1997).

[5] V. Athitsos, M. J. Swain, C. Frankel. "Distinguishing photographs and graphics on the world wide web". In *Proceedings of the 1997 Workshop on Content-Based Access of Image and Video Libraries (CBAIVL '97)*, pages 10–17, Washington, DC, USA, (juin 1997). IEEE Computer Society.

[6] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, M. Jordan. "Matching words and pictures", (2003).

[7] N. Ben-Haim, B. Babenko, S. Belongie. "Improving web-based image search via content based clustering". In *Conference on Computer Vision and Pattern Recognition Workshop CVPRW*, pages 106–111, Washington, DC, USA, (2006).

[8] T. Berk, L. Brownston, A. Kaufman. "A new color-naming system for graphics languages", *IEEE Computer Graphics Applications*, **2**(3), pp. 37–44, (mai 1982).

[9] I. Biederman. "Recognition-by-components : a theory of human image understanding", *Psychological Review*, **94**(2), pp. 115–147, (1987).

[10] R. Billen, E. Clementini. "Étude des caractéristiques projectives des objets spatiaux et de leurs relations", *Revue Internationale de Géomatique (Les ontologies spatiales)*, **14**(2), pp. 145–165, (2004).

[11] D. Blei, M. Jordan. "Modeling annotated data". In *Proceedings of the 26th ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 127–134, Toronto, Canada, (2003).

[12] D. Blei, A. Ng, M. Jordan. "Latent dirichlet allocation", *Journal of Machine Learning Research*, **3**, pp. 993–1022, (2003).

[13] I. Bloch, A. Ralescu. "Directional relative position between objects in image processing : a comparison between fuzzy approaches", *Pattern Recognition*, **36**, pp. 1563–1582, (2003).

[14] A. Bosch, X. Muñoz, J. Martí. "Using appearance and context for outdoor scene object classification". In *IEEE International Conference on Image Processing*, Volume 2, pages 1218–1221, Gènes, Italie, (2005).

[15] A. Bosch, A. Zisserman, X. Muñoz. "Scene classification via plsa". In *European Conference on Computer Vision*, Volume 4, pages 517–530, Graz, Autriche, (2006).

[16] A. Bosch, X. Muñoz, R. Marti. "Which is the best way to organize/classify images by content ?", *Image and Vision Computing*, **25**(6), pp. 778–791, (2007).

[17] M. R. Boutell. "Review of the state of the art in semantic scene classification". Technical Report 799, The University of Rochester, Rochester NY, USA, (décembre 2002).

[18] D. Cai, X. He, Z. Li, W.-Y. Ma, J.-R. Wen. "Hierarchical clustering of www image search results using visual, textual and link information". In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 952–959, New York, USA, (2004).

[19] P. Carbonetto, N. Freitas, K. Barnard. "A statistical model for general contextual object recognition". In *8th European Conference on Computer Vision (ECCV)*, Volume 1, Prague, République tchèque, (mai 2004).

[20] G. Carneiro, A. B. Chan, P. J. Moreno, N. Vasconcelos. "Supervised learning of semantic classes for image annotation and retrieval", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **29**(3), pp. 394–410, (mars 2007).

[21] C.-C. Chang, C.-J. Lin. *LIBSVM : a library for support vector machines*, (2001). Software available at http ://www.csie.ntu.edu.tw/ cjlin/libsvm.

[22] E. Chang, K. Goh, G. Sychay, G. Wu. "Cbsa : Content-based soft annotation for multimodal image retrieval using bayes point machines", *IEEE Transactions on Circuits and Systems for Video Technology Special Issue on Conceptual and Dynamic Aspects of Multimedia Content Description*, **13**(1), pp. 26–38, (janvier 2003).

[23] Y.-C. Cheng, S.-Y. Chen. "Image classification using color, texture and regions", *Image and Vision Computing*, **21**(9), pp. 759–776, (2003).

[24] P. Clark, R. Boswell. "Rule induction with CN2 : some recent improvements". In *Proceedings of the Fifth European Working Session on Learning*, pages 151–163, Berlin, Allemagne, (1991). Springer.

[25] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, C. Bray. "Visual categorization with bags of keypoints". In *Proceedings of ECCV International Workshop on Statistical Learning in Computer Vision*, pages 59–74, Prague, République Tchèque, (2004).

[26] C. Cusano, G. Ciocca, R. Schettini. "Image annotation using svm". In *Proceedings of the SPIE*, Volume 5304, pages 330–338, San Jose, USA, (décembre 2003).

[27] R. Datta, W. Ge, J. Li, J. Z. Wang. "Toward bridging the annotation-retrieval gap in image search by a generative modeling approach". In *MULTIMEDIA '06 : Proceedings of the 14th annual ACM international conference on Multimedia*, pages 977–986, New York, NY, USA, (2006). ACM Press.

[28] J. Daugman. "Two-dimensional spectral analysis of cortical receptive field profiles", *Vision Research*, **20**, pp. 847–856, (1980).

[29] A. Dempster, N. M. Laird, D. B. Rubin. "Maximum likelihood from incomplete data via the em algorithm", *Journal of the Royal Statistical Society*, **39**(1), pp. 1–38, (1977).

[30] T. Deselaers, D. Keysers, H. Ney. "Features for image retrieval : A quantitative comparison". In *Proceedings of the 26th DAGM Symposium on Pattern Recognition (DAGM 2004)*, Volume LNCS 3175, pages 228–236, Tübingen, Allemagne, (2004).

[31] T. G. Dietterich, G. Bakiri. "Solving multiclass learning problems via error-correcting output codes", *Journal of Artificial Intelligence Research*, **2**, pp. 263–286, (1995).

[32] G. Dorko, C. Schmid. "Object class recognition using discriminative local features". Technical Report 5497, INRIA Rhone-Alpes, (février 2005).

[33] S. T. Dumais. "Improving the retrieval of information from external sources", *Behavior Research Methods, Instruments, and Computers*, **23**(2), pp. 229–236, (1991).

[34] S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester. "Using latent semantic analysis to improve information retrieval". In *Proceedings of CHI'88 : Conference on Human Factors in Computing*, pages 281–285, Washington, D.C., USA.

[35] P. Duygulu, K. Barnard, J. de Freitas, D. Forsyth. "Object recognition as machine translation : learning a lexicon for a fixed image vocabulary". In *Proceedings of the European Conference on Computer Vision, ECCV*, pages 97–112, Copenhague, Danemark, (2002).

[36] P. Duygulu. *Translating images to words : A novel approach for object recognition.* PhD thesis, Middle East Technical University, Turquie, (2003).

[37] J. P. Eakins. "Towards intelligent image retrieval", *Pattern Recognition*, **35**(1), pp. 3–14, (janvier 2002).

[38] P. Enser, C. Sandom. "Towards a comprehensive survey of the semantic gap in visual image retrieval". In *Proceedings of the Second International Conference on Image and Video Retrieval; CIVR 2003*, pages 291–299, Urbana-Champaign, IL, USA, (2003).

[39] J. Fan, Y. Gao, H. Luo, G. Xu. "Statistical modeling and conceptualization of natural images", *Pattern Recognition*, **38**(6), pp. 865–885, (2005).

[40] J. Fauqueur. *Contributions pour la recherche d'images par composantes visuelles.* PhD thesis, l'Université de Versailles Saint-Quentin en Yvelines, (novembre 2003).

[41] L. Fei-Fei, R. Fergus, P. Perona. "Learning generative visual models from few training examples : an incremental bayesian approach tested on 101 objects categories". In *Computer Vision and Pattern Recognition, Workshop on Generative-Model Based Vision (CVPRW)*, Volume 12, Washington, DC, USA, (2004).

[42] L. Fei-Fei, P. Perona. "A bayesian hierarchical model for learning natural scene categories". In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 524–531, Washington, DC, USA, (2005).

[43] L. Fei-Fei, R. VanRullen, C. Koch, P. Perona. "Rapid natural scene categorization in the near absence of attention". Volume 99, pages 9596–9601, (juillet 2002).

[44] C. Fellbaum. *WordNet - An Electronic Lexical Database*. Bradford books, (1998).

[45] S. Feng, V. Lavrenko, R. Manmatha. "Multiple bernoulli relevance models for image and video annotation". In *Proceedings of the IEEE CVPR Conference on Computer Vision and Pattern Recognition*, pages 1002–1009, Washington, DC, USA, (2004).

[46] A. Ferencz, E. G. Learned-Miller, J. Malik. "Building a classification cascade for visual identification from one example". In *ICCV '05 : Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, pages 286–293, Washington, DC, USA, (2005). IEEE Computer Society.

[47] R. Fergus, L. Fei-Fei, P. Perona, A. Zisserman. "Learning object categories from google's image search". In *Proceedings of the 10th International Conference on Computer Vision*, Volume 2, pages 1816–1823, (octobre 2005).

[48] R. Fergus, P. Perona, A. Zisserman. "Object class recognition by unsupervised scale-invariant learning". In *IEEE Conference on Computer Vision and Pattern Recognition*, Volume 2, pages 264–271, Madison, Wisconsin, USA, (2003).

[49] R. Fergus, P. Perona, A. Zisserman. "A visual category filter for google images", pages 242–256, (2004).

[50] J. Fournier. *Indexation d'images par le contenu et recherche interactive dans les bases généralistes*. PhD thesis, Université de Cergy-Pontoise, (octobre 2002).

[51] Y. Freund. "Boosting a weak learning algorithm by majority". In *Proceedings of the Third Annual Workshop on Computational Learning Theory, COLT*, University of Rochester, Rochester, NY, USA, (1990). Morgan Kaufmann Publishers.

[52] J. Friedman. "Another approach to polychotomous classification". Technical report, Stanford University, Department of Statistics, (1996).

[53] G. Furnas, T. Landauert, L. Gomez, S. Dumais. "The vocabulary problem in human-system communication", *Communications of the Association for Computing Machinery*, **30**, pp. 964–971, (1987).

[54] N. García-Pedrajas, D. Ortiz-Boyer. "Improving multiclass pattern recognition by the combination of two strategies", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**(6), pp. 1001–1006, (2006).

[55] M. M. Gorkani, R. W. Picard. "Texture orientation for sorting photos "at a glance"". In *Proceedings of the 12th International Conference on Pattern Recognition (ICPR)*, Volume 1, pages 459–464, Jerusalem, Israel, (1994).

[56] G. Grefenstette. "The color of things : towards the automatic acquisition of information for a descriptive dictionary", *Revue française de linguistique appliqué*, **10**(2), pp. 83–94, (2005).

[57] G. Griffin, A. Holub, P. Perona. "Caltech-256 object category dataset". Technical Report 7694, California Institute of Technology, (2007).

[58] M. Grubinger, P. Clough, H. Müller, T. Deselaers. "The IAPR TC-12 benchmark : a new evaluation resource for visual information systems". In *International Workshop OntoImage'2006 Language Resources for Content-Based Image Retrieval*, (2006).

[59] A. Hanbury. "Review of image annotation for the evaluation of computer vision algorithms". Technical Report PRIP-TR-102, PRIP, TU Wien, (2006).

[60] C. Harris, M. Stephens. "A combined corner and edge detector". In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, Manchester, Royaume-Uni, (1988).

[61] T. Hastie, R. Tibshirani. "Classification by pairwise coupling". In M. I. Jordan, M. J. Kearns, S. A. Solla, editors, *Advances in Neural Information Processing Systems, NIPS*, Volume 10, Denver, Colorado, USA, (1998). The MIT Press.

[62] R. Herbrich, T. Graepel, C. Campbell. "Bayes point machines", *Journal of Machine Learning Research*, **1**, pp. 245–279, (2001).

[63] L. Hollink, G. Schreiber, B. Wielinga. "Query expansion for image content search", (2006).

[64] P. Howarth, S. Rüger. "Evaluation of texture features for content-based image retrieval". In *Proceedings of the International Conference on Image and Video Retrieval, CIVR*, pages 326–334, Dublin, Ireland, (2004).

[65] C. Hsu, C. Lin. "A comparison of methods for multi-class support vector machines". Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, (2001).

[66] C. Hudelot. *Towards a cognitive vision platform for semantic image interpretation ; application to the recognition of biological organisms.* PhD thesis, Université de Nice - Sophia Antipolis, (avril 2005).

[67] J. Jeon, V. Lavrenko, R. Manmatha. "Automatic image annotation and retrieval using cross-media relevance models". In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 119–126, Toronto, Canada, (2003).

[68] Y. Jin, L. Khan, L. Wang, M. Awad. "Image annotations by combining multiple evidence & wordnet". In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 706–715, New York, NY, USA, (2005). ACM Press.

[69] T. Kadir, M. Brady. "Saliency, scale and image description", *International Journal of Computer Vision*, **45**(2), pp. 83–105, (novembre 2001).

[70] S. Knerr, L. Personnaz, G. Dreyfus. "Single-layer learning revisited : a stepwise procedure for building and training a neural network". In N. Y. S.-V. J.Fogelman, editor, *Neurocomputing : Algorithms, Architectures, and Applications*, (1990).

[71] E. Kong, T. Diettrich. "Why error-correcting output coding works with decision trees". Technical report, Departement of Computer Science, Oregon State University, Corvallis, (1995).

[72] V. Lavrenko, R. Manmatha, J. Jeon. "A model for learning the semantics of pictures". In S. Thrun, L. Saul, B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, (2004).

[73] S. Lazebnik, C. Schmid, J. Ponce. "Beyond bags of features : Spatial pyramid matching for recognizing natural scene categories". In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2169–2178, New York, USA, (2006).

[74] J. Li, J. Z. Wang. "Automatic linguistic indexing of pictures by a statistical modeling approach", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25**(9), pp. 1075–1088, (2003).

[75] J. Li, J. Z. Wang. "Real-time computerized annotation of pictures". In *Proceedings of the ACM Multimedia Conference*, pages 911–920, Santa Barbara, CA, USA, (octobre 2006).

[76] Y. Li, L. G. Shapiro, J. A. Bilmes. "A generative/discriminative learning algorithm for image classification". In *IEEE International Conference on Computer Vision (ICCV)*, Volume 2, pages 1605–1612, Pékin, Chine, (2005).

[77] S. Liapis, E. Sifakis, G. Tziritas. "Colour and texture segmentation using wavelet frame analysis, deterministic relaxation, and fast marching algorithms", *IEEE Transactions on Multimedia*, **6**(5), pp. 676–686, (octobre 2004).

[78] R. Lienhart, A. Hartmann. "Classifying images on the web automatically", *Journal of Electronic Imaging*, **11**(4), pp. 445–454, (octobre 2002).

[79] R. Lienhart, A. Kuranov, V. Pisarevsky. "Empirical analysis of detection cascades of boosted classifiers for rapid object detection". In *DAGM-Symposium*, pages 297–304, (2003).

[80] J.-H. Lim, J. S.Jin. "Discovering recurrent image semantics from class discrimination", *EURASIP Journal on Applied Signal Processing*, pages 1–11, (2006).

[81] W.-H. Lin, R. Jin, A. Hauptmann. "Web image retrieval re-ranking with relevance model". In *Proceedings of the IEEE/WIC International Conference on Web Intelligence*, pages 242–248, Pékin, Chine, (octobre 2003).

[82] Y. Liu, D. Zhang, G. Lu, W.-Y. Ma. "A survey of content-based image retrieval with high-level semantics", *Pattern Recognition*, **40**, pp. 262–282, (2007).

[83] D. Lowe. "Distinctive image features from scale-invariant keypoints", (2003).

[84] J. Luo, A. Savakis. "Indoor vs outdoor classification of consumer photographs using low-level and semantic cues". In *Proceedings of IEEE International Conference on Image Processing*, Thessaloniki, Grèce, (octobre 2001).

[85] J. Luo, A. E. Savakis, A. Singhal. "A bayesian network-based framework for semantic image understanding", *Pattern Recognition*, **38**(6), pp. 919–934, (2005).

[86] J. Luo, A. Singhal, S. P. Etz, R. T. Gray. "A computational approach to determination of main subject regions in photographic images", *Image and Vision Computing*, **22**(3), pp. 227–241, (2004).

[87] N. Maillot. *Ontology based object learning and recognition.* PhD thesis, Université de Nice - Sophia Antipolis, (décembre 2005).

[88] J. Mao, A. K. Jain. "Texture classification and segmentation using multiresolution simultaneous autoregressive models", *Pattern Recognition*, **25**(2), pp. 173–188, (1992).

[89] B. Marcotegui, S. Beucher. "Fast implementation of waterfall based on graphs". In C. Ronse, L. Najman, E. Decencière, editors, *Mathematical morphology : 40 years on*, Volume 30 of *Computational Imaging and Vision*, pages 177–186. Springer-Verlag, Dordrecht, (2005).

[90] R. Marée. *Classification automatique d'images par arbres de décision.* PhD thesis, Université de Liège, (2005).

[91] O. Maron, T. Lozano-Perez. "A framework for multiple-instance learning". In *Proceedings of Neural Information Processing Systems, NIPS*, Volume 10, pages 570–576, Denver, CO, USA, (1997).

[92] J. Martínez-Otzeta, B. Sierra, E. Lazkano, A. Astigarraga. "Classifier hierarchy learning by means of genetic algorithms", *Pattern Recognition Letters*, **27**(16), pp. 1998–2004, (12 2006).

[93] P. Matsakis. *Relations spatiales structurelles et interprétation d'images.* PhD thesis, Université Paul Sabatier, (1998).

[94] P. Matsakis, S.Andrefouet. "The fuzzy line between among and surround". In *Proceedings of the 2002 IEEE International Conference on Fuzzy Systems FUZZ-IEEE'02*, Volume 2, pages 1596–1601, Honolulu, HI, USA, (mai 2002).

[95] D. Metzler, R. Manmatha. "An inference network approach to image retrieval". In *Proceedings of the International Conference on Image and Video Retrieval, CIVR*, pages 42–50, Dublin, Ireland, (juillet 2004).

[96] V. Mezaris, I. Kompatsiaris, M. Strintzis. "An ontology approach to object-based image retrieval". In *Proceedings of the IEEE International Conference on Image Processing, ICIP03*, Volume 2, pages 511–514, Barcelona, Espagne, (septembre 2003).

[97] G. A. Miller. "Wordnet : a lexical database for english", *Commununications of the ACM*, **38**(11), pp. 39–41, (1995).

[98] C. Millet, I. Bloch, A. Popescu. "Using the knowledge of object colors to segment images and improve web image search". In *Proceedings of Recherche d'Information Assistée par Ordinateur, RIAO*, (2007).

[99] C. Millet, G. Grefenstette, I. Bloch, P.-A. Moëllic, P. Hède. "Automatically populating an image ontology and semantic color filtering". In *International Workshop Ontoimage'2006 Language Resources for Content-Based Image Retrieval*, pages 34–39, Gènes, Italie, (2006).

[100] K. Miyajima, A. Ralescu. "Spatial organization in 2d segmented images : representation and recognition of primitive spatial relations", *Fuzzy Sets and Systems : Special issue on fuzzy methods for computer vision and pattern recognition*, **65**, pp. 225–236, (1994).

[101] P.-A. Moëllic, C. Fluhr. "Imageval 2006 official campaign". Technical report, CEA List, (2007).

[102] P.-A. Moëllic, P. Hède, G. Grefenstette, C. Millet. "Evaluating content based image retrieval techniques with the one million images clic testbed". In *Proceedings of the Second World Enformatika Congress, WEC'05*, pages 171–174, Istanbul, Turquie, (février 2005).

[103] A. Mojsilovic, J. Gomes, B. Rogowitz. "Isee : perceptual features for image library navigation". In *Proceedings SPIE Human Vision and Electronic Imaging*, Volume 4662, pages 266–277, San Jose, Californie, (2002).

[104] Y. Mori, H. Takahashi, R. Oka. "Image-to-word transformation based on dividing and vector quantizing images with words". In *Proceedings of the First International Workshop on Multimedia Intelligent Storage and Retrieval Management, MISRM'99*, New Orleans, Louisiane, USA, (1999).

[105] H. Müller, S. Marchand-Maillet, T. Pun. "The truth about corel – evaluation in image retrieval". In *Proceedings of The Challenge of Image and Video Retrieval, CIVR*, pages 38–49, London, Royaume-Uni, (juillet 2002).

[106] T. Ojala, M. Pietikäinen, T. Mäenpää. "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(7), pp. 971–987, (2002).

[107] A. Oliva, A. B. Torralba. "Modeling the shape of the scene : a holistic representation of the spatial envelope", *International Journal of Computer Vision*, **42**(3), pp. 145–175, (2001).

[108] A. Oliva, A. B. Torralba. "Scene-centered representation from spatial envelope descriptors". In *Proceedings of Biologically Motivated Computer Vision*, (2002).

[109] J.-F. Omhover, M. Detyniecki. "Fast gradual matching measure for image retrieval based on visual similarity and spatial relations", *International Journal of Intelligent Systems*, **21**(7), pp. 711–723, (juin 2006).

[110] S. Paek, S.-F. Chang. "A knowledge engineering approach for image classification based on probabilistic reasoning systems". In *Proceedings of the IEEE International Conference on Multimedia and Expo*, Volume 2, pages 1133–1136, New York, USA, (2000).

[111] P. J. Phillips, P. J. Rauss, S. Z. Der. "Feret recognition algorithm development and test results". Technical Report 995, Army Research Lab, (octobre 1996).

[112] G. Pitel, C. Millet, G. Grefenstette. "Deriving a priori co-occurrence probability estimates for object recognition from social networks and text processing". In *3rd International Symposium on Visual Computing (ISVC07)*, Lake Tahoe, Nevada/Californie, USA, (novembre 2007).

[113] J. Platt, N. Cristianini, J. Shawe-Taylor. "Large margin dags for multiclass classification", pages 547–553, (2000).

[114] A. Popescu. "Image retrieval using a multilingual ontology". In *Proceedings of Recherche d'Information Assistée par Ordinateur, RIAO*, (2007).

[115] A. Quattoni, M. Collins, T. Darrell. "Conditional random fields for object recognition", pages 1097–1104, (2004).

[116] A. Quattoni, M. Collins, T. Darrell. "Incorporating semantics constraints into a discriminative categorization and labelling model". In *International Workshop on Semantic Knowledge in Computer Vision*, pages 1877–1877, (2005).

[117] P. Quelhas, F. Monay, J. Odobez, D. Gatica-Perez, T. Tuytelaars, L. V. Gool. "Modeling scenes with local descriptors and latent aspects". In *International Conference on Computer Vision*, pages 883–890, Pékin, Chine, (2005).

[118] F. Rossant, I. Bloch. "A fuzzy model for optical recognition of musical scores", *Fuzzy sets and systems*, **141**, pp. 165–201, (2004).

[119] B. C. Russel, A. A. Efros, J. Sivic, W. T. Freeman, A. Zisserman. "Using multiple segmentations to discover objects and their extent in image collections". In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPRS*, Volume 2, pages 1605–1614, New York, NY, USA, (2006).

[120] R. Schapire. "Strength of weak learnability", *Journal of Machine Learning*, **5**, pp. 197–227, (1990).

[121] R. Schettini, C. Brambilla, C. Cusano, G. Ciocca. "Automatic classification of digital photographs based on decision forests", *International Journal of Pattern Recognition and Artificial Intelligence, IJPRAI*, **18**(5), pp. 819–845, (2004).

[122] N. Serrano, A. Savakis, J. Luo. "A computationally efficient approach to indoor/outdoor scene classification". In *Proceedings of International Conference on Pattern Recognition, ICPR*, pages 40–46, Quebec, Canada, (août 2002).

[123] N. Serrano, A. E. Savakis, J. Luo. "Improved scene classification using efficient low-level features and semantic cues", *Pattern Recognition*, **37**(9), pp. 1773–1784, (2004).

[124] J. Sivic. *Efficient visual search of images and videos.* PhD thesis, University of Oxford, (2006).

[125] J. Sivic, B. Russell, A. A. Efros, A. Zisserman, B. Freeman. "Discovering objects and their location in images". In *International Conference on Computer Vision, ICCV*, pages 370–377, Pékin, Chine, (octobre 2005).

[126] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain. "Content-based image retrieval at the end of the early years", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(12), pp. 1349–1380, (2000).

[127] P. Stanchev, D. G. Jr., B. Dimitrov. "High level color similarity retrieval", *International Journal of Information Theorie Applications*, **10**(3), pp. 363–369, (2003).

[128] R. O. Stehling, M. A. Nascimento, A. X. Falcão. "A compact and efficient image retrieval approach based on border/interior pixel classification". In *Proceedings of the eleventh international conference on Information and knowledge management, CIKM*, pages 102–109, New York, NY, USA, (2002). ACM Press.

[129] J. Sunderland. "Image collections : librarians, users and their needs", *Art Libraries Journal*, **7**(2), pp. 41–49, (1982).

[130] C. Swain, T. Chen. "Defocus-based image segmentation". In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, Volume 4, pages 2403–2406, Detroit, MI, USA, (mai 1995).

[131] M. J. Swain, C. Frankel, V. Athitsos. "Webseer : An image search engine for the world wide web". Technical Report TR-96-14, University of Chicago Department of Computer Science, (31 1996).

[132] M. Szummer, R. W. Picard. "Indoor-outdoor image classification". In *IEEE International Workshop on Content-based Access of Image and Video Databases, in conjunction with ICCV*, pages 42–51, Bombay, Inde, (1998).

[133] H. Tamura. "Texture features corresponding to visual perception.", *IEEE Transactions on Systems, Man and Cybernetics*, **8**(6), pp. 460–473, (1978).

[134] S. Thorpe, D. Fize, C. Marlot. "Speed of processing in the human visual system", *Nature*, **381**, pp. 520–522, (juin 1996).

[135] S. Tollari. *Indexation et recherche d'images par fusion d'informations textuelles et visuelles.* PhD thesis, Université du Sud Toulon-Var, (octobre 2006).

[136] A. Torralba. "Contextual priming for object detection", *International Journal of Computer Vision, IJCV*, **53**(2), pp. 169–191, (2003).

[137] A. Torralba, A. Oliva. "Semantic organization of scenes using discriminant structural templates". In *International Conference on Computer Vision, ICCV*, pages 1253–1258, Korfu, Grèce, (1999).

[138] A. M. Treisman, G. Gelade. "A feature-integration theory of attention", *Cognitive Psychology*, **12**(1), pp. 97–136, (1980).

[139] A. Vailaya, M. Figueiredo, A. Jain, H. Zhang. "A bayesian framework for semantic classification of outdoor vacation images". In *Proceedings of the SPIE Conference on Electronic Imaging*, pages 415–426, San Jose, Californie, (1999).

[140] A. Vailaya, M. A. T. Figueiredo, A. K. Jain, H. Zhang. "Content-based hierarchical classification of vacation images". In *International Conference on Multimedia Computing and Systems, ICMCS*, Volume 1, pages 518–523, Florence, Italie, (1999).

[141] A. Vailaya, M. A. T. Figueiredo, A. K. Jain, H. Zhang. "Image classification for content-based indexing", *IEEE Transactions on Image Processing*, **10**(1), (janvier 2001).

[142] A. Vailaya, A. K. Jain, H. Zhang. "On image classification : city images vs. landscapes", *Pattern Recognition*, **31**(12), pp. 1921–1935, (1998).

[143] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, (1995).

[144] P. Viola, M. Jones. "Robust real-time object detection", *International Journal of Computer Vision, ICVR*, (2002).

[145] J. Vogel. *Semantic scene modeling and retrieval*. PhD thesis, Hartung-Gorre Verlag Konstanz, (octobre 2004).

[146] J. Vogel, B. Schiele. "Natural scene retrieval based on a semantic modeling step". In *Conference on Image and Video Retrieval, CIVR*, pages 207–215, Dublin, Ireland, (juillet 2004).

[147] L. von Ahn, L. Dabbish. "Labeling images with a computer game". In *Proceedings of the SIGCHI conference on Human factors in computing systems CHI '04*, pages 319–326, New York, NY, USA, (2004). ACM Press.

[148] K. Vézina. "Survol du monde de l'indexation des images", *Cursus*, **4**(1), (1998).

[149] J. Z. Wang, J. Li, G. Wiederhold. "SIMPLIcity : Semantics-sensitive integrated matching for picture LIbraries", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**(9), pp. 947–963, (2001).

[150] T. Windeatt, R. Ghaderi. "Coding and decoding strategies for multi-class learning problems", *Information Fusion*, **4**(1), pp. 11–21, (2003).

[151] C. Yang, M. Dong, F. Fotouhi. "Region based image annotation through multiple-instance learning". In *Proceedings of ACM International Conference on Multimedia*, pages 435–438, Singapour, (novembre 2005).

[152] A. Yavlinsky. "Behold : a content based image search engine for the world wide web". Technical report, Department of Computing, South Kensington Campus Imperial College London, London SW7 2AZ, UK, (2006).

[153] A. Yavlinsky, E. Schofield, S. M. Rüger. "Automated image annotation using global features and robust non-parametric density estimation". In *International Conference on Image and Video Retrieval, CIVR*, pages 507–517, Singapour, (2005).

[154] E. Yiu. "Image classification using color cues and texture orientation". Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, (1996).

[155] D. Zhang, A. Wong, M. Indrawan, G. Lu. "Using gabor texture features". In *Proceedings of the Pacific-Rim Conference on Multimedia*, pages 392–395, Hong Kong, (2000).

[156] S. Zinger, C. Millet, B. Mathieu, G. Grefenstette, P. Hède, P.-A. Moëllic. "Clustering and semantically filtering web images to create a large scale image ontology". In *Proceedings of the IS&T/SPIE 18th Symposium Electronic Imaging*, pages 89–97, San Jose, Californie, USA, (janvier 2006).