Hiérarchies sémantiques pour l'annotation multifacette d'images

Anne-Marie Tousch

▶ **To cite this version:**

Anne-Marie Tousch. Hiérarchies sémantiques pour l'annotation multifacette d'images. Traitement des images [eess.IV]. Ecole des Ponts ParisTech, 2010. Français. NNT : 2010ENPC1002 . pastel-00555122

HAL Id: pastel-00555122
https://pastel.hal.science/pastel-00555122

Submitted on 12 Jan 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Ecole des Ponts – ParisTech
ONERA-Dtim / LIGM-Imagine

Thèse

présentée pour l'obtention du grade de Docteur de l'Ecole des Ponts ParisTech,
spécialité « Mathématiques, Informatique »

par

Anne-Marie Tousch

Hiérarchies sémantiques pour l'annotation multifacette d'images

Thèse soutenue le devant le jury composé de :

M.	Matthieu Cord	Université Paris VI	(Rapporteur)
M.	Frédéric Jurie	Université de Caen	(Rapporteur)
M.	Jean-Yves Audibert	Ecole des Ponts ParisTech	(Examinateur)
M.	Stéphane Herbin	ONERA	(Examinateur)
M.	Henri Maître	Télécom ParisTech	(Président)
M.	Renaud Keriven	Ecole des Ponts ParisTech	(Directeur de thèse)

Titre :
Hiérarchies sémantiques pour l'annotation multifacette d'images

Résumé :
Cette thèse a pour sujet l'annotation automatique d'images. Pour plus de souplesse, nous utilisons un vocabulaire structuré, permettant de construire des annotations multifacettes et à différents niveaux d'interprétation. Une annotation prend alors la forme d'un ensemble de multilabels associés à des indices de confiance et permet d'exprimer un compromis fiabilité/précision sémantique.

Le traitement proposé se déroule en deux phases : extraction de caractéristiques informatives et calcul de probabilités normalisées sur un espace de multilabels. Chacune exploite des mécanismes d'apprentissage. La démarche est évaluée sur deux jeux de données : un ensemble d'images de voitures et la base d'objets génériques Caltech-101. Les résultats suggèrent d'utiliser le vocabulaire structuré à différentes étapes selon la nature des données.

Mots-clés :
Annotation d'image, reconnaissance d'objets, vocabulaires structurés, classification à facettes, apprentissage statistique

---

Title:
Semantic hierarchies for multi-faceted image annotation

Abstract:
In this thesis, we address the problem of automatic image annotation. For a more flexible system, we build multi-faceted annotations organized in a semantic hierarchy. Thus, an annotation is defined by a set of multilabels coupled with confidence levels. A tradeoff between reliability and semantic precision allows greater flexibility.

The proposed algorithm proceeds in two stages. First, informative image features are extracted. Second, normalized probabilities are computed on a set of multilabels. Both rely on statistical learning machines. We evaluate the approach on two datasets : a set of car images and a generic database, Caltech-101. Results show different behaviour depending on the data, suggesting that the vocabulary structure is useful at different stages of the algorithm.

Keywords:
Image annotation, object recognition, structured vocabularies, semantic hierarchies, faceted classification, machine learning

Remerciements

Mes premiers remerciements vont à Stéphane Herbin et à Jean-Yves Audibert, qui m'ont encadrée tout au long de cette thèse. Merci pour leur inspiration, leur soutien, et leur disponibilité qui n'ont pas discontinué pendant trois ans. Je tiens à remercier Renaud Keriven, pour avoir accepté de diriger ma thèse et m'avoir accueilli au sein de l'équipe IMAGINE. Je remercie également Frédéric Jurie et Matthieu Cord, pour avoir accepté d'être mes rapporteurs, et Henri Maître, pour avoir accepté de faire partie de mon jury.

Je voudrais ensuite remercier tous les autres doctorants de l'ONERA-DTIM et d'IMAGINE qui ont contribué à faire de cette thèse une expérience heureuse à la fois sur le plan scientifique et humain. En premier lieu, je ne peux que citer Antoine, avec qui j'ai partagé mon bureau pendant 3 ans à l'ONERA, à mi-temps. Merci pour sa patience, son écoute, et pour tous les débats philosophiques. Merci également à Adrien, notre plus proche voisin à tous égards, à Evangeline pour tous ses dépannages pratiques, à Walid pour les cours d'arabe, à Joseph pour ses goûters et ses échanges scientifiques, à Nadia pour sa bonne humeur, à Ibrahima, Aurélien, Christophe, Olivier et à tous les nouveaux que je n'aurai pas eu le temps de mieux connaître, malheureusement : Guillaume, Pauline, Laure, Paul. Je n'oublie pas Nicolas, dont la bonne humeur nous aura manqué sur la fin.

Merci aux doctorants de l'équipe IMAGINE, et d'abord à Anne-Laure pour son entrain continuel. Merci à Cédric et Alexandre pour les dépannages informatiques et pour la bonne ambiance qu'ils savent créer autour d'eux, à Maxime pour ses PhD Comics oubliés dans l'armoire, à Hiep parce que lui aussi a eu la patience de m'avoir comme collègue de bureau pendant presque 2 ans, à Nicolas, Ehsan, Jaonary, Jérôme, pour leur solidarité, et à David pour son enthousiasme.

Je remercie tous ceux qui se sont intéressés à mon travail, à IMAGINE comme à l'ONERA, et qui ont pris le temps d'échanger à ce sujet : Hichem, Hui, d'un côté, Bertrand de l'autre. Merci à Patrick et à Martial d'avoir pris le temps de résoudre mes nombreux soucis informatiques. Merci à Jonathan d'avoir pris le temps de relire un bon bout de mon manuscrit.

Merci à Brigitte, au CERTIS, à Françoise, à l'ONERA, et à Alice, à l'ENPC, pour leur aide dans tous les détails administratifs. Combien de fois vous ai-je bénies !

Enfin, merci à ma famille et à tous mes frères et soeurs de m'avoir aidée à garder l'équilibre pendant ce temps où le travail m'a souvent éloignée.

Sommaire

Sommaire vi

Notations xi

1 Introduction 1
- 1.1 Contexte et problématique . . . 1
- 1.2 Objectifs et démarche adoptée . . . 4
- 1.3 Organisation du manuscrit . . . 5

2 Annotation sémantique d'images : problématique et état de l'art 7
- 2.1 Le problème de l'annotation d'image : aspects théoriques . . . 8
  - 2.1.1 Comment annoter une image ? . . . 8
  - 2.1.2 Les besoins de l'utilisateur . . . 12
  - 2.1.3 Les effets de l'interaction . . . 13
  - 2.1.4 Annotation manuelle ? . . . 15
  - 2.1.5 Le fossé sémantique . . . 16
  - 2.1.6 Reconnaissance d'objets et vocabulaires . . . 17
- 2.2 Annotation d'images avec un vocabulaire non structuré . . . 23
  - 2.2.1 Introduction . . . 23
  - 2.2.2 Les approches directes . . . 24
  - 2.2.3 Les approches linguistiques . . . 24
  - 2.2.4 Les approches compositionnelles . . . 25
  - 2.2.5 Les approches structurelles . . . 25
  - 2.2.6 Les approches compositionnelles hiérarchiques . . . 26
  - 2.2.7 Les approches communicantes . . . 26
  - 2.2.8 Les approches hiérarchiques . . . 27
  - 2.2.9 Les approches multilabels . . . 28
  - 2.2.10 Conclusion . . . 28
- 2.3 Annotation d'images avec un vocabulaire structuré . . . 29
  - 2.3.1 Introduction . . . 29
  - 2.3.2 Les approches linguistiques . . . 31
  - 2.3.3 Les approches compositionnelles et structurelles . . . 32
  - 2.3.4 Les approches communicantes . . . 32
  - 2.3.5 Les approches hiérarchiques . . . 33
- 2.4 Evaluation . . . 34
- 2.5 Discussion . . . 35
- 2.6 Conclusion . . . 38

3 Bases d'images et descripteurs pour l'annotation multifacette hiérarchique 41
- 3.1 Une base d'images pour l'annotation multifacette hiérarchique 41
  - 3.1.1 Introduction . . . 41

3.1.2 Constitution de la base d'images de voitures . . . 42
3.2 Tests de référence . . . 44
3.2.1 Algorithmes de référence . . . 48
3.2.2 Performances sur la base de voitures . . . 51
3.2.3 Conclusion . . . 53
3.3 Extraction de signatures spécifiques . . . 55
3.3.1 Introduction . . . 55
3.3.2 Méthode . . . 56
3.3.3 Sélection de descripteurs discriminants . . . 59
3.3.4 Extraction efficace des caractéristiques . . . 64
3.3.5 Classification de détails . . . 65
3.4 Evaluation et discussion . . . 69
3.4.1 Evaluation . . . 69
3.4.2 Expériences . . . 70
3.4.3 Discussion . . . 74

4 Annotation multifacette hiérarchique d'images 79
4.1 Introduction et formalisation du problème . . . 79
4.1.1 Objectif . . . 79
4.1.2 Formalisation . . . 80
4.2 Remarques bibliographiques . . . 82
4.2.1 Exploitation de hiérarchies . . . 82
4.2.2 Exploitation de hiérarchies en vision . . . 84
4.2.3 Conclusion . . . 85
4.3 Méthode . . . 85
4.3.1 De la hiérarchie sémantique à l'hypergraphe . . . 86
4.3.2 Probabilité d'un multilabel . . . 90
4.3.3 Régularisation des probabilités . . . 92
4.3.4 Annotation multifacette hiérarchique . . . 94
4.3.5 Autres méthodes . . . 97
4.3.6 Comparaison avec une taxonomie visuelle . . . 98
4.3.7 Conclusion . . . 100
4.4 Evaluation d'annotations multifacettes hiérarchiques . . . 100
4.4.1 Problématique et objectifs . . . 100
4.4.2 Construction de la courbe erreur/complexité . . . 102
4.4.3 Fonctions de coût hiérarchiques . . . 104
4.4.4 Analyse de la courbe . . . 105
4.5 Expériences . . . 106
4.5.1 Influence du classifieur binaire et des paramètres . . . 106
4.5.2 Différents lissages de probabilités . . . 107
4.5.3 Comparaison des différentes méthodes hiérarchiques . . . 111
4.5.4 Comparaison avec Caltech-101 . . . 111
4.5.5 Comparaison de l'hypergraphe sémantique avec une taxonomie visuelle . . . 114
4.5.6 Résultats en recherche d'images . . . 117
4.6 Discussion et perspectives . . . 120

5 Sélection hiérarchique de caractéristiques 125
5.1 Introduction . . . 125
5.1.1 Remarques bibliographiques . . . 126
5.1.2 Problématique . . . 126
5.1.3 Démarche adoptée . . . 127

5.2 Sélection de caractéristiques – étude locale . . . 127
5.2.1 Algorithmes de sélection de caractéristiques . . . 127
5.2.2 Interprétation du classement . . . 129
5.2.3 Interprétation avec extensions des signatures . . . 129
5.3 Sélection hiérarchique de caractéristiques . . . 129
5.3.1 Méthodes ascendantes . . . 130
5.3.2 Méthode globale . . . 130
5.4 Expériences et évaluation . . . 134
5.4.1 Analyse qualitative . . . 134
5.4.2 Analyse quantitative . . . 143
5.5 Discussion et perspectives . . . 145

6 Conclusion **149**
6.1 Contributions . . . 149
6.2 Perspectives . . . 150

A Extraction de caractéristiques **153**
A.1 Introduction . . . 153
A.2 Descripteurs locaux . . . 153
A.2.1 Sélection de zones d'intérêts . . . 153
A.2.2 Descripteurs SIFT . . . 154
A.2.3 Histogrammes d'orientations de gradients . . . 155
A.2.4 Descripteurs de segments . . . 155
A.2.5 Image intégrale . . . 155
A.3 Combinaisons de descripteurs locaux . . . 156
A.3.1 sacs-de-mots . . . 156
A.3.2 Pyramides de descripteurs . . . 157
A.3.3 Hiérarchies de caractéristiques . . . 158

B Techniques d'apprentissage statistique **161**
B.1 Introduction . . . 161
B.1.1 Formalisation du problème . . . 161
B.1.2 La sélection de modèle . . . 161
B.2 Le classifieur Bayésien Naïf . . . 162
B.2.1 Principe . . . 162
B.2.2 Remarques bibliographiques . . . 163
B.3 Les *K*-plus-proches-voisins . . . 163
B.3.1 Principe . . . 163
B.3.2 Remarques bibliographiques . . . 164
B.4 Machines à Vecteurs Supports . . . 165
B.4.1 Formulation générale . . . 165
B.4.2 SVM probabilisés . . . 167
B.4.3 Implémentation . . . 170
B.4.4 Remarques bibliographiques . . . 170
B.5 La régression logistique . . . 170
B.5.1 Principe . . . 170
B.5.2 Lien avec les SVMs . . . 171
B.5.3 Résolution par IRLS . . . 171
B.5.4 Seconde formulation avec noyau . . . 172

C Résultats complémentaires **175**
C.1 Signatures . . . 175

C.2 Annotation multifacette hiérarchique 175
C.3 Sélection hiérarchique de caractéristiques 175

Glossaire **179**

Index **181**

Bibliographie **183**

Notations

$\prec$	$A \prec B$ signifie que un A "est un" B (Is-A), ou encore que toute instance de A est aussi une instance de B.	12
$\sqsubset$	la relation Part-Of : $A \sqsubset B$ si A est une partie de B	12
$\wedge$	"et" logique, mais aussi relation de co-occurence (objets trouvés dans une même image). $A \wedge B$ signifie que les labels A et B peuvent être assignés ensemble à une même image.	12
$\vee$	"ou" logique.	97
x	vecteur caractéristique, aussi appelé signature en association avec une image, en général $x \in \mathbb{R}^d$.	50
d	dimension de l'espace des caractéristiques.	49
y	label, en général $y \in \{-1, 1\}$ ou $y \in \{0, 1\}$.	80
$\mathbf{y}$	vecteur multilabel, $\mathbf{y} = \{y_1, \ldots, y_m\} \in \{0, 1\}^m$.	80
$\mathbf{t}$	vecteur multilabel correspondant à la vérité terrain.	81
$z_i^{(j)}$	label binaire $z_i^{(j)}$ au nœud j pour l'entrée x_i	90
n	taille de la base d'image ou de la base d'apprentissage, selon le contexte.	68
$[\cdot]$	$[n]$, où $n \in \mathbb{N}$, est l'ensemble des n premiers entiers. $[n] = \{1, \ldots, n\}$.	57
$\lvert \cdot \rvert$	$\lvert A \rvert$ est le nombre d'éléments de A, où A est un ensemble fini.	80
$[\![\cdot]\!]$	$[\![f]\!]$, où f est une proposition, est la fonction binaire qui vaut 1 si f est vraie et 0 si f est fausse.	69
$\oplus$	$\mathbf{y}_1 \oplus \mathbf{y}_2$ est l'opération binaire XOR, qui correspond au nombre de labels différents entre deux multilabels.	104
$\mathcal{C}(\mathbf{y})$	complexité d'un multilabel $\mathbf{y}$, égale au nombre de labels qu'il contient, i.e. $\mathcal{C}(\mathbf{y}) = \lvert \mathbf{y} \rvert$.	81
$\mathcal{G}$	désigne le graphe Is-A, décrivant les liens d'hypernymie/hyponymie entre labels	82
$\mathcal{H}$	désigne le diagramme de Hasse, graphe représentant l'hypergraphe décrivant les liens entre les multilabels consistants avec un graphe $\mathcal{G}$.	81

$\hat{\mathcal{H}}$	ensemble des feuilles de $\mathcal{H}$.	82
$fils(\cdot)$	ensemble des fils d'un nœud, dans $\mathcal{G}$ ou dans $\mathcal{H}$.	81
$par(\cdot)$	ensemble des parents d'un nœud	81
$desc(\cdot)$	ensemble des descendants d'un nœud	81
$anc(\cdot)$	ensemble des ancêtres d'un nœud	81
$feuilles(\cdot)$	Soit $\mathbf{y} \in \mathcal{H}$,$feuilles(\mathbf{y})$ est l'ensemble des feuilles descendantes de $\mathbf{y}$, i.e. $feuilles(\mathbf{y}) = \hat{\mathcal{H}} \cap desc(\mathbf{y})$	93

Introduction

1

1.1 Contexte et problématique

L'image est désormais à la portée de tous, et, avec le développement du web social, partagée par tous. Ainsi, en novembre 2009, Facebook annonce plus de 2 milliards de photos mises en ligne chaque mois [58]. Sur un autre plan, les caméras de vidéo-surveillance se multiplient : si en France, elles se comptent encore "seulement" en centaines de milliers, le Royaume-Uni en comptait déjà 4,2 millions en 2009 [106]. Toutes ces images représentent de l'information. Beaucoup d'information. Pour les organiser, le moyen le plus classique actuellement est de demander à l'utilisateur d'associer aux images quelques mots, ou *tags*, décrivant leur contenu. En associant aux images des *métadonnées*, on peut remplacer les données numériques par un résumé textuel de leur contenu informatif. Alors que les banques d'images prennent des dimensions gigantesques, et que les heures de vidéos accumulées par les caméras sont quasiment inexploitables, il devient capital d'automatiser la tâche d'extraction de ces métadonnées.

Cependant, bien que l'ordinateur soit un outil de calcul performant, extraire l'information d'une image est un problème qui est encore bien loin d'être résolu. La recherche est très active et a fourni de grandes avancées, traduites en particulier par l'introduction de moteurs de recherche dits "visuels", maintenant accessibles à tous (par exemple avec Exalead, Bing, Google). En combinant l'information textuelle donnée par des opérateurs humains à des calculs de similarité visuelle, ces moteurs offrent déjà des performances intéressantes, sans pourtant résoudre le fond du problème : extraire automatiquement les métadonnées, i.e. le contenu sémantique de l'image. Une difficulté supplémentaire apparaît en outre avec le fait qu'une simple image peut avoir de multiples interprétations, et un mot plusieurs sens.

L'annotation d'images se trouve au point de convergence de deux domaines de recherche voisins : la reconnaissance d'objets et la recherche d'images par le contenu. Le premier s'est attaché très tôt à reconnaître le contenu des images, mais souvent dans des applications assez contraintes (processus industriels... [134]). Le second a eu pour objectif d'organiser les images par rapport à leur similarité visuelle, sans tenir compte des objets représentés [169]. Dès le tournant des années 2000, il est devenu clair que la similarité visuelle était insuffisante, et qu'il était nécessaire de chercher plutôt des similarités conceptuelles, c'est-à-dire d'annoter les images [159]. Du côté de la reconnaissance d'objets, il s'est agi de reconnaître de plus en plus d'objets différents, et dans des conditions moins contraintes. Dans les deux cas, les chercheurs ont été confrontés à la même difficulté : l'ordinateur n'est pas capable d'interpréter automatiquement le contenu des images à la manière du cerveau humain. Ce défi est appelé le *fossé sémantique*, et il est au cœur de tout problème d'annotation d'images.

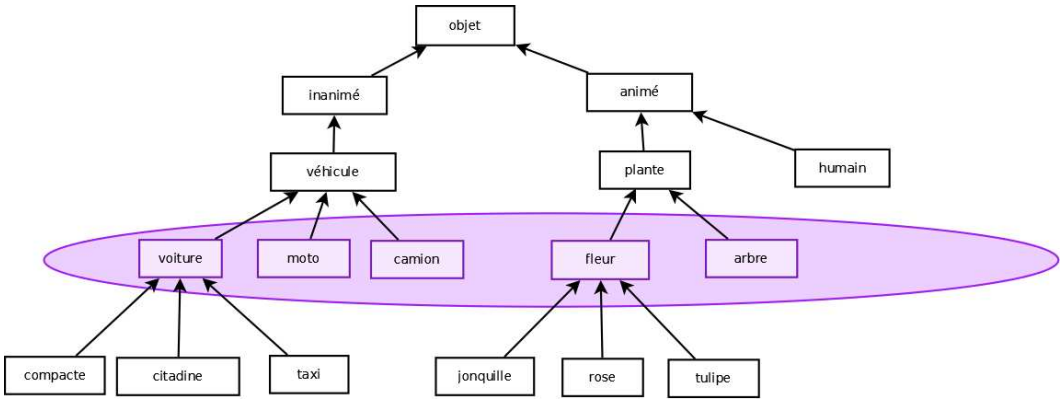


FIGURE 1.1 – *Exemple de hiérarchie sémantique. Le niveau de reconnaissance de base, en violet, est appelé niveau fondamental.*

De nombreuses approches ont été proposées pour franchir ce fossé. La sélection de bases d'images adéquates est une étape cruciale (Ponce et al. [148]). En reconnaissance d'objets, de nouvelles bases sont proposées régulièrement, de difficultés croissantes : lorsque le nombre de catégories est moyen, de l'ordre de la centaine, on atteint déjà de très bonnes performances (Bosch et al. [27]). Les dernières bases multiplient le nombre d'images et le nombre de catégories (par exemple IMAGENET [46]). Cependant, ces bases et les algorithmes associés proposent en général une interprétation univoque d'une image : un unique label est associé à chaque image. Un premier pas pour dépasser cette limitation a été franchi en introduisant un aspect multilabel : non seulement en annotant les différentes parties de l'image (Barnard et al. [16]), mais en associant différents labels à un même objet (Boutell et al. [30], Carneiro et al. [32]).

Dans un autre contexte, les algorithmes utilisés pour la classification de documents textuels s'appuient de plus en plus sur des *hiérarchies sémantiques*, c'est-à-dire des représentations structurées du vocabulaire. La figure 1.1 donne un exemple de hiérarchie. L'idée de traduire les images par des concepts sémantiques fait l'objet de plus en plus de recherches. Pour montrer cette évolution de tendance dans la recherche sur les images, nous avons compté le nombre de réponses renvoyées par Google-Scholar lorsqu'on lui donne certains mots-clés. Les résultats sont rapportés figure 1.2, et montrent que la recherche en annotation d'images en général est en plein essor. De plus, bien que les notions de "labels sémantiques" et de "hiérarchies sémantiques" soient encore relativement peu abordées dans la littérature, on observe un nombre fortement croissant d'articles à ce sujet sur ces dernières années (depuis 2004-2005 environ), illustrant un intérêt récent pour ces notions.

L'idée a été d'introduire ce genre de représentations pour aider à l'interprétation des images. Jusqu'à présent, elles l'ont été essentiellement de deux manières :

1. par analogie, en essayant de construire ce que l'on a appelé des *taxonomies visuelles*, avec par exemple les travaux récents de Ahuja et Todorovic [4], Bart et al. [18], Griffin et Perona [82], Marszałek et Schmid [127], Sivic et al. [167]. Dans ces taxonomies, les relations entre les nœuds représentent la similarité visuelle entre les catégories associées.
2. comme une enveloppe, en exploitant la structure pour relâcher les contraintes sur le vocabulaire disponible pour l'utilisateur, ou pour effectuer des traitements supplémentaires sur les annotations, sans changer le cœur du système d'annotation. Les travaux de Aslandogan et al. [9], Yang

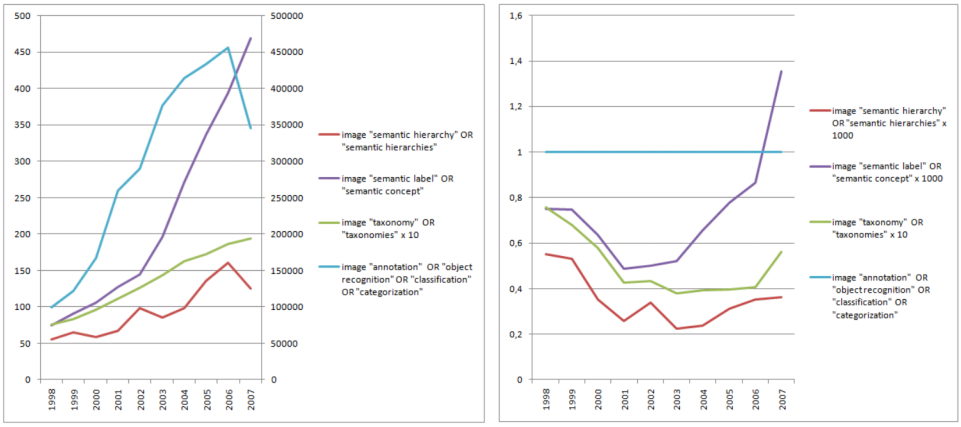


FIGURE 1.2 – *Tendance de la recherche à se tourner vers les aspects sémantiques : nombre de résultats renvoyés par requête, en fonction des années, entre 1998 et 2007. Les échelles ne sont pas toutes les mêmes pour une meilleure visualisation. La figure de gauche donne une évolution du nombre d'articles publiés, accessibles par Google-Scholar, et mentionnant au moins un des termes. Les 2 premières courbes correspondent à l'axe de gauche, et les 2 dernières à l'axe de droite. La figure de droite reprend les mêmes chiffres, et donne le rapport du nombre d'articles retournés pour une requête donnée par rapport à ceux retournés pour la requête plus générale «image "annotation" OR "object recognition" OR "classification" OR "categorization"». Pour plus de clarté, certaines courbes ont été multipliées par un facteur indiqué en légende. Requêtes effectuées le 25/09/09.*

et al. [195], Barnard et al. [16], Datta et al. [44], Lam et Singh [111], Popescu et al. [150] rentrent dans cette catégorie.

Des travaux très récents ont cherché à introduire les hiérarchies sémantiques d'une troisième manière, directement au niveau de l'apprentissage (Marszałek et Schmid [126], Torralba et al. [179], Fan et al. [61]). C'est dans cette catégorie d'approches que nous nous positionnerons. En effet, à notre connaissance, et au commencement de cette thèse, il n'existait pas de travaux exploitant la structure du vocabulaire à la fois pour relâcher la contrainte d'univocité des interprétations et pour orienter l'apprentissage. Les méthodes que nous venons de mentionner utilisent les hiérarchies en conservant les processus d'annotation ou de reconnaissance habituels : un objet, un label. Or un objet peut être catégorisé selon différents critères, nommés *facettes* dans les systèmes de classification de documents [152, 165]. Par ailleurs, les méthodes de l'état de l'art ne permettent pas de contrôler la précision de la réponse : l'algorithme annotera toujours l'image avec les mêmes labels, et les performances seront conditionnées en conséquence.

C'est ce qui justifie notre étude : il s'agit de développer une nouvelle méthode d'annotation permettant d'extraire une information **multifacette et hiérarchique** des images en s'appuyant sur des hiérarchies sémantiques, et en gérant la multiplicité des interprétations par un compromis précision/fiabilité. La notion de **multifacette** traduit le fait qu'un objet peut être décrit selon plusieurs points de vue indépendants. Prenons par exemple l'image de voiture de la figure 1.3. On peut dire que c'est une voiture "rouge", mais cela n'informe en rien sur le fait que c'est une "Citroën". De même, savoir que c'est une "Citroën" ne dit rien sur le segment automobile ("citadine"). Ces informations sont des interprétations liées à ce que nous appellerons différentes facettes. La notion de **hiérarchie** est plus intuitive : elle traduit le fait que certaines interprétations en impliquent d'autres. Pour reprendre l'exemple de la figure 1.3, si on dit que cette voiture est une "C3", on attribue implicitement les labels "Citroën" et "citadine". Ces labels sont liés par des relations de hiérarchie. La notion de **précision sémantique** en découle :

FIGURE 1.3 – *Exemple d'une image de voiture pouvant prêter à diverses interprétations, selon le point de vue : c'est une voiture rouge ; une Citroën ; une citadine ...*

plus un label se situe vers le bas de la hiérarchie, plus il est *précis sémantiquement*. Par exemple, "citadine" est un label plus précis que "voiture" (voir figure 1.1).

1.2 Objectifs et démarche adoptée

Notre étude s'intéresse à l'interprétation sémantique des images, dans la perspective de création de métadonnées. Par rapport aux fonctions d'interprétation décrites dans la littérature, notre approche se distingue par les deux objectifs suivants :

- la possibilité de produire des métadonnées de manière non univoque par le contrôle d'un compromis fiabilité/précision sémantique ;
- l'utilisation d'une structure de représentation des connaissances d'un domaine donné contraignant la distribution des métadonnées.

Nous nous intéressons dans un premier temps au choix d'une base de données, le but étant (a) de travailler sur une base d'images dont les annotations correspondent à un problème multifacette hiérarchique, et (b) d'extraire des caractéristiques images suffisamment informatives pour être exploitables pour l'annotation. La figure 1.4 illustre le type de données et le domaine visé. Nous nous intéressons à la caractérisation d'images de voitures prises de trois quart face. La difficulté de ce problème tient à la non maîtrise des conditions d'illumination, et à la grande variété des types d'objets. Une fois la base d'images constituée, nous nous penchons sur l'extraction de caractéristiques, qui constitue un élément indispensable à tout problème de reconnaissance d'objet. Elle a une influence décisive sur les performances finales du système, et nous lui apportons donc un soin particulier.

Dans un second temps, l'objectif est de développer le système d'annotation multifacette hiérarchique, en l'appliquant sur plusieurs bases d'images, dont celle des voitures. Dans ce contexte, le type de description attendue est illustré par le tableau 1.1 dans le cas où l'image contient par exemple une 206 3-portes. Il s'agit d'annoter les images par une distribution de multilabels, associés à des

FIGURE 1.4 – *Quelques exemples d'images de la base de données, représentant 7 catégories différentes, de haut en bas et de gauche à droite, respectivement : Corsa I 3-portes, Zafira I, C3, C3, 206 3-portes, 206 5-portes, Twingo I, Clio II-1 5-portes*

Multilabel	Confiance
Berline	0.78
Citadine	0.66
Berline, Citadine	0.6
Peugeot	0.55
Berline, Peugeot	0.5
Berline, Citadine, Peugeot	0.45
Berline, Citadine, Peugeot, 206	0.4
Berline, Citadine, Peugeot, 206, 3 portes	0.2

TABLE 1.1 – *Exemple de type de description attendue pour une image de 206 3 portes. Les valeurs indiquées ici sont indicatives.*

niveaux de confiance. Les multilabels énoncés sont conformes à la structure de représentation des connaissances, c'est-à-dire aux hiérarchies sémantiques. Les niveaux de confiance permettent le contrôle du compromis précision/fiabilité.

Ce type d'annotation étant particulier, il s'agira également de développer une méthode d'évaluation des performances adaptée.

Enfin, dans un dernier temps, nous étudierons comment appliquer les méthodes de sélection de caractéristiques dans les hiérarchies sémantiques, dans le but d'améliorer les performances de la méthode d'annotation et de prévoir certaines optimisations éventuelles.

1.3 Organisation du manuscrit

Le manuscrit est organisé de la manière suivante. Dans le chapitre 2, l'objectif sera de montrer (i) l'importance des aspects multifacette et hiérarchique des annotations, leur lien avec les hiérarchies sémantiques, (ii) la façon dont ces aspects ont été abordés dans l'état de l'art. Ce chapitre permettra également de préciser les objectifs.

Dans le chapitre 3, nous introduirons une nouvelle base de données spécifiquement conçue pour l'annotation multifacette hiérarchique. Nous verrons comment extraire des caractéristiques images performantes pour ces données, l'objectif étant de se donner les outils préalables à la conception du système d'annotation.

Le chapitre 4 correspond au cœur de notre étude. L'objectif est de développer une méthode d'annotation intégrant des contraintes de hiérarchie et d'exclusion dans un modèle multilabel "classique", et permettant le contrôle d'un compromis

précision/fiabilité. Nous présenterons aussi la méthode d'évaluation associée à ces annotations.

Le chapitre 5 s'intéresse au problème plus particulier de la sélection hiérarchique de caractéristiques. L'idée n'est pas simplement d'améliorer les performances, mais de fournir des éléments favorisant l'interprétabilité des classifieurs.

Les annexes A et B permettent de compléter l'état de l'art sur un plan plus technique, en reprenant les méthodes d'extraction de caractéristiques (A), et les bases des techniques d'apprentissage statistique (B). Quelques résultats complémentaires sont fournis en annexe C.

Les travaux présentés dans les chapitres 3 et 4 ont fait l'objet d'une communication internationale à la conférence MIR (Multimedia Image Retrieval) en 2008 (Tousch et al. [181]). Dans un contexte de vidéosurveillance, ils ont également été résumés dans un article de la Revue de l'Electricité et de l'Electronique (Herbin et al. [91]). L'étude présentée au chapitre 2 fera l'objet d'une prochaine soumission.

2 Annotation sémantique d'images : problématique et état de l'art

Il y a besoin d'annotations

Les images numériques sont maintenant utilisées dans toutes sortes de circonstances, dans un cadre professionnel ou privé, de manière artistique ou fonctionnelle. Devant les masses de données à leur disposition, les utilisateurs n'ont pas tous les mêmes besoins, ni les mêmes exigences. Alors qu'un journaliste cherchera une image correspondant à un événement particulier, un voyageur voudra organiser ses photos par thèmes, ou par lieux. En vidéo-surveillance, on pourra compter les clients entrant dans un magasin, ou sélectionner les images contenant un véhicule particulier. Actuellement ces opérations sont extrêmement coûteuses en temps de travail humain. Une solution, c'est que l'ordinateur soit capable d'analyser les images, c'est-à-dire de comprendre automatiquement leur contenu.

C'est l'objectif de l'annotation d'image : associer des termes à l'image, supposés traduire le contenu de l'image sous une forme facilement interprétable par un humain.

Dans ce chapitre, nous étudierons les caractéristiques souhaitables d'un système d'annotation automatique d'image (section 2.1), et comment l'annotation est faite en pratique (sections 2.2 et 2.3). Dans la section 2.1.1, nous verrons que l'information concernant une image peut être décrite à plusieurs niveaux de précision, et selon plusieurs approches. Pour savoir à quel niveau annoter l'image, il convient donc d'étudier plus précisément quels sont les besoins et les comportements des utilisateurs, ce qui fera l'objet de la section 2.1.2 et de la section 2.1.3. Nous préciserons pourquoi il est préférable de recourir à une annotation automatique dans la section 2.1.4 et quelles sont les difficultés rencontrées par la machine dans les sections 2.1.5 et 2.1.6.

Nous verrons ensuite comment l'analyse sémantique d'images est abordée dans l'état de l'art. Dans la section 2.2, nous présenterons les travaux n'utilisant pas un vocabulaire structuré en entrée, en insistant plus particulièrement sur ceux qui introduisent une structure. Dans la section 2.3, nous verrons comment on peut utiliser un vocabulaire structuré en entrée. Nous comparerons les performances des systèmes présentés dans cet état de l'art en section 2.4, et nous en ferons un certain nombre de commentaires (section 2.5), qui nous permettront de justifier plus précisément l'objectif de la thèse.

Two-time Formula One champion Mika Hakkinen drives a McLaren Mercedes F1 car down a section of the proposed F1 street circuit in Singapore March 30, 2008. Hakkinen says the first night race on a Singapore street circuit will pose unique challenges to drivers but safety concerns can be allayed by organisation and preparation. Hakkinen drove on the street as part of an anti-drink driving campaign.

FIGURE 2.1 – *Exemple d'annotation pour une image de presse. Crédit photo : Vivek Prakash/Reuters (avec autorisation).*

2.1 LE PROBLÈME DE L'ANNOTATION D'IMAGE : ASPECTS THÉORIQUES

2.1.1 Comment annoter une image ?

Prenons comme exemple l'image de la figure 2.1. Cette photo de l'agence de presse Reuters est accompagnée d'une description très précise, qui contient les informations utiles pour un journaliste.

L'image est replacée dans son contexte d'origine. Sans même avoir l'image sous les yeux, on peut assez bien imaginer sa composition : si l'on a déjà vu quelques photos de Singapour, et des photos de courses automobiles, on imagine les immeubles en arrière-plan, la F1 passant devant, les barrières délimitant le circuit, la foule massée derrière, etc.

Cette description contient aussi des informations qui ne sont pas visuelles. Par exemple,

- que Mika Hakkinen soit double champion du monde de formule 1,
- qu'il participe à une campagne contre l'alcoolisme au volant,
- que la photo ait été prise le 30 mars 2008,
- que le circuit soit difficile,

sont des faits qui ne peuvent pas être déduits de l'image seule.

Différents types d'annotations possibles

Les informations associées à une image sont généralement appelées des *métadonnées* [52]. Selon les contextes, ces métadonnées peuvent être plus ou moins structurées. Shatford [166] décrit 4 types de métadonnées pouvant être associées à une image :

1. des données biographiques : informations sur la création de l'image (lieu, date, ...),
2. le sujet : une description du contenu de l'image, qui peut prendre plusieurs aspects, que nous verrons ci-après,
3. le type d'illustration : photo, clipart, esquisse, peinture, affiche etc.,
4. des liens vers d'autres images : par exemple, un tableau peut être relié à une esquisse, un plan à une photo du même bâtiment, etc.

Enser [52] inclut dans les métadonnées un possible identifiant unique, et des *annotations de contenu*, qui comprennent un titre, des mots-clés ou des phrases, une légende, ...

Les métadonnées peuvent donc déjà se regrouper en plusieurs catégories. Pour décrire le contenu, on pourra encore avoir des mots-clés, ou des phrases, ou des données structurées. Shatford montre qu'il y a plusieurs niveaux de description possibles :

1. l'image peut être décrite de manière objective, concrète (*Of-ness*), mais aussi de manière subjective et abstraite (*About-ness*). Par exemple, l'image de quelqu'un qui pleure (objectif) pourra aussi représenter la tristesse (subjectif).
2. l'image peut être décrite à différents niveaux entre généricité et spécificité. Par exemple, le *pont de Brooklyn* (description très spécifique) pourra être simplement un *pont* (terme très générique), ou un *pont suspendu* (générique, mais plus spécifique que *pont*).
3. l'image peut être décrite par différentes facettes. Shatford en relève 4 : temporelle (époque), spatiale (localisation géographique), selon l'activité ou l'événement, et selon les objets (animés ou inanimés).

Jörgensen [104] précise cette dernière idée, en montrant que l'image peut être étudiée sous différents aspects, et donc que l'on doit pouvoir y accéder par plusieurs point d'accès. Formellement, cela signifie que les liens existant entre les concepts ne sont pas purement hiérarchiques : ils sont liés par une multiplicité de hiérarchies. Elle distingue par ailleurs des attributs *perceptuels*, objectifs (objets, personnes, couleurs...), *interprétatifs* (*interpretive*), c'est-à-dire sujets à interprétation et donc plus subjectifs (activité, environnement, atmosphère..), et des attributs plus personnels décrivant la réaction du sujet à la vue de l'image.

Enser et Sandom [53] adaptent le modèle de Jörgensen [104] en séparant contenu *perceptuel*, *interprétatif générique*, *interprétatif spécifique* et *abstrait*.

Jaimes et Chang [96] proposent de structurer le contenu de l'image selon dix niveaux présentés dans le tableau 2.1. Les quatre premiers niveaux sont liés à l'aspect syntaxique, purement perceptuel, et les six derniers sont liés à l'aspect sémantique, correspondant à des concepts visuels. Les auteurs font remarquer que ces divisions ne sont pas toujours strictes, mais qu'elles aident à la compréhension du problème. Plus le niveau est élevé, plus il y a besoin de connaissances pour procéder à l'interprétation. Les quatre premiers niveaux sont des éléments totalement objectifs et correspondent à différents niveaux de description structurelle de l'image (mesures quantitatives globales et locales, aspects géométriques...). Les six niveaux conceptuels correspondent aux mêmes niveaux que Shatford, avec les descriptions concrète générique, concrète spécifique et abstraite. Cependant ils font encore une distinction entre la description des objets (par exemple, une F1, des immeubles, une route...) et la description de la scène (une course automobile). Un exemple est donné figure 2.2.

Hollink et al. [93] reprennent ces différents niveaux et les organisent plus formellement à l'aide du langage UML [1]. Une description peut se faire à trois niveaux, générique, spécifique et abstrait, et se rapporter à une scène ou un objet. Une scène peut contenir des objets. Un objet lui-même peut être une composition d'objets. Enfin, la description est un ensemble de caractéristiques : localisation géographique ou spatiale, temporelle, type d'évènement, et les objets déjà mentionnés.

1. "UML (Unified Modeling Language) est un langage graphique de modélisation des données et des traitements". source Wikipédia, consulté le 18 juin 2009.

1.	Type, technique
2.	Distribution globale
3.	Structure locale
4.	Composition globale
5.	Objets génériques
6.	Scène générique
7.	Objets spécifiques
8.	Scène spécifique
9.	Objets abstraits
10.	Scène abstraite

TABLE 2.1 – *Les différents niveaux d'indexation proposés par Jaimes et Chang [96].*

1.	photo
2.	(histogramme)
4.	(segmentation)
5.	fleur, feuille, eau
6.	nature
7.	nénuphar et ses feuilles
8.	pièce d'eau
10.	tranquillité, froideur

FIGURE 2.2 – *Exemple d'annotation d'une image à différents niveaux.*

Hare et al. [87] proposent une gradation de la description entre l'image brute et l'interprétation sémantique haut-niveau : image brute > descripteurs (distribution globale, structures locales) > Objets (composition globale) > Objets nommés > Sémantique (niveau scénique).

Eakins et al. [51] donne différents niveaux d'abstraction pouvant être mis en relations avec les niveaux 6,8,9,10 de Jaimes et Chang [96] (voir Tableau 2.2). L'abstraction *contextuelle* dépend de la connaissance de l'environnement ; l'abstraction *culturelle* dépend d'une connaissance culturelle spécifique (par exemple, pour comprendre la signification d'une cérémonie religieuse particulière) ; l'abstraction *émotionnelle* dépendra fortement de la personne qui regarde l'image ; l'abstraction *technique* nécessite une expertise dans le domaine concerné, par exemple pour détecter les traces d'un cancer sur une radiographie.

Le tableau 2.2 résume les différents niveaux de description sémantique proposés par les auteurs. Il permet de constater que les différentes divisions en niveau d'abstraction se correspondent globalement, en particulier tant que l'on reste au niveau objectif. Dès lors que l'interprétation devient subjective, il est naturellement plus difficile de faire des distinctions entre les niveaux d'abstraction.

Différents niveaux de précision des annotations

Par exemple, dans l'image de la figure 2.1, un observateur nommera les objets avec une précision variable, selon son niveau de connaissance :

- une scène d'extérieur,
- une voiture, une rue, des immeubles, des personnes, du ciel,
- une ville,
- une voiture de sport,
- une course automobile, des spectateurs,
- une ville moderne, un quartier d'affaires,
- une Formule 1,
- une McLaren Mercedes F1,
- la ville de Singapour,

Article	**Niveau d'interprétation :** Générique ⟶ spécifique ⟶ abstrait					
[166]	Objectif, *Of-ness*				Subjectif, *About-ness*	
	Generic-of		*Specific-of*		*Generic-about*	*Specific-about*
[104]	Perceptuel		Interprétatif			
	Generic		*Specific*		*Abstract*	
[96],[93]	Objet	Scène, Composition	Objet	Scène, Composition	Objet	Scène, Composition
[53]	Perceptuel	Interprétatif générique	Interprétatif spécifique		Abstrait	
[51]	unités sémantiques	abstraction contextuelle	unités sémantiques	abstraction technique	abstraction émotionnelle	abstraction culturelle
[87]	Objets nommés	Sémantique				
Exemple	voiture, ciel, immeuble	extérieur, ville	voiture de sport, F1	course automobile, Singapour	pollution, spectacle	événement sportif de rang mondial

TABLE 2.2 – *Récapitulatif et mise en relation des niveaux de descriptions proposés par les différents auteurs. En ligne : niveaux de description de chaque article. En colonne : niveau de généricité. La colonne la plus à gauche correspond au niveau fondamental de généricité pour des objets (partagé par un grand nombre d'utilisateurs). La colonne la plus à droite correspond à un niveau d'abstraction spécifique, autrement dit un niveau d'interprétation plus personnel. Rappelons que ces divisions sont à prendre avec précaution : les frontières sont en réalité très floues, voire inexistantes, entre les différents niveaux d'interprétation.*

– la McLaren Mercedes F1 de Mika Hakkinen.

Dans ces énoncés, on peut relever plusieurs hiérarchies. Notons $\prec$ la relation Is-A : $A \prec B$ si A est un B, c'est-à-dire si tout objet de la classe A appartient aussi à la classe B. Notons $\sqsubset$ la relation PART-OF : $A \sqsubset B$ si A est une partie de B, et $\wedge$ la relation de co-occurrence (objets trouvés dans une même image). Dans notre exemple, nous avons donc (entre autres) :

– McLaren Mercedes F1 $\prec$ F1 $\prec$ voiture de sport $\prec$ voiture,
– ville moderne avec des gratte-ciel $\prec$ ville,
– voiture de sport $\wedge$ spectateur $\sqsubset$ course automobile,
– immeubles $\wedge$ rue $\sqsubset$ ville.

Nous nous intéressons ici à l'étude du contenu de l'image, c'est-à-dire de son sujet. Nous ne traiterons donc pas les annotations non-visuelles (biographie, liens), qui relèveraient plutôt de l'étude de documents multimédia (texte, vidéo, pages web). De même, le type d'illustration (photo, dessin, plan...) ne sera pas l'objet de notre intérêt. C'est un aspect qui peut être traité de manière indépendante du sujet de l'image.

La description du sujet d'une image est un objet d'étude en soi, bien connu des archivistes. Nous avons vu que l'image peut être décrite à différents niveaux de généricité, et selon plusieurs modes de description. Selon l'objectif de l'utilisateur, il ne s'intéressera pas au même niveau de description. A quel niveau de description doit-on s'intéresser de manière privilégiée ?

2.1.2 Les besoins de l'utilisateur

Des exemples

Barnard et al. [14] relèvent quelques unes des manières d'exploiter les bases d'images, en citant notamment les études de Markkula et Sormunen [125], concernant les besoins des journalistes, et de Frost et al. [73], concernant la recherche d'œuvres d'art. Markkula et Sormunen [125] présentent les besoins des journalistes pour la recherche d'images dans les archives des journaux. Les journalistes cherchent des images à partir de quelques mots-clés. Ces images d'archives ont été annotées au préalable par les archivistes avec les mots-clés qui leur paraissaient les plus pertinents. L'étude montre qu'il est difficile de trouver des images représentant des concepts génériques par ce moyen.

Plusieurs aspects à considérer

Hollink et al. [93] proposent une classification des utilisateurs selon trois axes : le domaine de recherche, le type de tâche effectuée, et le niveau d'expertise de l'utilisateur. Le domaine correspond à la collection d'images qui est traitée . Il est qualifié par son étendue (la variété de son contenu), par la taille du vocabulaire (le nombre de termes employés), et par les niveaux d'expertise applicables (il y a des domaines où il y a des disparités plus marquées entre spécialistes et profanes,

Le domaine des images

La finalité

comme dans le cadre de l'imagerie médicale). La tâche est définie par le but de la recherche, c'est-à-dire le besoin, par la précision de la requête (si elle est vague, on optera pour la navigation ; elle peut aussi être catégorielle ou spécifique), et par la méthode de recherche (navigation, mots-clés, texte libre, utilisation ou non d'opérateurs logiques, exemples, croquis).

Eakins [50] répertorie trois modes de recherche d'images :

1. par les caractéristiques *primitives* de l'image (par exemple, "trouver des images contenant des étoiles jaunes en cercle"),
2. par des attributs dits *logiques*, nécessitant un premier degré d'interprétation de l'image (par exemple, "trouver un train passant sur un pont"),
3. par des attributs *abstraits*, nécessitant un raisonnement complexe (ex : "trouver des images illustrant la notion de liberté").

La plupart des systèmes existants font une recherche basée sur le contenu au niveau 1. Cependant, la grande majorité des requêtes exprimées par les utilisateurs (d'après Armitage et Enser [8] cité par [50]) se situent au niveau 2, et beaucoup se situent au niveau 3.

Eakins et al. [51] identifie sept classes d'utilisation des images :

Illustration lorsque les images sont utilisées avec un autre média, typiquement un texte,

Traitement de l'information lorsque les données contenues dans l'image sont l'objet principal de l'étude (par exemple, des radiographies),

Diffusion d'information lorsqu'il s'agit de transmettre l'information contenue dans l'image à quelqu'un d'autre (par exemple, une photo d'identité judiciaire aux services de police),

Apprentissage lorsque les images permettent d'accroître les connaissances des gens (par exemple, des œuvres d'art),

Création d'idée lorsque les images sont une source d'inspiration ou de réflexion,

Valeur esthétique lorsque les images servent de décoration,

Valeur émotionnelle lorsque c'est l'impact des images sur les personnes qui compte (par exemple, pour la publicité).

L'arrière-plan de l'utilisateur

Enser et Sandom [53] distinguent seulement deux niveaux d'utilisateur, généraliste ou spécialiste. Hollink et al. [93] montrent qu'il y a différents degrés de spécialisation des utilisateurs, et que cela dépend des domaines. Tous n'ont pas non plus la même attitude : Jaimes [97] relève les différents types de comportements que peut avoir le sujet à la recherche d'information (exploration, intuition, curiosité, focalisé...). Santini et Dumitrescu [161] montrent que le problème vient aussi de ce que les images ont un contenu subjectif, et donc que pour répondre correctement aux besoins de l'utilisateur, il faut connaître son contexte de travail.

Varier les points de vue descriptifs

Dans l'étude de Yee et al. [197], des étudiants en histoire de l'art explorent une base d'image d'art avec deux types d'interfaces : une interface permettant de naviguer par rapport aux concepts selon différents directions, et une interface classique. L'interface classique permet une recherche par mots-clés. La deuxième interface s'appuie sur des métadonnées organisées selon plusieurs *facettes* : autrement dit, les images peuvent être regroupées selon différents jeux de catégories indépendants (orthogonaux). Des facettes possibles sont le thème (militaire, religieux...), le type de média (tapisserie, céramique...), le domaine artistique, la localisation géographique... Pour une facette, les labels peuvent être hiérarchiques ou non, multilabel ou non. Les utilisateurs montrent une préférence pour cette dernière interface, qui permet une navigation plus intuitive.

Nécessité d'une interprétation

Toutes les études menées s'accordent sur le fait que les utilisateurs sont plus intéressés par les ressemblances conceptuelles entre les images que par les ressemblances visuelles. L'annotation est donc cruciale pour l'organisation, l'indexation et la recherche dans les base d'images. Il ressort également qu'il est nécessaire de pouvoir extraire une information à différents niveaux, et selon différentes facettes, pour pouvoir répondre aux besoins d'un maximum d'utilisateurs.

2.1.3 Les effets de l'interaction

Nous avons vu que l'étude de la sémantique d'une image ne peut pas être séparée de l'étude des besoins de l'utilisateur. Déjà en 1998, Santini et Jain [159] ont montré les limites d'un simple système de recherche d'image par le contenu, et ont insisté sur l'importance de l'interface avec l'utilisateur. Scherp et Jain [162] dé-

La sémantique peut varier selon l'utilisateur...

loppent une théorie dans laquelle plusieurs types de sémantique sont définies par rapport à l'interaction de l'utilisateur avec le monde environnant. Ils présentent cinq types de sémantiques inter-dépendants :

- La *sémantique naturelle*, qui permet la reconnaissance des objets familiers, des personnes, des événements de la vie courante (par ex., une voiture, un panneau de circulation) ;
- La *sémantique analytique*, qui nécessite une analyse plus poussée. Sans avoir de frontière nette avec la sémantique naturelle, elle nécessite cependant des opérations par rapport aux éléments de celle-ci, au moins dans la phase d'apprentissage, (par ex., reconnaître un accident de voiture nécessite une analyse de la scène),
- La *sémantique de l'utilisateur*, qui permet une reconnaissance spécifique des choses, dépendant entièrement de l'arrière-plan culturel, de l'expérience, de l'environnement de l'utilisateur, et qui influence toutes les autres, (par ex., un panneau de circulation spécifique à l'Australie, le chiffre 4 qui a des connotations particulières en Extrême-Orient),
- La *sémantique expressive*, qui définit comment l'utilisateur va exprimer son modèle mental par la création d'objets dans le monde physique. Elle peut être codifiée, comme en peinture ou en cinéma, mais elle est essentiellement personnelle, (par ex., une banderole protestataire, une photo avec un angle de vue particulier),
- La *sémantique émergente*, issue d'une boucle d'analyse et de synthèse entre le monde physique et le modèle mental, qui varie au cours des expériences de la personne. C'est le fruit de l'interaction entre l'utilisateur et son environnement. Un exemple de sémantique émergente peut être une émotion provoquée par une photo : forte peu de temps après l'événement, atténuée longtemps après.

Ainsi, les auteurs présentent une théorie expliquant comment évolue le sens des choses pour l'utilisateur au cours de ses interactions. Pour eux, tout est subjectif, et la compréhension du monde dépend entièrement de la culture et du contexte de l'utilisateur. C'est également la théorie soutenue par Santini [160], qui soutient que l'on ne peut aucunement décrire une image au moyen d'ontologies, puisque le sens d'une image est toujours subjectif, et toujours lié à un contexte (culture, activité...).

... mais une partie de la sémantique reste objective

Il est incontestable que la culture et le contexte ont une influence sur la compréhension du monde, mais, comme nous l'avons vu avec Shatford Layne [166], Jaimes et Chang [96], Hollink et al. [93], une partie de l'interprétation de l'image est parfaitement objective, et peut être donnée à partir d'un tronc commun de connaissances, supposées partagées par la majorité des êtres humains. Ainsi, quels que soient le lieu, l'heure, l'époque, l'environnement ou la langue qu'on emploie, un chien est un chien, une image de chien représente un chien, et lui attribuer le label *chien* aura un sens. Ce n'est pas forcément ce sens-là qui sera utile à un utilisateur donné dans un contexte donné, mais ce sera toujours juste. Dans un autre contexte, c'est peut-être simplement la représentation d'un animal qui sera intéressante pour l'utilisateur.

Dans le contexte du traitement automatique des images, on retiendra simplement que certains aspects d'une image peuvent être décrits objectivement, et que d'autres dépendent fortement de l'utilisateur et de son contexte.

2.1.4 Annotation manuelle ?

Comme nous l'avons vu, le contenu de l'annotation est un problème en soi : il faut définir le vocabulaire, décider à quel(s) niveau(x) annoter l'image, selon quelle(s) facette(s), et comment les organiser. C'est déjà une difficulté pour les spécialistes de l'indexation de contenu. En admettant que ces éléments soient délimités, si elle était faite par des spécialistes, l'annotation humaine serait la référence. Étant donné la quantité croissante de données numériques, faire appel à des spécialistes pour l'annotation à grande échelle est impossible.

L'annotation manuelle est difficile...

Même pour annoter l'image à un niveau non-spécialisé, cela pose plusieurs problèmes :

- C'est un travail laborieux, quand il y a beaucoup d'images. On ne peut pas demander à tous ceux qui ajoutent des images dans une base d'annoter précisément chaque image, quand il y a une centaine d'images à ajouter, qu'il faut 5 mn pour les charger, et qu'il faut 3h pour les annoter manuellement.
- C'est sujet à ambiguïtés. L'utilisateur lambda n'annotera pas forcément l'image au niveau idéal, mais selon son contexte : si ce sont des images personnelles, il aura tendance à donner le lieu, les circonstances etc. au lieu de parler du contenu.

...et incertaine

- Pour résoudre les ambiguïtés, il faudrait plusieurs annotateurs, ce qui ajoute au travail.
- C'est un travail sans fin : de nos jours, une base d'image est rarement fixe. Elle s'agrandit très rapidement.

Dans la majorité des cas, les annotations faites par des humains sont peu fiables et/ou incomplètes et/ou inutiles. Santini et Dumitrescu [161] relèvent de manière sardonique plusieurs défauts humains qui affectent les annotations dans le cas plus général des documents :

1. *Les gens mentent* : En sus de l'exemple typique des spams, on peut facilement comprendre que des commerciaux exagèrent la description de leurs produits.
2. *Les gens sont paresseux* : La plupart du temps, ils ne voudront pas mettre d'annotations, ou alors elle seront minimalistes ou très générales.
3. *Les gens sont stupides* : Même en toute bonne foi, les gens se trompent dans ce qu'ils disent, parfois même sur des choses évidentes.

Hanbury [86] recense plusieurs méthodes pour mutualiser l'effort d'annotation des images. Chacune comporte des avantages et des inconvénients : il est nécessaire de trouver un compromis entre fiabilité, richesse des annotations, et main d'œuvre. Plusieurs initiatives utilisent Internet pour annoter de manière collaborative : LabelMe [2], ESP-Game [3] sont des exemples. Bien qu'efficaces dans le sens où l'on peut annoter beaucoup d'images dans ce sens, ces méthodes gardent les défauts de l'annotation humaine. Elles sont par ailleurs restreintes aux applications pour lesquelles l'annotation n'est pas nécessaire rapidement, et pour lesquelles les images peuvent être diffusées.

Il est donc bien nécessaire d'automatiser la tâche d'annotation. D'une manière générale, l'annotation automatique d'image se fait en deux étapes :

1. extraction d'indices visuels bas-niveau,
2. interprétation de ces indices et liens avec les concepts haut-niveau.

2. site LabelMe : http ://labelme.csail.mit.edu/, Russell et al. [158]
3. Site du jeu ESP : http ://www.gwap.com/gwap/gamesPreview/espgame/

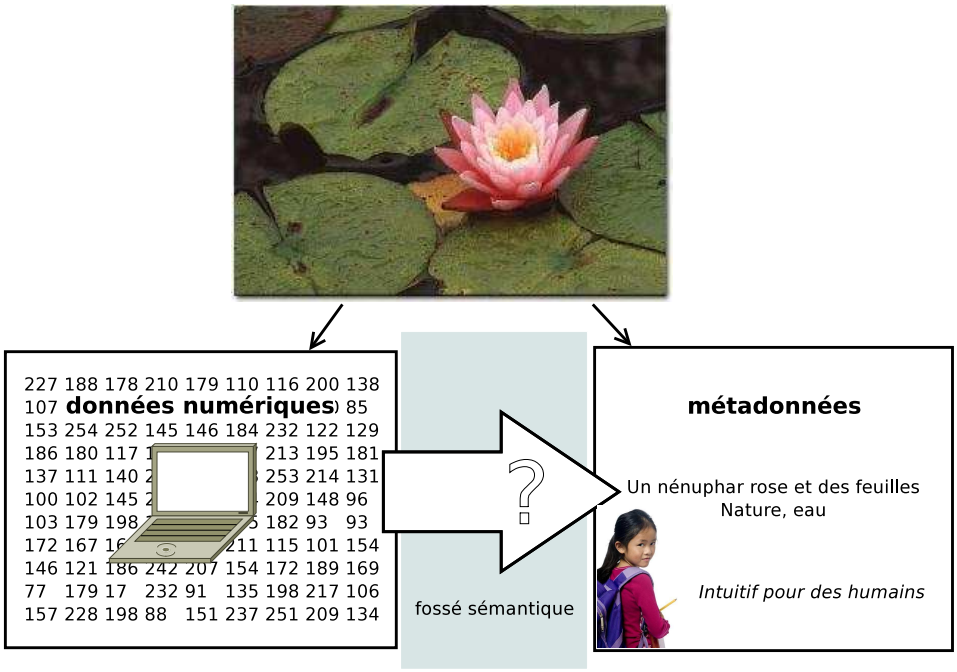


FIGURE 2.3 – *Illustration du problème du fossé sémantique.*

La deuxième étape peut consister simplement à organiser les images selon les indices bas-niveau, sans mettre en jeu une annotation explicite, en utilisant des méthodes d'apprentissage non-supervisé. Mais pour être vraiment exploitable, on utilisera en général des méthodes d'apprentissage supervisé : une partie des images est annotée et sert à apprendre des correspondances entre les indices visuels et les concepts. Il y aura donc toujours un effort à fournir pour annoter les images servant à l'apprentissage. Cette annotation sera forcément imparfaite, et il faudra tâcher de faire en sorte d'éliminer les ambiguïtés et les incohérences lors de l'annotation automatique.

2.1.5 Le fossé sémantique

Tous les travaux au sujet de l'annotation d'image, et plus généralement de la reconnaissance d'objets, font état de ce qu'ils appellent le *fossé sémantique*. D'une manière générale, celui-ci est présenté de manière schématique, de manière analogue à la figure 2.3 : le lien entre les données numériques et les concepts contenus dans l'image n'est pas naturel pour une machine comme pour un être humain.

Un problème pour la machine et non pour l'homme

La reconnaissance visuelle, pour l'être humain, consiste à faire le lien entre l'image telle qu'elle est imprimée sur la rétine, et une représentation présente en mémoire [182]. On a pu étudier assez précisément ce qui se passait au niveau de la rétine et dans les premiers stades de la reconnaissance. Cependant, la façon dont les objets et concepts connus et exprimés par le langage sont représentés en mémoire sont plus difficiles à comprendre, et donc à imiter. Notre cerveau reconnaît un grand nombre d'objets en un clin d'œil, et lui associe des termes tout aussi rapidement. A l'heure actuelle, on n'a pas pu reproduire ce résultat avec autant d'efficacité. Cet espace non comblé entre les aires visuelles bas-niveau, où l'on a des données comparables à des données numériques, et les aires où sont stockés des modèles pour chaque chose, correspond au fossé sémantique.

Sémantique et image

La notion de sémantique est utilisée de manière très libre par les chercheurs pour indiquer tout ce qui peut s'exprimer naturellement en langage courant, par opposition aux données numériques, qui n'ont aucun sens en elles-mêmes. Peu se

préoccupent de ce que signifie réellement un *concept*, ou le *sens* de l'image, l'essentiel est de pouvoir attacher des termes naturels à l'image. Nous ne chercherons pas non plus à rentrer dans un débat pour savoir à partir de quel moment on peut parler de sémantique. Nous emploierons ce terme dès lors que l'on utilise des termes du langage naturel pour décrire l'image : ainsi des tâches comme la catégorisation d'image ou même la détection d'objets rentreront dans ce que nous appellerons *analyse sémantique*. Pour une étude plus poussée de la notion même de sémantique, on pourra voir la thèse de Bordes [25].

Différentes communautés étudient le problème de l'analyse sémantique d'image, qui correspondent à différentes approches et différentes influences. Alors que dans la communauté *Intelligence Artificielle* on construira des systèmes d'inférence entièrement dépendants d'ontologies ultra-développées, dans la communauté *Vision* classique, les méthodes statistiques auront le dessus. Le vocabulaire employé sera alors souvent très restreint, et généralement non structuré. Enfin, la communauté *Indexation* s'est beaucoup inspirée des méthodes de la linguistique, et combinera souvent des méthodes statistiques (ou parfois, logiques) avec des structures de vocabulaires telles que thésauri, taxonomies ou ontologies.

Nous avons séparé les approches en deux grandes familles : celles qui exploitent et celles qui n'exploitent pas une structure du vocabulaire imposée.

Dans la section suivante, nous expliqueront succinctement les liens entre reconnaissance d'objets et vocabulaires contrôlés (section 2.1.6). Nous étudierons ensuite, dans un premier temps, les travaux de la première famille, cherchant à résoudre "de front" le problème du fossé sémantique (section 2.2). Dans un deuxième temps, nous verrons les méthodes de la deuxième famille, cherchant d'abord à réduire ce fossé sémantique, en introduisant des étapes intermédiaires à la reconnaissance (section 2.3). Nous nous intéresserons plus particulièrement aux méthodes introduisant des liens de hiérarchie entre les termes, et celles exploitant les ontologies.

2.1.6 Reconnaissance d'objets et vocabulaires

Décrire un monde complexe...

Outre la difficulté d'interprétation due aux différents niveaux de sémantique possible, la reconnaissance d'objets à un niveau donné est un problème très complexe, dont la difficulté est due à de multiples facteurs :

- il existe de nombreux objets différents. L'humain est capable de reconnaître aisément environ 30 000 catégories d'objets Biederman [21].
- visuellement, les variations inter-catégories peuvent être assez faibles, et les variations intra-catégories relativement importantes.
- les conditions de prise de vue ne sont pas prévisibles, en général. Ceci inclut les variations de luminosité, de contexte, d'angle de vue, ainsi que les occultations.
- les objets eux-mêmes peuvent parfois avoir des formes variables : un visage humain peut prendre diverses expressions, un animal diverses positions.

En fait, la complexité du problème vient directement de la complexité du monde environnant. Depuis les débuts de la recherche en reconnaissance d'objets, on a cherché à modéliser les objets pour des cas simples. Ces efforts ont été couronnés de succès, mais dans le cadre restreint d'applications particulières.

Les études faites en psychologie cognitive ont montré que dans le cerveau humain, la reconnaissance se fait d'abord à un niveau de base ou niveau *fondamental* (en anglais *basic level*, pour plus de précisions, se référer à Rosch et al. [155], Murphy et Smith [135], Jolicoeur et al. [102]). Partant de cette catégorie, et

en regardant les variations des caractéristiques spécifiques de cette classe, on peut trouver la sous-catégorie correspondant à l'image.

Il est donc logique de chercher à catégoriser les objets à ce niveau de base, avant de chercher à faire d'autres distinctions. Dans un problème de catégorisation, une image est associée à un terme unique décrivant l'objet contenu dans l'image. Le vocabulaire $\mathcal{V}$ sera donc de taille finie K, $\mathcal{V} = \{w_1, \ldots, w_K\}$, où K est le nombre de catégories à distinguer.

Un *vocabulaire contrôlé* est un vocabulaire de référence, de taille fixe, utilisé pour l'indexation. Il peut être structuré sous la forme de thésaurus, de taxonomie ou d'ontologie. Gilchrist [77] rappelle les définitions "classiques" de ces trois types de structure, et explique comment leur usage a évolué parmi les scientifiques des différentes communautés. Garshol [75] donne des définitions plus précises de ces structures.

Le mot *taxonomie* se réfère à l'origine à la structure arborescente utilisée pour la classification du vivant. Plus généralement, nous utiliserons ce terme pour représenter toute structure hiérarchique de type Is-A entre les termes.

Il y a contradiction entre les différents auteurs quant à la différence entre taxonomie et thésaurus. Ces deux structures étant issues de deux domaines distincts (classification du vivant / indexation de documents), leur usage a évolué de manière assez indépendante. Nous dirons qu'un *thésaurus* étend la notion de taxonomie en lui ajoutant d'autre types de relations entre les termes :

- les *synonymes*, ou équivalents, en précisant quel est le terme préconisé,
- les termes *associés*, ou "liés", appartenant au même domaine.

Par ailleurs, chaque terme est accompagné d'une note d'usage précisant quel est le sens du terme tel qu'il est employé dans le thésaurus.

Une *ontologie* est un modèle de description du monde environnant (c'est-à-dire des concepts) défini explicitement par différents types de concepts, de propriétés et de relations entre ces concepts. L'ontologie décrit les concepts avec un modèle formel, directement implémentable en machine. Les relations incluses comprennent celles décrites plus haut, mais l'ontologie peut intégrer, *a priori*, n'importe quel type de relation.

Les *réseaux sémantiques* sont une structure intermédiaire entre thésaurus et ontologie, en ce qu'ils décrivent un plus grand nombre de relations que les thésauri, et sont moins développés et formalisés que les ontologies. La figure 2.4 présente quelques relations entre des catégories d'objets classiques. On distingue les nœuds représentant des concepts, qui correspondent à des catégories d'objets (par exemple, "voiture"), et les nœuds représentant des instances d'objets, c'est-à-dire des occurrences d'un concept (par exemple, "la voiture de M.Dupont").

...avec un vocabulaire adapté ?

Les différents problèmes résolus par la reconnaissance d'objets utilisent en général de simples vocabulaires que l'on apparentera à des vocabulaires contrôlés. La figure 2.5 présente les différentes manières d'ajouter un contenu "sémantique" à l'image.

On distingue en particulier :

- la *détection* d'objets, qui consiste à décider de la présence d'un objet d'une catégorie donnée dans une image, donnant une décision binaire présent/absent,
- la *localisation*, souvent associée à la détection, mais pas nécessairement, donne la position et l'échelle d'un objet donné dans une image,
- la *catégorisation*, qui consiste à attribuer un unique label choisi parmi une liste donnée (voir figure 2.6),
- l'*identification*, qui consiste, soit en une catégorisation d'objets particuliers

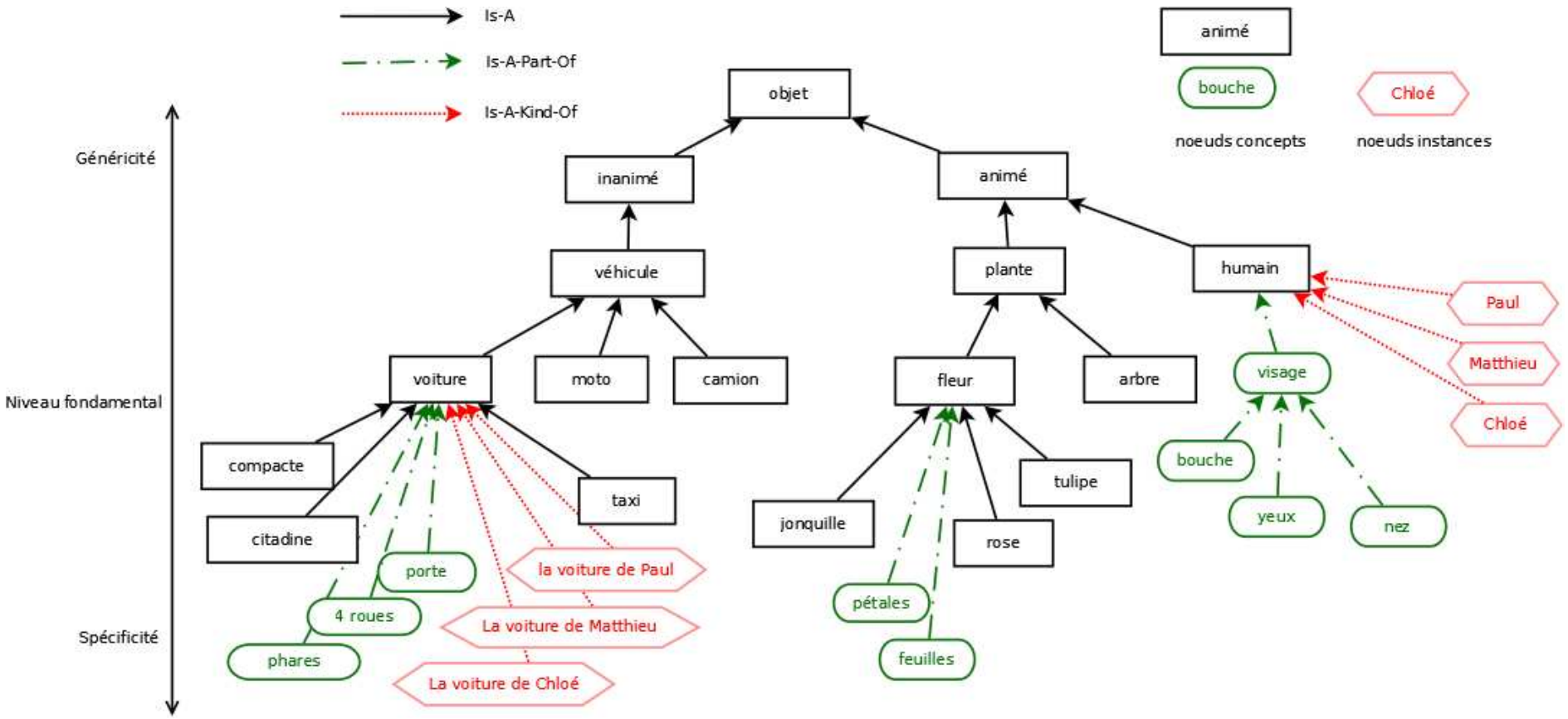


FIGURE 2.4 – *Exemple de réseau sémantique représentant les relations entre des catégories classiquement utilisées en reconnaissance d'objets.*

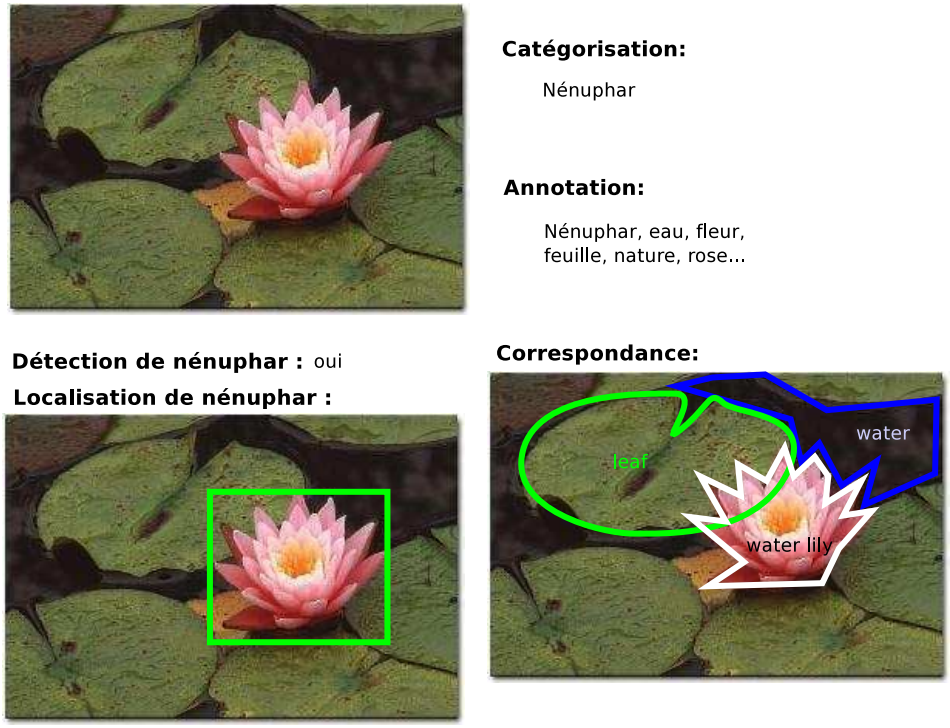


FIGURE 2.5 – *Les différents problèmes d'interprétation sémantique d'image.*

(en sous-catégories), soit en la détection de présence d'instances d'objets d'une catégorie donnée,

- l'*annotation*, qui consiste à associer des termes à l'image, sélectionnés parmi un vocabulaire fixé, et qui correspond à une analyse multilabel,
- la *correspondance*, souvent assimilée à un problème d'annotation, associe un terme unique à chacune des régions de l'image (catégorisation des régions).

Ces problèmes sont voisins : la correspondance est souvent utilisée pour la catégorisation, la catégorisation pour l'annotation. La catégorisation est généralement un préalable à l'identification. Leur résolution s'appuie lourdement sur l'usage de techniques de traitement d'image et d'apprentissage statistique.

La plupart des problèmes étudiés jusqu'ici en reconnaissance d'objets peuvent se situer par rapport une représentation sous forme de réseau sémantique.

- la catégorisation d'objets fait la distinction entre des catégories correspondant à des nœuds fondamentaux (cf. Fig 2.7),
- la détection d'objets se situe généralement sur un nœud du niveau fondamental (cf. Fig 2.8) et peut s'appuyer sur ses méronymes,
- on parle d'identification, lorsque la catégorisation se fait à un niveau plus spécifique que le niveau fondamental, et que toutes les catégories descendent d'un même nœud. Les relations de méronymies sont alors facilement exploitables (cf. Fig 2.9).

La fin de ce chapitre est consacrée à l'étude des différentes méthodes générales utilisées dans l'état de l'art pour l'analyse sémantique d'images. Pour le lecteur non-accoutumé aux principes de la reconnaissance d'objets, nous proposons en annexe un mini-état de l'art présentant quelques méthodes de traitement d'image (annexe A) et d'apprentissage statistique (annexe B) classiquement utilisées dans ce domaine.

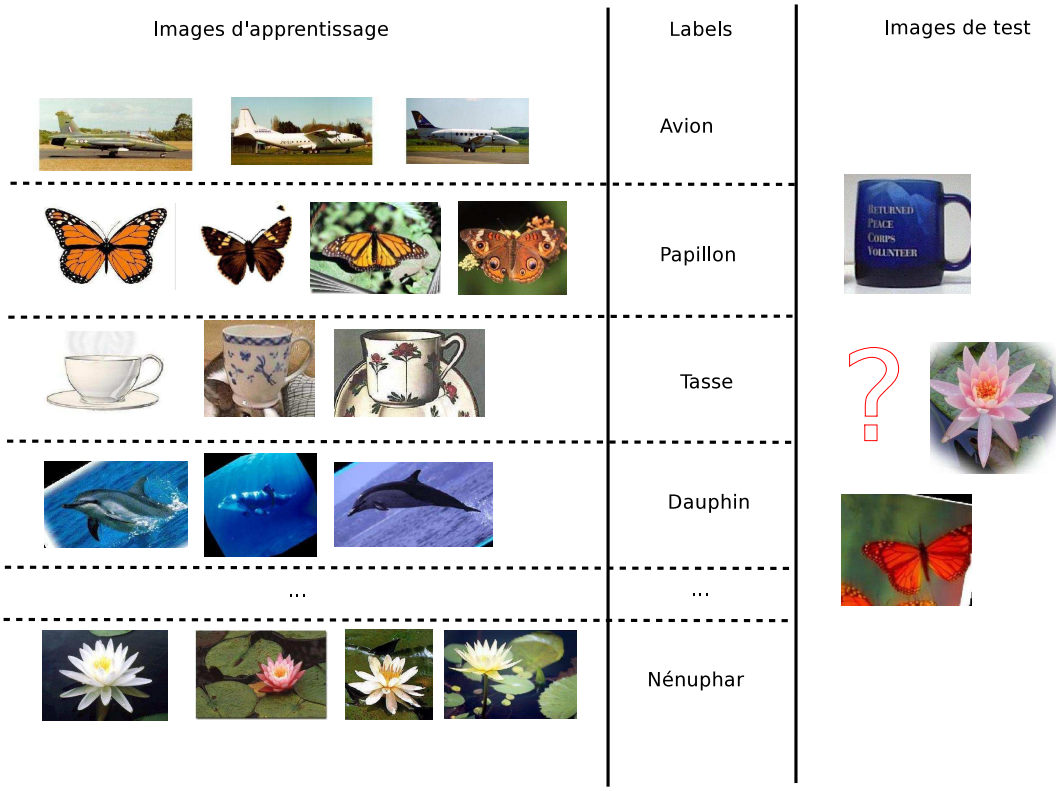


FIGURE 2.6 – *Schéma classique de la catégorisation d'objets : la base d'images d'apprentissage contient un certain nombre d'images pour chaque catégorie. Un label est associé à chaque image, correspondant à cette catégorie. L'apprentissage sert à estimer une fonction de décision, qui permet ensuite de catégoriser de nouvelles images "test".*

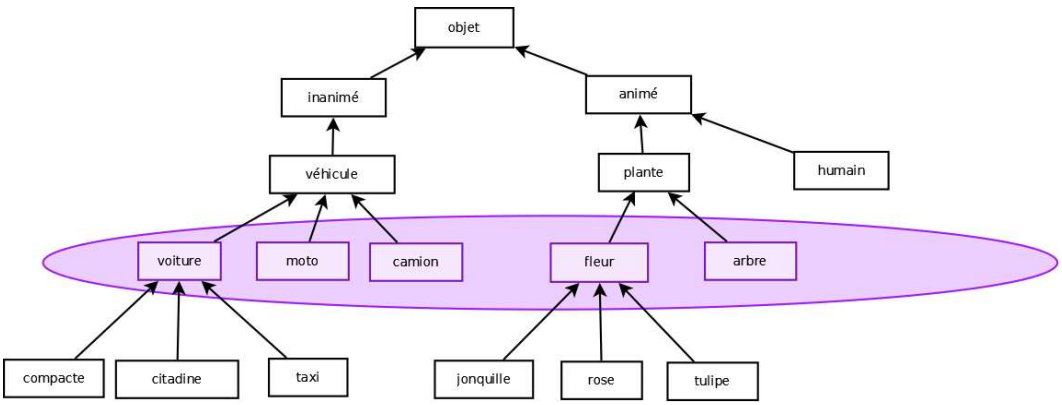


FIGURE 2.7 – *Classification multiclasse sur des objets appartenant à des catégories* fondamentales (basic level).

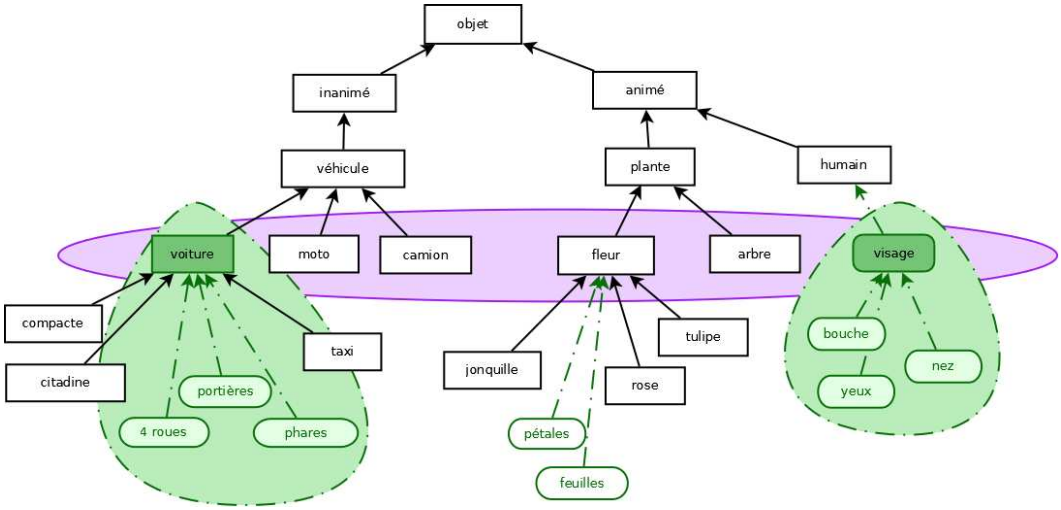


FIGURE 2.8 – *Utilisation des relations de méronymie pour la détection d'objets.*

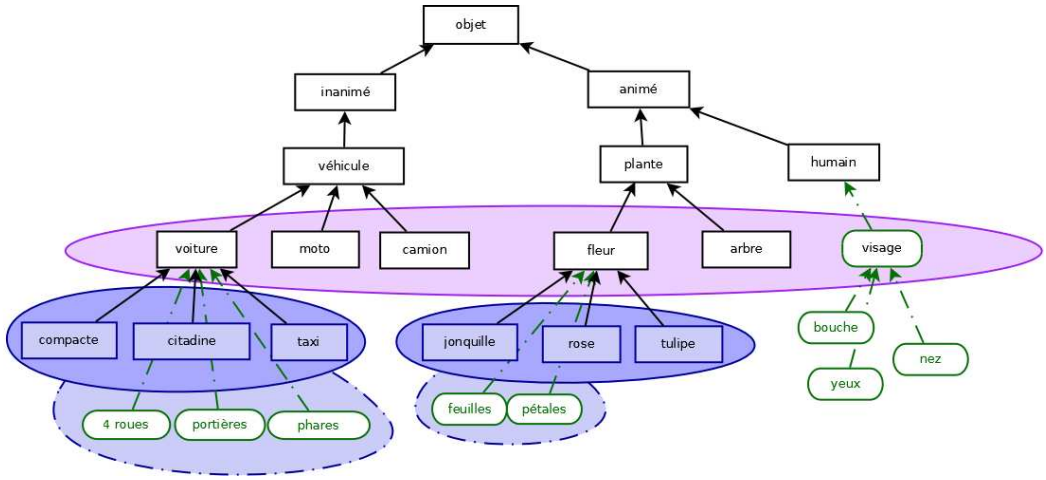


FIGURE 2.9 – *Positionnement du problème d'identification. L'utilisation des relations de méronymie est possible.*

2.2 Annotation d'images avec un vocabulaire non structuré

2.2.1 Introduction

Des recherches importantes ont été effectuées dans le but de résoudre le problème de l'analyse sémantique des images, et donc pour trouver une solution au fossé sémantique. Les données sur lesquelles les méthodes sont testées varient énormément : de quelques classes pour la reconnaissance de scènes ou d'objets génériques, à des centaines voire des milliers, plus récemment, pour la reconnaissance d'objets ou pour la recherche d'images par le contenu.

Les recherches sur la compréhension des images se sont principalement effectuées dans deux familles de la vision par ordinateur : en reconnaissance d'objet, pour qui le problème est abordé de front, et en recherche d'image par le contenu, pour lequel l'aspect sémantique n'a pas toujours été prioritaire. Pour des états de l'art en reconnaissance d'objets, on pourra consulter Mundy [134] qui propose une revue historique, Pinz [145] pour les techniques de catégorisation d'objets, et Bosch et al. [27] pour le problème plus spécifique de la reconnaissance de scènes. Plusieurs articles de la fin de la dernière décade proposent un état de l'art des travaux accomplis en recherche d'image par le contenu (Rui et al. [157], Smeulders et al. [169]), et depuis, le nombre de travaux sur le sujet a augmenté de manière exponentielle. Des états de l'art plus récents sont ceux de Eakins [50], qui prone l'introduction de plus d'intelligence artificielle dans le domaine, Liu et al. [120], qui s'intéresse plus particulièrement aux systèmes avec interprétation sémantique, et Datta et al. [45], plus général, présentant diverses tendances et difficultés. Le problème de la compréhension du contenu de l'image (autrement dit de l'aspect sémantique "haut-niveau" des images) est maintenant toujours relevé.

Nous distinguerons deux grandes catégories parmi les approches proposées : (a) les approches classiques, cherchant des liens entre les données numériques et un ensemble de labels de même niveau ; (b) les approches basées sur l'exploitation des connaissances, s'appuyant sur une organisation des labels préalablement connue.

Pour les approches de la première classe, on peut encore distinguer différents types de méthodes :

- les approches *directes*, qui en général cherchent à résoudre le problème par des méthodes statistiques,
- les approches *linguistiques*, basées sur la construction d'un vocabulaire visuel comme intermédiaire entre les données numériques et sémantiques,
- les approches *compositionnelles*, dans lesquelles des parties de l'image sont identifiées (segmentation) avant de classifier l'image dans son ensemble ou pour annoter ses parties,
- les approches *structurelles*, qui en plus de reconnaître des parties de l'image, s'intéresse à des aspects géométriques,
- les approches *compositionnelles hiérarchiques*, dans lesquelles une hiérarchie de parties est construite,
- les approches *communicantes*, dans lesquelles les catégories peuvent partager des informations,
- les approches *progressives/hiérarchiques*, cherchant à trouver des liens hiérarchiques entre les catégories,
- les approches *multilabels*, permettant d'assigner plusieurs labels simultanément à une même image, avec ou sans segmentation préalable.

Les approches de la deuxième classe font l'objet de la section 2.3. Les performances des différentes méthodes seront précisées dans la section 2.4.

2.2.2 Les approches directes

Ces méthodes cherchent directement des liens entre les caractéristiques images bas-niveau et les caractéristiques sémantiques haut-niveau. Elles cherchent un pont direct au-dessus du fossé sémantique, et en général permettent de définir des méthodes qui servent de base aux méthodes plus évoluées que nous verrons par la suite. Plusieurs travaux de cette catégorie ont influencé profondément le domaine de l'interprétation d'image, en particulier celui de Lowe [121], qui propose une méthode de reconnaissance basée sur des caractéristiques images invariantes aux symétries et aux rotations/changement d'échelle, les SIFT. Ces SIFT sont particulièrement efficaces pour décrire localement les images.

Barla et al. [13] et Boughorbel et al. [29] utilisent des noyaux intersection d'histogrammes avec des caractéristiques descriptives globales.

Li et Wang [116] présentent le système d'annotation automatique d'image ALIPR, dont l'objectif est d'avoir un système rapide (en temps réel), c'est-à-dire environ 1 seconde. pour annoter une image. Leur système repose sur des distances à des prototypes, calculés par *clustering* de distributions discrètes. Pour chaque image, la probabilité qu'un mot lui soit associé est calculée, et les mots les plus probables sont sélectionnés. Les performances sont estimées sur la base Corel, avec 332 termes répartis sur 599 catégories. Pour l'apprentissage, ils utilisent 80 images par catégorie. Les mots sont répartis de manière très inégale entre les catégories.

Plus récemment, Makadia et al. [123] proposent une méthode d'annotation destinée à servir de référence dans le domaine, qui utilise des caractéristiques image "simples" (histogrammes de couleur et textures). Les labels sont assignés aux images par transfert à partir des plus proches voisins, en utilisant deux distances possibles.

2.2.3 Les approches linguistiques

Nous classons parmi les approches linguistiques les méthodes de type *sacs-de-mots*, méthode proposée initialement par Dance et al. [42]. Le principe des sacs-de-mots est inspiré du traitement de documents textuels. Il consiste à extraire les caractéristiques de l'image et à les grouper par rapport à des clusters constituant un vocabulaire visuel. Des histogrammes d'occurrences de ces mots visuels sont calculés, permettant une description de l'image très efficace.

Nous comprenons également dans cette catégorie les méthodes ajoutant un certain niveau de structure géométrique ou de hiérarchie sous la forme de grilles successives. Ainsi, la méthode SPM (*Spatial Pyramid Matching*) de Lazebnik et al. [113] consiste à faire des sacs-de-mots en découpant l'image de manière pyramidale, et en concaténant et pondérant les histogrammes obtenus à chaque niveau. Les histogrammes sont classifiés en utilisant des noyaux intersection d'histogrammes, particulièrement efficaces. Cette méthode, ré-implémentée par Griffin et al. [81], sert actuellement de référence en catégorisation d'objets pour les bases Caltech-101 et Caltech-256. Bosch et al. [26] proposent une méthode similaire s'appuyant sur des pyramides d'histogrammes d'orientations de gradients.

Hare et al. [88] utilise une méthode de factorisation algébrique (*latent semantic indexing*), inspirée de la classification de documents textuels, pour projeter les images dans un "espace sémantique" construit à partir de "termes" visuels.

Carneiro et al. [32] n'utilisent pas explicitement de vocabulaire visuel, mais représentent les images par les paramètres d'un mélange de gaussiennes estimé sur des caractéristiques locales.

2.2.4 Les approches compositionnelles

Nous appelons "approches compositionnelles" les approches cherchant à reconnaître des parties de l'image avant d'étiqueter le tout (éventuellement). Dans ce schéma, l'image est considérée comme une composition d'objets, ou plus généralement de concepts sémantiques.

Barnard et Forsyth [15] segmentent l'image et utilisent un modèle génératif hiérarchique pour associer des labels aux différentes régions et un label global à l'image. Duygulu et al. [49] proposent un modèle de "traduction automatique", associant pareillement les régions à des mots (une région est *traduite* par un mot), et regroupent les mots correspondant à des régions similaires. Jeon et al. [98] modifient ce modèle en supprimant la contrainte de bijection entre mots et régions. Feng et al. [67] améliorent encore les performances à partir de cette méthode, en remplaçant la segmentation par un simple découpage de l'image en rectangles et en utilisant des distributions de Bernoulli. Fan et al. [60] utilisent une méthode similaire pour détecter des objets d'intérêts utiles pour annoter l'image.

Vogel et Schiele [190] proposent une méthode un peu différente, dans laquelle ils décomposent l'image en concepts locaux, dont ils comptent les occurrences, et qu'ils utilisent ensuite pour la classification de scènes. Ils montrent que l'utilisation d'occurrences de concepts sémantiques est bien plus efficace que l'utilisation d'occurrences de caractéristiques visuelles. Cela leur permet également d'avoir une mesure de la "typicalité" des scènes, rejoignant par là les théories de Rosch et al. [155].

2.2.5 Les approches structurelles

Ces approches intègrent une notion de géométrie par rapport aux approches linguistiques et compositionnelles. Des liens géométriques peuvent être intégrés à différents niveaux : entre les parties d'un objet, ou entre des objets d'une scène, par exemple.

Fergus et al. [69] présentent une méthode de classification représentant les objets comme des constellations de parties, intégrant des aspects géométriques, avec des modèles graphiques. Les travaux de Kushal et al. [109] vont dans le même sens et confirment l'importance des caractéristiques géométriques pour la reconnaissance d'objets. Leibe et al. [114] construisent des groupes de parties d'objets en s'appuyant non seulement sur les similarités visuelles, mais aussi sur la proximité et la simultanéité d'apparition dans les images. Ils obtiennent ainsi un vocabulaire robuste pour la reconnaissance d'objets. Par ailleurs, ils remarquent que les "mots" ainsi obtenus sont souvent liés à des concepts sémantiques (par exemple, pour une voiture, les mots correspondront aux roues, etc.).

Au niveau scénique, Aksoy et al. [5] essaient de réduire le fossé sémantique en explicitant plus particulièrement les relations spatiales entre les régions de l'image (et non plus simplement entre des parties d'objets), en utilisant une forme de grammaire visuelle. Dans un premier temps, des classifieurs permettent d'attribuer des labels aux régions. La grammaire construite permet ensuite de classifier les images selon les scènes. Datta et al. [43] exploitent aussi les relations spatiales entre les régions de l'image pour l'annotation. Gupta et Davis [83] cherchent à résoudre le problème de correspondance en exploitant non seulement les labels correspondant aux objets, mais aussi les prépositions qui les relient (par exemple, la voiture est *sur* la route), ce qui permet de réduire les ambiguïtés. Parikh et Chen [142] proposent les hSOs, *Hierarchical Semantics of Objets*. À partir d'un ensemble d'images représentant une même catégorie de scènes, ils retrouvent les

objets saillants et apprennent les liens contextuels qui les relient : par exemple, l'écran d'ordinateur est proche du clavier, et sur un bureau, on trouvera souvent aussi un téléphone à proximité.

2.2.6 Les approches compositionnelles hiérarchiques

Les approches présentées ici forment des combinaisons de parties sur plusieurs niveaux successifs. Mojsilovic et al. [131] cherchent à réduire le fossé sémantique en s'appuyant sur des indices sémantiques déterminés expérimentalement et extraits de l'image à partir d'un grand nombre de caractéristiques visuelles différentes (contours, segmentation en régions, lignes). Ces caractéristiques locales sont regroupées en caractéristiques globales sur l'image, associées à d'autres caractéristiques globales, et sont ensuite analysées pour découvrir des indices sémantiques. Les caractéristiques globales sont déjà discrétisées de manière à pouvoir leur attribuer des labels sémantiques (les couleurs, positions, textures sont nommées). Les indices sémantiques sont obtenus à partir des combinaisons de ces caractéristiques : par exemple, le ciel peut être décrit par une zone bleue, uniforme, en haut de l'image. Ils utilisent une méthode variationnelle pour apprendre les indices sémantiques à partir des caractéristiques. Sudderth et al. [174] propose un modèle apprenant hiérarchiquement les parties, les objets et les scènes. La représentation des images est similaire à Fergus et al. [69], avec ceci de plus qu'ils peuvent partager des parties entre les objets, et des objets entre les scènes. Li et al. [117] proposent une structure globale pour segmenter l'image, l'annoter (par régions et globalement) et classifier la scène en utilisant un modèle génératif hiérarchique.

Epshtein et Ullman [54] construisent une hiérarchie de caractéristiques visuelles. Partant d'un fragment informatif, ils cherchent récursivement dans celui-ci des fragments informatifs (plus petits). Dans [56], ils étendent cette méthode à des fragments "sémantiques", déterminés automatiquement.

Fidler et Leonardis [71] partent de caractéristiques extraites par filtres de Gabor, et apprennent successivement des compositions de ces parties pour construire des hiérarchies de caractéristiques contenant jusqu'à des fragments d'objets reconnaissables (comme les roues pour les voitures). Les caractéristiques des niveaux supérieurs sont spécifiques aux catégories.

Une autre méthode pour reconnaître simultanément plusieurs objets et leurs relations dans une image est celle proposée par Ahuja et Todorovic [4]. Ils construisent une hiérarchie regroupant les objets par rapport à leur apparition dans des mêmes objets : par exemple, "toit" et "cheminée" sont des objets apparaissant presque toujours simultanément, et peuvent donc partager un parent ("toit avec cheminée"). Les "vitres" peuvent apparaître sur plusieurs supports, et auront donc plusieurs parents, comme "fenêtre" et "porte".

2.2.7 Les approches communicantes

Les approches communicantes correspondent aux approches dans lesquelles on cherche à avoir un socle d'informations partagé entre les différentes catégories. Un des problèmes récurrents en apprentissage est celui du nombre de catégories que peut apprendre un modèle. Souvent, la généralisation à de nouvelles catégories n'est pas directe. D'autre part, plus il y a de catégories, plus le modèle risque d'être complexe, et gourmand en mémoire. Les approches communicantes peuvent correspondre à ces deux objectifs : faciliter l'intégration de nouvelles catégories, et réduire la complexité du système.

Perronnin [143] apprend un vocabulaire séparé en deux parties : un vocabulaire général ou universel, partagé par toutes les catégories, et un ensemble de vocabulaires spécifiques à chacune d'entre elles. Fei-Fei et al. [63] proposent un modèle permettant d'apprendre de nouvelles catégories avec très peu d'exemples. Ils utilisent une approche bayésienne, dans laquelle un modèle *a posteriori* d'une catégorie est obtenu par mise à jour d'une connaissance *a priori* du monde, en quelques observations. L'idée est d'intégrer le plus possible de connaissances utiles et génériques dans le modèle *a priori*. Wang et al. [192] modélisent la distribution de thèmes latents comme couche intermédiaire entre le vocabulaire et les catégories. Ces thèmes regroupent donc les mots du vocabulaire, et sont partagés entre les catégories. Todorovic et Ahuja [177] apprennent à retrouver des occurrences d'objets ou de sous-objets partagés par plusieurs catégories. Ils obtiennent des performances supérieures aux SPM sur Caltech-256.

Amit et al. [6] proposent une méthode de classification multiclasse mettant simultanément en commun les caractéristiques des classes et les paramètres des classifieurs. Pour cela, ils reformulent le problème d'optimisation convexe des SVM multiclasse en remplaçant la norme de Frobenius par la trace-norme. Le partage d'information se fait de manière implicite, c'est-à-dire qu'il n'est pas évident de retrouver quelles sont les caractéristiques communes à quelles classes. Torralba et al. [180] proposent une approche multi-tâche pour le problème de la classification multiclasse, utilisant du boosting et des classifieurs binaires. Ils remarquent que cette approche réduit considérablement le nombre de caractéristiques, et accélère en conséquence la classification. Alors que pour une méthode classique, le nombre de caractéristiques nécessaires croît linéairement avec le nombre de catégories, leur méthode permet une croissance logarithmique.

Plus récemment, Thomas et al. [176] proposent de faire du *cognitive feedback* : ils exploitent la reconnaissance d'objets déjà effectuée sur certaines images pour trouver des métadonnées de plus bas-niveau sur des images d'objets inconnus. Ils peuvent donc reconnaître des parties d'objets, par exemple.

2.2.8 Les approches hiérarchiques

Quelque soit le domaine, les hiérarchies sont un outil de choix lorsque l'on souhaite simplifier un problème, ou accélérer sa résolution. C'est le principe "Diviser pour régner", et il est applicable aussi pour la reconnaissance d'objets. Il n'est cependant pas évident de voir comment les classes peuvent être regroupées en hiérarchies. La plupart des hiérarchies construites automatiquement sont des hiérarchies dites "visuelles", s'appuyant uniquement sur les données numériques et non sur le vocabulaire (et l'organisation des concepts).

Vasconcelos [187] propose une méthode d'indexation des images basée sur des mélanges de densité regroupés de manière hiérarchique. Fan et Geman [62] forment une hiérarchie de classifieurs permettant d'avoir une progression dans les décisions, le noeud terminal donnant la catégorisation. Wang et al. [192], utilisent les thèmes latents trouvés et la manière dont ils sont partagés par les différentes catégories pour construire une hiérarchie de concepts. Cependant la taxonomie est obtenue comme produit de la classification et n'est exploitée à aucun niveau.

Plusieurs méthodes permettant de construire une "taxonomie visuelle", autrement dit de regrouper les catégories de manière hiérarchiques, ont été proposées très récemment et montrent que le sujet suscite un intérêt croissant. Bart et al. [18] proposent une méthode bayésienne pour estimer une taxonomie telle que chaque image est associée à un chemin dans l'arbre. Des images similaires ont beaucoup de nœuds communs sur leur chemin, et se retrouvent donc à faible distance l'une

de l'autre. Sivic et al. [167] développent également un modèle génératif, de type LDA, appelé *hierarchical-LDA*. Le modèle LDA (*Latent Dirichlet Allocation*) part de l'hypothèse que les mots visuels d'une image sont générés à partir d'un ensemble fini de thèmes latents. Le modèle de LDA hiérarchique suppose en plus que ces thèmes ont une structure d'arbre. Griffin et Perona [82] construisent des hiérarchies pour accélérer la classification : au lieu d'utiliser un algorithme multiclasse sur toutes les catégories, ils descendent progressivement dans la hiérarchie. Pour la construire, ils utilisent d'abord un algorithme pour classifier les images par catégories, estiment la matrice de confusion, et construisent un arbre de manière *bottom-up* en regroupant successivement les catégories les plus sujettes à confusion. Ils le comparent avec un algorithme *top-down* divisant successivement l'ensemble des classes par rapport à une mesure de similarité, et trouvent des performances similaires tant en vitesse qu'en taux de réussite. Marszałek et Schmid [127] étendent ce type de méthodes en relâchant la contrainte de séparabilité : si au moment de diviser l'ensemble des catégories, certaines sont sur la frontière, elles sont classifiées comme faisant partie de chacun des deux camps. La décision sur la position de la catégorie est remise à plus tard.

D'autres approches font intervenir l'humain pour construire des hiérarchies sémantiques. Rege et al. [153] utilisent l'intervention de plusieurs utilisateurs par l'intermédiaire d'une boucle de pertinence (*relevance feedback*) pour construire une hiérarchie dite "sémantique". Les catégories ne sont pas données explicitement par des labels, mais par les informations données par le retour de pertinence. En combinant les réponses des divers utilisateurs, ils estiment l'organisation des images, permettant une recherche et une navigation plus intuitives.

2.2.9 Les approches multilabels

La plupart des approches que nous avons vues jusqu'ici concernent la classification d'objets ou de scènes. Quelque fois, l'appellation de multilabel est employée pour le problème de correspondance. Nous nous intéressons plus précisément au problème de l'annotation multilabel, c'est-à-dire qu'on cherche *plusieurs* labels décrivant l'image *dans son entier*.

Le problème de l'assignation simultanée de labels indépendants à une image a été formalisé par Boutell et al. [30] : si lors de la classification, une image peut être classifiée avec raison dans deux classes données, comment la gérer au moment de l'apprentissage pour qu'elle puisse être classifiée correctement ensuite ? Les auteurs proposent divers scénarios et montrent que le plus efficace est d'utiliser l'image comme exemple positif pour chacune des deux classes (plutôt que de l'ignorer, par exemple). L'approche proposée par Carneiro et al. [32] utilise le *Multiple Instance Learning*. Dans ce modèle, les labels sont attribués à des ensembles de caractéristiques plutôt que de manière individuelle. Chaque catégorie correspond à un label, et un label est attribué à un groupe d'images. Une image peut recevoir plusieurs labels. Ils proposent un algorithme efficace pour annoter et rechercher des images.

2.2.10 Conclusion

Nous avons vu dans cette partie les méthodes d'annotation sémantique utilisant un vocabulaire simple, non structuré. Cependant on peut remarquer que des structures apparaissent déjà à plusieurs niveaux. Le contexte d'un objet dans une image se traduit par des structures compositionnelles (pour les scènes en particulier) et géométriques. Les liens de composition, dans les objets, peuvent égale-

ment traduire les liens de composition sémantiques. A cet égard, certains travaux montrent que l'on cherche souvent à rapprocher les "parties" de "concepts", par exemple [56],[71],[4] sont capables de nommer les parties d'objets sélectionnées par l'algorithme.

Enfin, il apparaît qu'il est utile d'avoir des liens entre les catégories : que ce soit pour partager l'information ou pour accélérer la classification, de plus en plus on cherche à trouver ces liens. Lorsqu'il n'existe aucune structure *a priori* sur le vocabulaire, les liens sont à découvrir. Nous proposons dans la partie suivante de voir les méthodes qui, au contraire, se basent sur un vocabulaire structuré.

2.3 Annotation d'images avec un vocabulaire structuré

2.3.1 Introduction

Les vocabulaires structurés ont été introduits dans les problèmes d'annotation d'images de diverses manières. On pourrait regrouper les travaux déjà effectués en deux groupes :

1. ceux qui exploitent les relations de composition des objets, pour la détection ou l'identification, en particulier. Il existe de nombreux modèles *part-based* qui y sont liés.
2. ceux qui utilisent les relations d'héritages entre catégories. Nous verrons que ces liens peuvent être utilisés pour partager des descripteurs, ou au moment de la classification pour combiner des classifieurs.

Cette séparation est toute relative : il est assez naturel d'utiliser la catégorie supérieure pour trouver des parties communes à 2 sous-catégories (cf Fig 2.10).

Plutôt que de séparer selon le type de hiérarchie, nous préférerons séparer les approches par rapport à la façon dont sont utilisées les relations sémantiques, c'est-à-dire à quel niveau de la chaîne traitement elles sont exploitées.

Parmi les approches s'appuyant sur des labels organisés, on trouve :

- des approches *linguistiques*, exploitant les ontologies pour enrichir le vocabulaire ou pour travailler différemment au niveau des annotations,
- des approches *par modèles/compositionnelles/structurelles*, exploitant des ontologies ou autres structures comme templates, pour la description visuelle de l'image,
- des approches *communicantes* exploitant les liens entre les mots du vocabulaires pour partager l'information,
- des approches *hiérarchiques*, exploitant les relations d'hypernymie dans une hiérarchie sémantique, permettant ou pas des réponses aux différents niveaux.

Dans la littérature, le terme de "hiérarchie sémantique", ou autre équivalent, est utilisé de manière indifférente pour les deux types de relations. Ceci est parfaitement correct, mais ne permet pas de distinguer aisément la différence de nature qui existe entre les deux approches.

De nombreux articles de recherche d'images par le contenu s'inspirent des méthodes linguistiques, et en particulier, ils sont nombreux à exploiter WordNet [66]. WordNet est une base de données lexicale en langue anglaise, particulièrement riche, qui pour chaque mot donne entre autres la polysémie, la synonymie, les hypernymies/hyponymies, les méronymies/holonymies. Un exemple de mots liés au mot voiture est donné dans le tableau 2.3.

Avec le développement du web sémantique, des standards apparaissent pour l'annotation des documents, et donc aussi des images et vidéos (MPEG-7, voir

Relation	**Nom**	**Définition**
Synonymes	voiture, automobile	Un véhicule motorisé ayant quatre roues
Hyponymes	ambulance	Un véhicule transportant des personnes en direction ou à partir d'un hôpital.
	break	Une voiture avec une carrosserie allongée, des portes arrières, et un espace derrière les places arrières.
	compacte	Une voiture petite et économique
	décapotable	Une voiture dont le toit peut être replié ou enlevé
	...	
Hypernyme	véhicule motorisé	Un véhicule autopropulsé qui ne roule pas sur des rails
Méronymes	pare-choc	un système mécanique constitué par des barres à l'arrière et à l'avant du véhicule pour absorber les chocs et éviter les dégâts.
	portière	la porte d'une voiture
	moteur	le moteur qui propulse la voiture
	...	
Termes du domaine	location	Le fait de payer pour l'usage de quelque chose (maison, voiture...)
	alternateur	Terme vielli pour désigner un générateur électrique de courant alternatif (en particulier pour les automobiles)
	passager	voyageur empruntant un véhicule (bateau, bus, voiture, avion...) et qui ne le conduit pas.
Formes dérivées	automobiliste	personne qui conduit (ou voyage avec) une voiture.
	...	

TABLE 2.3 – *Définitions des relations de* WORDNET *par l'exemple : un des nœuds correspondant au mot* voiture. *Pour un mot donné, la polysémie correspond à différents nœuds. Le sens lié à un nœud est explicité par les mots synonymes qui le composent, et par la définition qui correspond. Les relations lient donc des "groupes de synonymes". La relation d'holonymie n'apparaît pas explicitement ici : elle correspond à la relation inverse de la méronymie (par ex.,* voiture *est un holonyme de* portière*).*

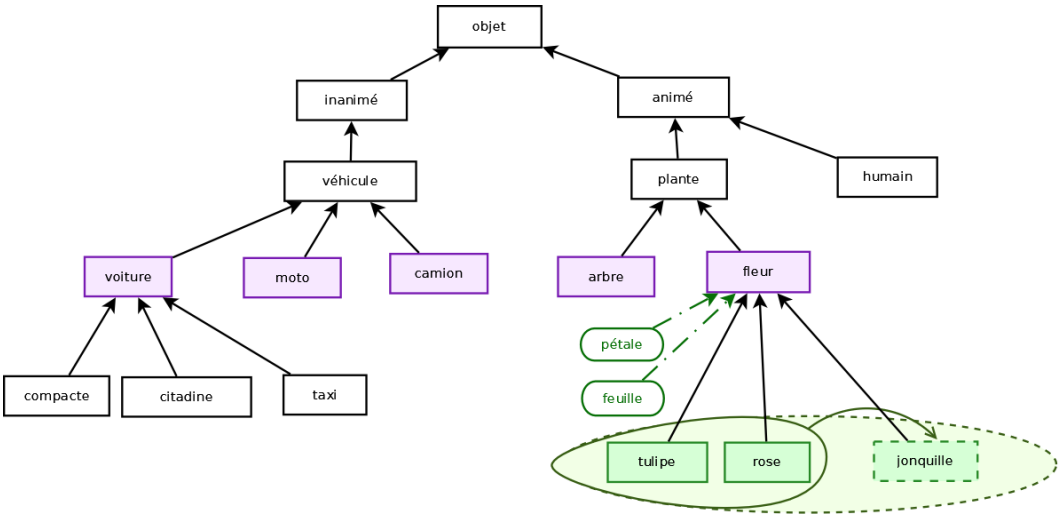


FIGURE 2.10 – *Exploitation des relations sémantiques pour apprendre de nouvelles catégories à partir d'un petit nombre d'exemples.*

Stamou et al. [173]). Nous ne citerons pas tous les travaux cherchant à s'adapter à ces nouveaux standards : ce ne sont pas les différentes manières de formatter les annotations qui nous intéressent, mais les manières de les acquérir.

2.3.2 Les approches linguistiques

Dans cette section, nous regroupons les approches exploitant une représentation des connaissances dans le but de structurer et/ou d'enrichir l'annotation d'images, mais ne les exploitant pas au niveau de l'extraction automatique d'annotations à partir du contenu-même de l'image. Aslandogan et al. [9] exploite la hiérarchie de WORDNET pour élargir le vocabulaire utilisé dans une requête ou dans la base de données elle-même. Cela permet d'aller au-delà de la simple mise en correspondance de mots, en cherchant des mots proches dans la structure WORDNET, par l'intermédiaire des relations IS-A ou MEMBER-OF, et en se limitant par rapport à la distance entre les deux mots dans l'arbre WORDNET. Yang et al. [195] étendent pareillement le vocabulaire, puis le réduisent aux termes les plus pertinents statistiquement. Leur base contient des images de 10 concepts différents, annotées à l'origine avec environ 20 mots par image. Après application de l'expansion, il y a 530 mots-clés pour les 10 concepts. Avec 1250 images utilisées pour l'apprentissage, ils obtiennent un taux de réussite moyen de 80%. Barnard et al. [16] exploitent WORDNET pour faire de la désambiguïsation : lorsqu'un mot a plusieurs sens possibles, le sens le plus pertinent est déterminé automatiquement par rapport à ses voisins dans WORDNET (hypernymes, holonymes, méronymes...) et dans l'image. Liu et al. [119] utilisent les descripteurs de MPEG-7 combinés à des ontologies et à WORDNET pour effectuer de la recherche d'image par le contenu. Une image est annotée par résolution des règles associées à l'ontologie, à partir des descripteurs de l'image en entrée. WORDNET permet d'élargir le vocabulaire et d'accéder à des termes plus spécifiques.

Datta et al. [44] exploitent encore WORDNET pour avoir une mesure de pertinence des termes sélectionnés pour annoter l'image, ainsi que pour effectuer la recherche (distance sur les mots-clés). Jin et al. [100] utilisent les relations sémantiques entre les termes et des mesures de corrélation pour éliminer les mots-clés non pertinents. Dans Lam et Singh [111], les auteurs définissent une mesure de similarité à partir de WORDNET et la combinent avec une mesure de similarité "vi-

suelle". Li et al. [117] utilisent WordNet pour supprimer des labels incohérents et pour grouper les labels synonymes.

Soo et al. [171] utilisent des ontologies pour formatter les annotations et avoir une mise en correspondance plus aisée de la requête avec des images de la base. Hollink et al. [92] combinent les annotations issues de différentes ontologies. Yang et al. [196] utilisent une boucle de pertinence travaillant directement au niveau sémantique, défini en utilisant WordNet, plutôt qu'en agissant au niveau des caractéristiques visuelles.

Wang et al. [193] extraient des mots-clés du texte associé aux images et les associent aux régions, en utilisant une structure particulière de thésaurus construite automatiquement à partir de WordNet et des co-occurrences entre mots-clés visuels et sémantiques. Popescu et al. [150] effectuent une recherche d'image par le contenu et cherchent des images similaires à deux niveaux : similarité conceptuelle, d'après les informations textuelles associées aux images, et une distance basée sur WordNet, et similarité visuelle. Ils montrent qu'il est plus efficace de chercher des images visuellement proches parmi d'autres images associées à des concepts spécifiques proches du concept cherché, plutôt que parmi des images associées à des concepts génériques. Leur méthode permet de rechercher des concepts à différents niveaux, mais les traitements hiérarchiques sont limités aux annotations et ne permettent donc pas de mise en commun des caractéristiques visuelles.

2.3.3 Les approches compositionnelles et structurelles

Nous présentons ici des approches qui pourraient être caractérisée de hiérarchiques, au sens de la relation méronymie/holonymie. Les relations de méronymies ne sont pas toujours explicitées en tant que telles, bien que tous les articles de cette section s'attachent à reconnaître des parties "sémantiques", nommées, des objets ou scènes avant de les reconnaître globalement.

Il y a deux manières principales de combiner les résultats obtenus pour les parties : soit en se basant sur des techniques d'intelligence artificielle (inférences logiques), soit en utilisant des méthodes statistiques.

Un travail précurseur est celui de Rosenthal et Bajcsy [156] entièrement basé sur un système logique. A partir d'une base de connaissance et d'une base de règles, selon les parties reconnues, le système fait des inférences pour reconnaître l'objet dans son ensemble.

Les méthodes exploitant les composantes choisies par rapport à une connaissance *a priori* du domaine sont souvent liées à la reconnaissance d'objets particuliers. Ainsi, pour la reconnaissance de personnes, Mohan et al. [130] retrouvent bras, jambes, têtes et leurs positions relatives. Pour la reconnaissance de visages, Arandjelovic et Zisserman [7] commencent par localiser les yeux, la bouche, et d'autres points fiduciaires choisis manuellement, ce qui est une manière d'introduire des connaissances *a priori*.

Dans le cas plus général des scènes, Srikanth et al. [172] reprennent le modèle translationnel de Duygulu et al. [49]. Ils utilisent la hiérarchie de WordNet à la fois pour la construction du vocabulaire et pour la classification (à partir de modèles probabilistes de mélanges).

2.3.4 Les approches communicantes

Comme on peut en avoir l'intuition en regardant la figure 2.10 (page 31), si on travaille dans un domaine restreint dans lequel toutes les catégories appartiennent

à une super-catégorie commune, (a) il y a potentiellement plus d'éléments communs entre ces catégories voisines qu'avec une catégorie plus distante ; (b) il est possible de savoir quels sont les éléments communs et en quoi ils diffèrent. Par exemple, si l'on travaille uniquement sur des véhicules, et que l'on veut distinguer entre des motos et des voitures, on pourra regarder les roues et les rétroviseurs.

Levi et al. [115] observent que les humains sont capables d'apprendre de nouveaux objets à partir d'un très petit nombre d'exemples. Ils partent du principe que des caractéristiques utiles à la reconnaissance d'une sous-catégorie seront aussi utiles pour reconnaître une nouvelle sous-catégorie (de la même catégorie). Ils construisent l'algorithme de manière à intégrer les caractéristiques les plus pertinentes en fonction de la classe.

Bar-Hillel et Weinshall [12] développent l'idée de manière plus explicite. Pour reconnaître des classes au niveau *de base*, classiquement utilisé en catégorisation d'objets, ils utilisent un modèle par parties. Pour faire des distinctions plus précises entre des sous-catégories (par exemple, entre des moto-cross et des motos de sport), ils utilisent des classifieurs discriminatifs utilisant ces mêmes parties "spécifiques". Il est intéressant de mettre ces résultats en relation avec les théories développées en psychologie cognitive, selon lesquelles, pour reconnaître des objets plus précisément, il faut une analyse plus précise (Jolicoeur et al. [102]). Des expériences suggèrent par ailleurs que les mêmes mécanismes ne sont pas utilisés dans tous les cas (Gerlach [76]), ce qui pourrait justifier l'emploi de caractéristiques visuelles complètement différentes d'une catégorie à l'autre.

2.3.5 Les approches hiérarchiques

Les travaux présentés dans cette section exploitent plus précisément les relations d'hypernymie pour favoriser l'apprentissage.

Maillot et Thonnat [122] combinent une représentation du domaine avec des techniques d'apprentissage pour catégoriser les objets, pour des domaines spécifiques. Les objets sont classifiés de manière *top-down*, c'est-à-dire que tant qu'une catégorie a des sous-catégories possibles, on cherche à classifier l'objet parmi celles-ci. Les traitements appliqués à l'image peuvent être spécifiés par rapport à la catégorie dans laquelle on cherche à classifier l'objet. Sur des images de véhicules de la base Caltech, ils obtiennent une précision entre 75% et 78% pour un rappel de 50%.

Torralba et al. [178] utilisent une base d'images de très faible résolution (32×32) compensée par le nombre (près de 80 millions d'images). Les images sont annotées avec des mots de WordNet . Ils font une catégorisation classique, en prenant un vocabulaire "plat", mais sur différents niveaux de l'arbre WordNet. Ils utilisent les K-plus-proches-voisins et une méthode de vote hiérarchique où un nœud vote aussi pour tous ses parents.

Zweig et Weinshall [203] comparent les performances d'algorithmes de classification binaire (catégorie vs. autre) selon les catégories utilisées pour l'apprentissage. Ils démontrent qu'il est utile dans certains cas de combiner les classifieurs de base avec des classifieurs correspondant aux catégories plus génériques, et précisent ces cas.

Marszałek et Schmid [126] exploitent la hiérarchie de WordNet pour faire de la classification multiclasse. Ayant extrait une hiérarchie simplifiée correspondant à leurs classes, ils proposent un algorithme partant de la racine et choisissent un hyponyme à chaque étape, jusqu'à arriver à un nœud terminal. Ils montrent que l'introduction d'une hiérarchie avec un algorithme adapté permet de mieux apprendre des concepts plus généraux (par rapport à un algorithme classique un-

contre-tous, ou un algorithme utilisant une hiérarchie "visuelle"). Ils proposent aussi d'exploiter les relations de méronymie : elles permettent d'améliorer très légèrement les performances pour certaines catégories, mais nécessitent l'apprentissage d'un beaucoup plus grand nombre de classifieurs. Par contre, les méronymies/holonymies peuvent être utilisées pour apprendre à reconnaître de nouveaux objets.

Gao et Fan [74] et Fan et al. [59; 61; 60] proposent d'utiliser une *ontologie conceptuelle* construite en combinant similarité conceptuelle (calculée à partir d'une distance dans WordNet) et similarité visuelle. Ils proposent un algorithme de boosting hiérarchique basé sur cette *ontologie*, permettant d'annoter les images à différents niveaux de généricité. La tâche de classification se fait de manière *top-down* avec un sytème évitant de propager les erreurs et essayant de les rattraper. Ils sont capables d'apprendre 120 concepts, avec 350 images par feuille pour l'apprentissage, et 120 images pour les concepts de plus haut niveau. Les auteurs ne donnent pas le taux de réussite moyen, mais la précision obtenue pour un certain nombre de concepts (précision évaluée sur toute la base de test). Alors qu'avec une méthode "plate", la précision varie autour de 70%, avec leur méthode elle varie entre 80 et 90%. Les images annotées sont des images de scène.

2.4 Evaluation

En catégorisation, les résultats sont évalués par le taux de réussite A. Celui-ci est donné par la moyenne de la diagonale de la matrice de confusion.

Les résultats d'annotations sont donnés sous la forme de résultats de détection de termes. Pour un mot donné $\mathcal{C}$, soit N_t le nombre d'images de la base de test annotées avec $\mathcal{C}$ comme vérité terrain. Soit N^+ le nombre d'images classées comme contenant un $\mathcal{C}$ et N_t^+ le nombre d'images correctement annotées avec $\mathcal{C}$. La précision P et le rappel R sont définis par :

$$P = \frac{N_t^+}{N^+} \quad ; \quad R = \frac{N_t^+}{N_t}. \tag{2.1}$$

En classification, le taux de réussite A pour une seule classe correspondra exactement au rappel R. Pour mesurer les annotations, on considère un terme comme une catégorie, et on peut utiliser les mêmes mesures.

Les méthodes présentées ici ont été évaluées sur différents jeux de données, en général par le taux de réussite en classification, et/ou par les valeurs de précision et de rappel en recherche d'images. Les résultats sont résumés dans le tableau 2.4. Le premier constat que l'on peut faire porte sur l'écart de performances entre la classification et la recherche d'image. Cet écart est directement lié au nombre de catégories et au nombre de mots-clés employés. Alors que les méthodes sont plus ou moins les mêmes (extraction de caractéristiques images, apprentissage de mots-clés correspondants), cela montre qu'il est difficile de conserver de bonnes performances lorsque la complexité du problème (c'est-à-dire la taille du vocabulaire) augmente. Par ailleurs, pour apprendre un plus grand nombre de termes, il faut augmenter aussi le nombre d'images d'apprentissages. De ce fait, le temps de calcul nécessaire au traitement acquiert de l'importance. Li et Wang [116] insistent sur la rapidité des calculs et sont par ailleurs capable d'apprendre à annoter les images avec relativement peu d'exemples.

Il faut encore noter que la taille du vocabulaire ne suffit pas à expliquer les différences de performances. En effet, les images constituant la base d'images (et notamment la base de test) peuvent être plus ou moins difficile. Pour exemple, Griffin

et al. [81] montrent que la chute de performance entre Caltech-101 et Caltech-256 en utilisant le même algorithme n'est pas seulement due au nombre de catégories : en utilisant 100 catégories de Caltech-256, ils obtiennent seulement environ 40-45% de taux de réussite. Les variations intra- et inter-classes font de Caltech-256 une base naturellement plus difficile que Caltech-101. D'une manière générale, on retiendra que les performances relatives des algorithmes testés sur des données différentes sont donc difficilement comparables.

Par ailleurs, les performances sont toujours rapportées en terme de précision/rappel/taux de réussite. D'une part, les confusions sont toujours traitées de la même manière : confondre un chien et un chat est pénalisé tout autant que confondre un chien et une voiture. D'autre part, ces chiffres devraient être mis en relation avec le niveau de généricité/spécificité (c'est-à-dire de précision sémantique) pour être plus pertinents. En effet, un taux de performance faible pour un objet ultra-spécifique n'est pas aussi problématique que pour un objet générique.

2.5 Discussion

Cependant, les résultats sont encore très insatisfaisants en pratique, si bien que les systèmes tels qu'ALIPR[4] [116], sont encore bien loin d'être adoptés par le grand public[5]. Quelques exemples d'annotations obtenues par ALIPR sont présentées figure 2.11. Une solution en pratique pourrait être l'introduction d'un apprentissage en ligne, permettant à l'utilisateur d'affiner les résultats (comme par exemple dans le système RETIN de Gosselin et al. [78]).

À la fin des années 90, on a expliqué que le problème venait de l'inadéquation avec les besoins de l'utilisateur, les systèmes proposés ne cherchant que des similarités visuelles, et aucunement sémantiques. Ce problème a été largement traité depuis, mais avec des résultats probants pour un petit nombre de catégories seulement. Après une difficulté de définition, on est maintenant confronté à un problème d'échelle : le système visuel humain reconnaît facilement des milliers de catégories, mais les ordinateurs en sont bien loin. Le problème du fossé sémantique n'est pas encore résolu.

Nous avons vu que les méthodes exploitant des hiérarchies comme WordNet ont permis d'agrandir le vocabulaire utilisé pour les annotations. Cependant peu de travaux les exploitent dès le stade de la reconnaissance, la plupart se cantonnant à un travail *a posteriori*, sur des annotations déjà existantes.

Un mouvement dans ce sens semble pourtant commencer. Jusqu'ici nous n'avons pas abordé le problème des bases de données, qui est pourtant crucial pour tout problème de reconnaissance d'objets (Ponce et al. [149]). Jusque récemment, il n'existait pas de bases annotées avec des multilabels associés à une hiérarchie. La plupart des travaux utilisaient des bases où les images sont réparties en catégories, et projetaient ces catégories dans une hiérarchie, telle WordNet. Depuis 2007, Barnard et al. [17] ont mis en place une base destinée à l'évaluation des algorithmes de correspondance : 1014 images sont segmentées, et les régions annotées avec des termes de WordNet. Un système d'évaluation exploitant les relations sémantiques est proposé. Cependant les images sont issues de la base Corel, et donc soumises à des restrictions de droits d'auteur. Depuis 2009, avec ImageNet, Deng et al. [46] ont proposé une première tentative pour créer une base

4. http://www.alipr.com/

5. voir par exemple les réactions à l'article ALIPR Helps People Decide : Hot Or Not ?, à l'adresse http://gizmodo.com/213698/alipr-helps-people-decide-hot-or-not

Système/Article	Nombre de catégories	Nombre de mot-clés	Nombre d'images (apprentissage)	Type d'images	Performances
Classification					
Yang et al. [195]	10	530	2500	Paysages	A : 80%
Lazebnik et al. [113]	101	N/A	30 par catégorie	Caltech-101	A : 64,6%
Griffin et al. [81]	101	N/A	30 par catégorie	Caltech-101	A : 67,6%
Griffin et al. [81]	256	N/A	30 par catégorie	Caltech-256	A : 34,1%
Bosch et al. [26]	256	N/A	30 par catégorie	Caltech-256	A : 45,3%
Marszałek et Schmid [126]	10	42/563	1277	Pascal VOC 2006	A : 80-90%
Bosch et al. [27]	6	N/A	600	Corel, Scènes de nature	A : 76,92%
Vogel et Schiele [190]	6	9	700 (600 tournantes)	Scènes	A : 74,1%
Fan et al. [61]	120		350+ par catégorie	Corel+LabelMe	P : 40-95%
Li et al. [117]	8 scènes, 30 objets	1256	600	Scènes de sport	R : 73% (objets), 54% (scènes)
Recherche d'images					
Srikanth et al. [172]	N/A	371 (42 prédits)	4500	Corel	P : 26,34%, R : 27,24%
Papadopoulos et al. [141]	N/A	4	40	Photos de vacances à la plage	A : 83,20%
ALIPR / Li et Wang [116]	599	332	80 par catégorie	Corel	P :≈ 40%, R :≈ 12% (1 mot)
Makadia et al. [123]	N/A	260	4500	Corel	P : 27%, R : 32%
		291	17825	IAPR-TC12	P : 28%, R : 29%
		269	19659	ESP-Game	P :22%, R : 25%

TABLE 2.4 – *Données et performances obtenues par les méthodes de l'état de l'art. A :* accuracy = *taux de réussite, R : rappel, P : précision.*

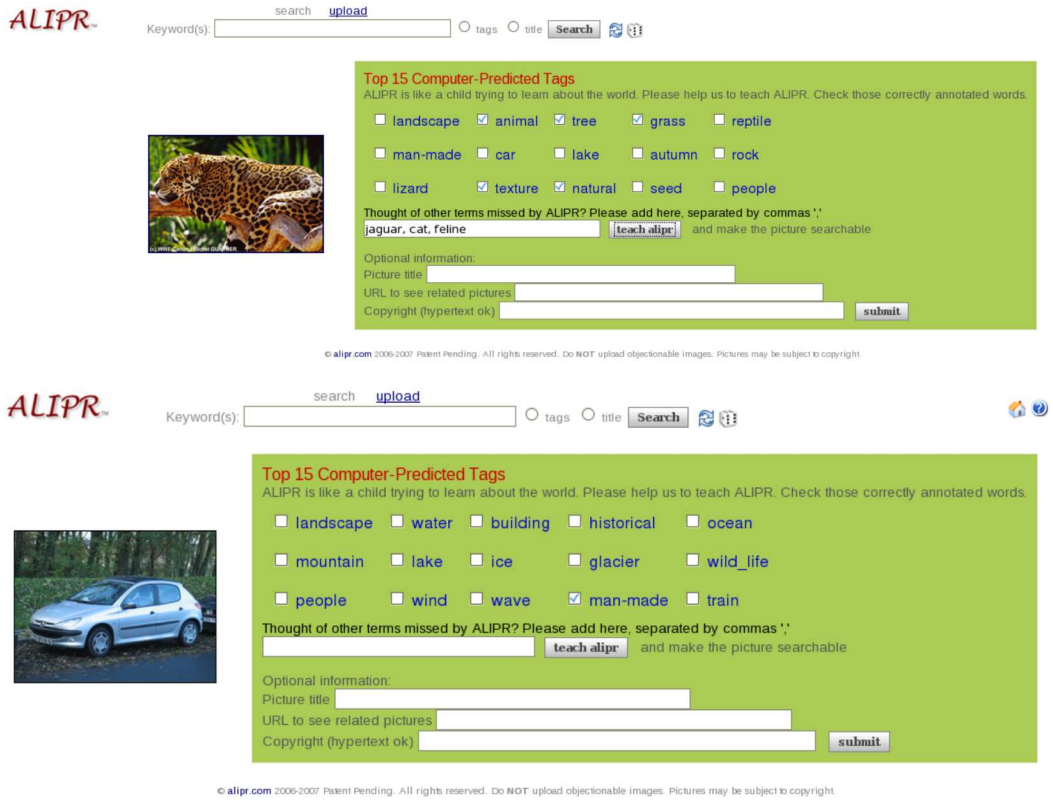


FIGURE 2.11 – *Exemples d'annotation automatique par le système ALIPR [116]. En haut, une photo de jaguar, plutôt bien annotée puisque 5 termes associés sont justes. En bas, une photo de voiture (Peugeot 206) avec des résultats très décevants : un seul terme est correct* (man-made, *i.e. "manufacturé"), et il est très générique. Les erreurs paraissent étonnantes d'un point de vue sémantique.*

d'image au vocabulaire très riche, et directement liée à la hiérarchie WordNet. Pour chaque concept, à chaque niveau, un certain nombre d'images est associé.

Les algorithmes proposant une analyse multi-niveaux devraient donc se multiplier ces prochaines années. Cependant les bases proposées ne sont pas encore prévues pour une analyse multi-facettes. En effet, dans ImageNet par exemple, les annotations sont univoques pour chaque image (une image est associée à un seul nœud), et ne sont donc pas directement multilabels.

Enfin, les algorithmes couramment utilisés ne sont pas capables de gérer la multiplicité des points de vue sémantiques. Une hiérarchie comme WordNet ne permet pas forcément certaines distinctions, et ne gère pas explicitement les contradictions entre certains termes. En effet, si on prend l'exemple du nœud *voiture* (cf. Tableau 2.3), *compacte* et *ambulance* sont des labels compatibles. En fait, ce sont deux manières indépendantes de caractériser une voiture. Ainsi, pour plus de précisions, nous préconisons d'expliciter ces différentes manières d'interpréter un objet. Ceci devrait se faire par l'intermédiaire de nouveaux nœuds, avec l'apparition de nouvelles branches dans la hiérarchie, remontant parallèlement à partir des feuilles : le graphe n'est plus un arbre. Par exemple, alors que *compacte* serait relié à un nœud du type "taille", *ambulance* serait relié à un nœud de type "fonction". Les labels correspondant à différentes tailles de voitures sont incompatibles entre eux, et de même pour les fonctions (une ambulance ne peut pas être en même temps un taxi ou une voiture de police). De nouveaux multilabels exprimant les différentes possibilités apparaîtraient (telles que ambulance-compact, taxi-compact, etc.)

2.6 Conclusion

Nous pouvons résumer plusieurs observations qui émergent de cet état de l'art :

1. le problème du fossé sémantique n'est pas résolu,
2. les approches hiérarchiques sont l'objet d'un intérêt grandissant, sans forcément utiliser la hiérarchie comme une donnée ; elles apparaissent comme une solution pratique pour réduire la complexité d'un système lorsque le nombre de catégories/termes augmente,
3. les hiérarchies sémantiques ont permis d'améliorer les performances en recherche d'images, sans même travailler directement sur le contenu,
4. une tendance récente tend à introduire les hiérarchies sémantiques directement pour l'analyse des images.

C'est dans cette dernière tendance que va s'inscrire notre travail. Peu de travaux jusqu'à présent ont utilisé la hiérarchie pour combiner les classifieurs. En novembre 2006, date à laquelle a commencé cette thèse, les premiers travaux de Fan et Gao venaient tout juste d'être publiés. Nous avons donc débuté avec un état de l'art quasi inexistant (concernant ce problème précis), à comparer avec un état de l'art gigantesque quant aux problèmes plus généraux de la reconnaissance d'objets ou de la recherche d'images par le contenu.

Les objectifs des travaux présentés dans cette thèse sont les suivants :

1. exploiter les relations de hiérarchie (hypernymies/hyponymies) pour la reconnaissance d'objets,
2. pouvoir annoter les images avec des labels à plusieurs niveaux,
3. pouvoir annoter une image avec plusieurs labels en s'assurant qu'ils soient cohérents entre eux,

4. permettre la gestion d'un compromis entre précision sémantique des annotations et fiabilité,
5. avoir un système dont on puisse "comprendre" la décision finale (le choix des annotations).

Avant d'aborder le problème en lui-même, nous introduisons dans le chapitre suivant une base de donnée adaptée, pour laquelle nous proposons d'extraire des caractéristiques images à fort contenu sémantique.

3 Bases d'images et descripteurs pour l'annotation multifacette hiérarchique

L'objectif de ce chapitre est de se munir des outils préalables à l'étude de l'annotation d'images multifacette hiérarchique. L'idée est de commencer par faire une classification "classique", sans vocabulaire structuré, sur des données pour lesquels l'organisation hiérarchique serait "naturelle". L'aspect hiérarchique sera introduit au chapitre suivant. Pour cela, nous introduisons une base d'images (section 3.1) constituée spécialement pour la reconnaissance d'objets à plusieurs niveaux, selon plusieurs points de vue descriptifs (facettes) et à l'intérieur d'une catégorie sémantique de niveau fondamental. Nous présentons les caractéristiques de cette base, et les performances obtenues par quelques algorithmes de reconnaissance de référence (section 3.2), pour évaluer les difficultés qu'elle présente. Nous décrivons ensuite un système permettant d'extraire des caractéristiques symboliques mieux adaptées au problème (sections 3.3 et 3.4).

3.1 Une base d'images pour l'annotation multifacette hiérarchique

3.1.1 Introduction

Notre objectif est d'étudier comment l'utilisation de vocabulaires structurés peut aider à la reconnaissance d'objets. Le choix d'une base d'images adaptée à ce genre de problématique est important. Nous souhaitons pouvoir reconnaître des objets assez spécifiques, à plusieurs niveaux de spécificité (voir figure 3.1), et selon plusieurs points de vue descriptifs. Il existe beaucoup de bases d'images utilisées pour la reconnaissance d'objets. Parmi les plus couramment utilisées, citons PASCAL VOC [57], Caltech-101 [64], Caltech-256 [81], ou encore LabelMe [158]. La base PASCAL VOC 2006 contient 5304 images de 10 catégories différentes, et n'est pas organisée hiérarchiquement. Pour des problèmes de catégorisation, c'est une base difficile malgré le petit nombre de catégories, du fait de la grande variété des images (orientations, occultations, échelles). Elle est également conçue pour des problèmes de détection d'objets. La base Caltech-101 est destinée à l'évaluation d'algorithmes de catégorisation avec 101 catégories d'objets génériques très divers (AK-47, visage, moto, piano, ibis, léopard, tasse...). Aucune hiérarchie n'est proposée avec la base. Une catégorie contient au minimum 31 images. La base Caltech-256 est une amélioration de la base Caltech-101, contenant 256 catégories. Une taxonomie est proposée sous la forme de deux arbres. L'aspect multifacette n'est pas directement applicable : avec autant de diversité entre les catégories, il n'est

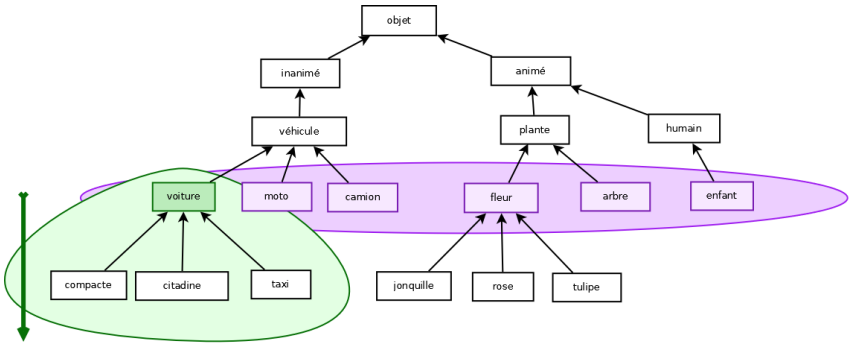


FIGURE 3.1 – *Positionnement du problème : différents niveaux de spécificité sous le niveau fondamental. L'aspect multifacette, non représenté ici, permettrait de faire la distinction, par exemple, entre les nœuds "compacte" et "citadine", caractérisant le segment de la voiture, et le nœud "taxi", caractérisant sa fonction.*

Base	Spécificité	Hiérarchie	multifacette
PASCAL VOC	-	-	-
Caltech-101	-	=	-
Caltech-256	-	+	-
LabelMe	=	+	-
ImageNet	=	+	-

TABLE 3.1 – *Bases d'images de la littérature : résumé des caractéristiques qui nous intéressent. (-) : absent, (+) : disponible, (=) : disponible sous certaines conditions. Pour Caltech-101, la hiérarchie est à construire. Pour LabelMe et ImageNet, la spécificité est contrainte par* WordNet *et par l'annotation de la base.*

pas évident de trouver des facettes qui fassent consensus. Une catégorie contient au minimum 80 images. La base LabelMe possède l'avantage d'être constamment enrichie. Plutôt que d'associer une image à une catégorie, elle propose une annotation des divers objets qu'elle contient, délimités par des polygones. Le nombre de labels n'est pas limité a priori : le vocabulaire est organisé via WordNet , ce qui permet à la fois de l'étendre et de l'uniformiser. Cependant, de la même manière que pour les autres bases de données, les descriptions restent très génériques (voiture, chaise, arbre, oiseau, tasse...). De plus, comme nous l'avons vu à la fin du chapitre 2, la structure de WordNet ne permet pas de gérer une multiplicité des points de vue [1]. La base ImageNet [46], plus récente (et qui n'était donc pas accessible lorsque ces travaux ont été entrepris), comporte les mêmes restrictions liées à WordNet.

Les caractéristiques de ces bases d'images sont regroupées dans le tableau 3.1 : aucune ne remplit complètement nos critères. En particulier, l'aspect multifacette est complètement ignoré. La constitution d'une base spécifique s'est donc imposée comme un préalable à notre étude. Nous avons opté pour une base de véhicules : par construction, ils sont organisés selon différents critères. Parmi les véhicules, nous avons choisi les voitures pour des raisons pratiques (facilité d'accès aux données). Nous détaillons la constitution de cette base dans la section suivante.

3.1.2 Constitution de la base d'images de voitures

Les bases de voitures existantes (par ex., *UIUC Cars*) sont destinées à la détection et ne sont pas adaptées à la reconnaissance des modèles, c'est-à-dire qu'elles ne permettent pas une reconnaissance plus spécifique. Nous avons donc construit

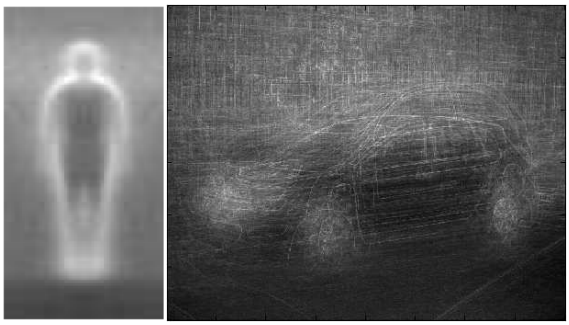
1. En fait, en trois ans, la base LabelMe s'est suffisamment augmentée pour devenir intéressante, bien que les inconvénients liés à WordNet subsistent.

FIGURE 3.2 – *Exemples de photos éliminées ou conservées de notre base d'images ; première ligne : vues de droite éliminées ; deuxième ligne : vues trop de face et de trop près, induisant des déformations géométriques, éliminées. Troisième et quatrième lignes : images conservées, montrant les variations de point de vue possibles.*

une base permettant de travailler sur ce problème particulier. Nous avons pris quelques centaines de photos de voitures stationnées dans les rues de la région parisienne. Parmi toutes ces photos, nous avons sélectionné certains modèles, les plus courants, en veillant à varier les constructeurs (Peugeot, Renault...), les segments automobiles (citadine, compacte, mini) et les types de carrosserie (berline, monospace...). 20 modèles ont été retenus : Peugeot 206 3 et 5 portes, 307 5 portes, Citroën C3 et Xsara-Picasso, Opel Corsa-1 3 portes et Zafira 1, et Renault Twingo-1 et 2 (2 versions), Clio I-1 3/5 portes, Clio I-2 5 portes, Clio II-1 3/5 portes, Clio II-2 3/5 portes, Scenic I-1, I-2 et II. Pour les Twingo-2, nous avons séparé celles qui ont des feux de brouillard à l'avant de celles qui n'en ont pas.

Par ailleurs, nous avons décidé de limiter les points de vue visuels. En effet, selon l'angle de prise de vue, l'apparence d'une voiture varie de manière importante. Par exemple, entre une voiture vue de l'avant et vue de l'arrière, il y a peu de points communs (si ce n'est la hauteur). Pour la reconnaissance de modèles, nous considérerons donc dans un premier temps un seul angle de vue – grossièrement, trois-quart face gauche. La figure 3.2 montre quelques unes des images qui ont été éliminées, comparées à quelques images conservées. Nous avons décidé de ne pas retourner les voitures vues de droite, car tous les modèles ne sont pas symétriques.

Limiter les variations géométriques lorsque l'on s'intéresse à un problème de



(a) (b)

FIGURE 3.3 – *(a) : Moyenne des images de gradients sur l'ensemble d'apprentissage pour la détection de piétons. Figure extraite de [41]. (b) : moyenne des images de gradients sur la base de voitures. La moyenne est moins lisse pour les voitures car il y a moins d'exemples.*

reconnaissance spécifique peut paraître une contrainte forte. Cependant, étant donné les variations d'apparences possibles, apprendre un objet sous toutes ses apparences revient quasiment à considérer des classes différentes pour chacune. C'est par exemple la démarche de Schneiderman et Kanade [163] : ils développent des détecteurs spécialisés pour une orientation spécifique des objets. Par exemple, pour des visages, ils détectent séparément les vues de faces et de profil. Pour les voitures, ils utilisent 8 détecteurs. Remarquons également que les détecteurs permettent de trouver une boîte englobante autour de l'objet recherché avec un cadrage stable. Par exemple, le détecteur de piétons de Dalal et Triggs [41] est entraîné sur des images très similaires (question cadrage), ainsi que le montre l'image de la moyenne des gradients (figure 3.3a). Nous ne cherchons pas à détecter des voitures, mais à identifier les modèles. Dans une chaîne de reconnaissance complète, nous supposons que la détection a déjà été faite, et que les voitures ont été localisées assez précisément, typiquement au moyen d'une boîte englobante. Dans ce contexte, il est naturel de considérer des images où les voitures sont cadrées à peu près de la même manière. Nous comparons ce cadrage à celui des piétons figure 3.3. L'image des gradients des voitures montre que les variations d'apparence sont assez importantes, en particulier au niveau de la calandre.

Au total, 644 images ont été retenues pour constituer la base d'images. Les tableaux 3.2 et 3.3 donnent quelques exemples d'images, qui permettent d'avoir une idée des variations de luminosité (sombre, surexposé, brouillard, ombres portées, reflets...), de pose (échelle, variations d'angle, hauteur de prise de vue, cadrage...) et d'environnement (immeubles, maisons, commerces, arbres...). On remarquera que certaines voitures ont des marquages publicitaires sur leur carrosserie. Les images sont prises avec une assez bonne résolution (1600x1200 ou 2592x1944). La figure 3.4 donne le nombre d'images par modèles, ce qui permet de constater le peu d'images à disposition pour l'apprentissage et l'inégalité de répartition, qui constituent des difficultés supplémentaires.

Afin d'évaluer le niveau de difficulté posé par la base d'images, nous commençons par présenter des résultats de classification sur la base de voitures avec des systèmes de l'état de l'art.

3.2 Tests de référence

Dans un premier temps, nous effectuons la classification en suivant un schéma classique, non structuré : seuls les labels-feuilles sont pris en compte.

Peugeot 206 3 portes		
Peugeot 206 5 portes		
Peugeot 307 5 portes		
Citroën C3 5 portes		
Citroën Xsara-Picasso		
Opel Corsa-1 3 portes		
Opel Zafira 1		
Renault Twingo-1		
Renault Twingo-2 avec feux		
Renault Twingo-2 sans feux		

TABLE 3.2 – *Caractéristiques de la base d'images "Voitures" construite (1/2).*

Renault Clio I-1 3 portes		
Renault Clio I-1 5 portes		
Renault Clio I-2 5 portes		
Renault Clio II-1 3 portes		
Renault Clio II-1 5 portes		
Renault Clio II-2 3 portes		
Renault Clio II-2 5 portes		
Renault Scenic I-1		
Renault Scenic I-2		
Renault Scenic II		

TABLE 3.3 – *Caractéristiques de la base d'images "Voitures" construite (2/2).*

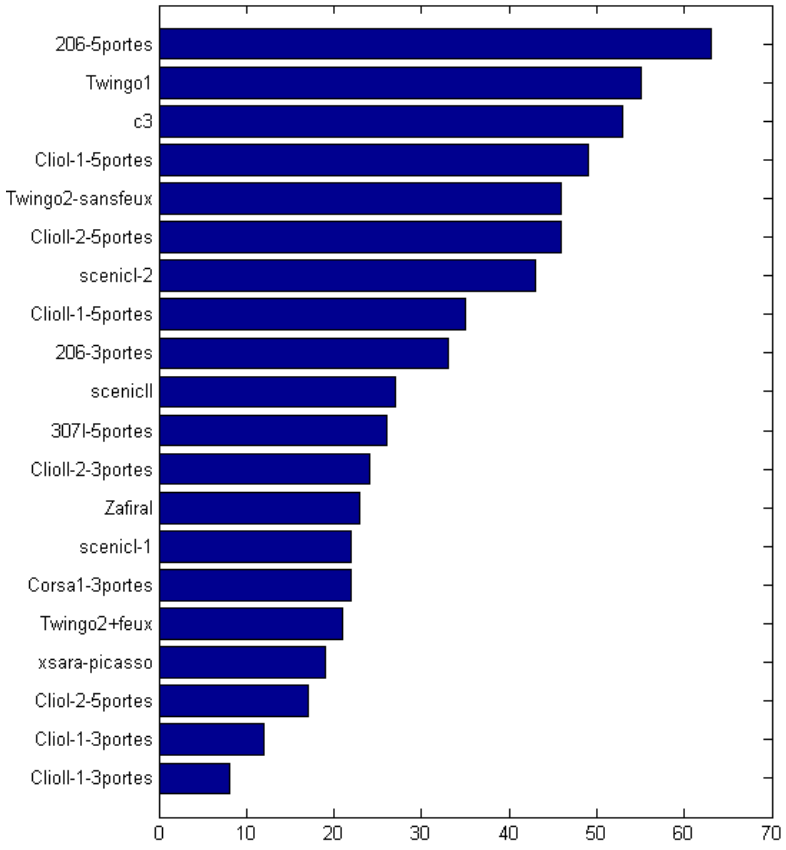


FIGURE 3.4 – *Répartition des 644 images par modèle. La répartition est très inégale entre les classes : de 63 photos de 206-5 portes à seulement 8 photos de Clio II-1 3 portes.*

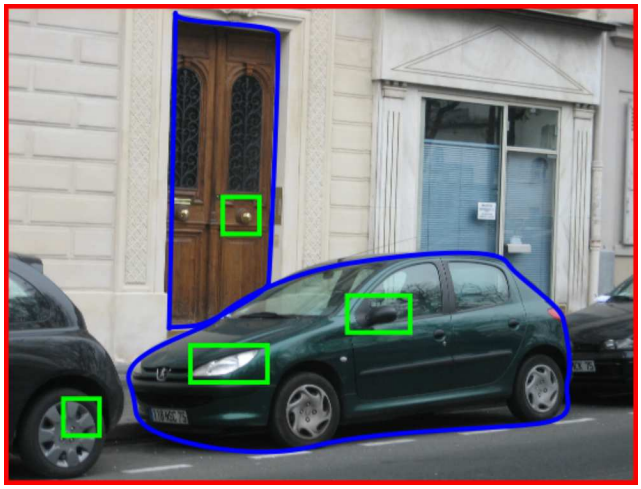


FIGURE 3.5 – *Différents niveaux d'extraction des caractéristiques images : (a) global, en rouge ; (b) régional, en bleu ; (c) local, en vert.*

3.2.1 Algorithmes de référence

3.2.1.1 Notes bibliographiques et objectifs

On peut extraire trois types de caractéristiques des images : des caractéristiques globales, "régionales" ou locales. La figure 3.5 permet de situer visuellement ces différents niveaux. Les caractéristiques globales prennent l'image dans son ensemble, indépendamment du nombre d'objets ou de leur taille. Les caractéristiques régionales décrivent des zones homogènes de l'image, issues d'une segmentation, qui dans le cas idéal correspondent aux objets. Les caractéristiques locales décrivent des petites zones ("patchs") autour de certains points de l'image. *A priori*, les caractéristiques régionales correspondent à l'échelle des objets et sont donc les plus intéressantes. Cependant, effectuer une segmentation automatique de l'image est à la fois coûteux et peu fiable. En pratique, on utilisera plus souvent les descripteurs globaux ou locaux.

Nous avons vu au chapitre 2 que la reconnaissance d'objet peut se faire à plusieurs niveaux. Les différentes échelles de l'analyse correspondent à différentes échelles d'extraction de caractéristiques. Le système visuel humain reconnaît des objets de plus en plus précis en considérant des caractéristiques de complexité croissante.

Nous testons plusieurs algorithmes de reconnaissance d'objets :

- un classifieur K-plus-proches-voisins sur les images transformées en vignettes,
- un classifieur SVM sur les images décrites par leur enveloppe spatiale,
- un classifieur basé sur les distances et des *eigencars*,
- un classifieur SVM sur les images décrites par leurs Spatial Pyramid Matching.

Les trois premiers sont basiques et s'appuient sur des caractéristiques purement globales. Le dernier, plus sophistiqué, calcule des caractéristiques globales à partir de caractéristiques locales. Nous détaillons chacun de ces algorithmes dans ce qui suit.

3.2.1.2 Vignettes

Une des manières les plus directes de représenter les images est celle utilisée par Torralba et al. [178] : chaque image est redimensionnée en une imagette

de très faible résolution, 32×32. Cette dimension a été choisie expérimentalement, en testant le taux de reconnaissance à différentes résolutions par des sujets humains. Les humains sont capables d'interpréter des scènes et de détecter des objets sur des images couleur de très faible résolution ; la résolution 32×32 représente un seuil en dessous duquel le taux de reconnaissance chute rapidement ; au-dessus, on pourra reconnaître plus de détails, mais le taux de reconnaissance n'augmente pas de manière significative. Les imagettes normalisées forment les vecteurs caractéristiques, de dimension $d = 3072$. Torralba et al. montrent que ces simples caractéristiques peuvent donner de bonnes performances avec des K-plus-proches-voisins, lorsque le nombre d'images est très grand (de plusieurs milliers à plusieurs millions) et pour des catégories génériques. Nous les testons ici pour des catégories spécifiques, et avec peu d'images, afin de démontrer, tout simplement, que le problème n'est pas trivial. Pour nos tests, nous avons repris la résolution 32×32, et nous avons fixé $K = 5$ pour les K-plus-proches-voisins (par validation), pour 11 exemples d'apprentissage par catégorie. Dans les expériences, nous surnommerons cette méthode TINY.

3.2.1.3 Gist

En vision par ordinateur, on appelle *gist* une représentation d'une image en faible dimension qui contient suffisamment d'information pour reconnaître la scène, ou le contexte [178]. En fait, tout descripteur global, pour être utile, doit s'approcher du *gist*. Oliva et Torralba [138; 139] tentent de capturer le *gist* de l'image en analysant les fréquences spatiales et les orientations. Le descripteur global est construit par combinaison des amplitudes obtenues en sortie d'un banc de K filtres de Gabor à différentes échelles et orientations. Pour réduire la dimension, chaque image en sortie du filtre est redimensionnée à une taille $N \times N$ (N entre 2 et 16), ce qui donne un vecteur de dimension $N \times N \times K$. Cette dimension est encore réduite par l'intermédiaire d'une analyse en composantes principales (ACP), qui donne aussi les poids appliqués aux différents filtres. Pour nos expériences, nous avons utilisé le code fourni par l'équipe LEAR [48] disponible en ligne [2], en gardant les mêmes paramètres et en redimensionnant toutes les images à la taille 128×128. La dimension du descripteur initial est égale à 960, elle est réduite à 50 via l'ACP.

3.2.1.4 Analyse en composantes principales

Une des premières méthodes utilisée en reconnaissance faciale est celle des *eigenfaces*, proposée par Turk et Pentland [184]. Nous l'appliquons selon le même principe aux voitures. Pour commencer, nous n'effectuons aucun recalage au préalable, considérant les variations de position et d'échelle suffisamment faibles pour appliquer directement la méthode. Les images sont redimensionnées à la taille $N \times N$. Dans un deuxième temps, nous appliquons un recalage à toutes les images, par rapport à une image de référence. Le recalage est obtenu par un algorithme robuste. Nous n'observons pas de transformations aberrantes. Nous effectuons des tests avec différentes valeurs de N, avec et sans couleur, avec et sans recalage, et avec l'image brute ou son gradient. Nous reportons les résultats en niveaux de gris (EIGEN), niveaux de gris recalés (EIGEN-R), et gradients recalés (EIGEN-R-GRAD). La

2. Gist : code disponible à l'adresse suivante : http://lear.inrialpes.fr/src/lear_gist-1.0.tgz

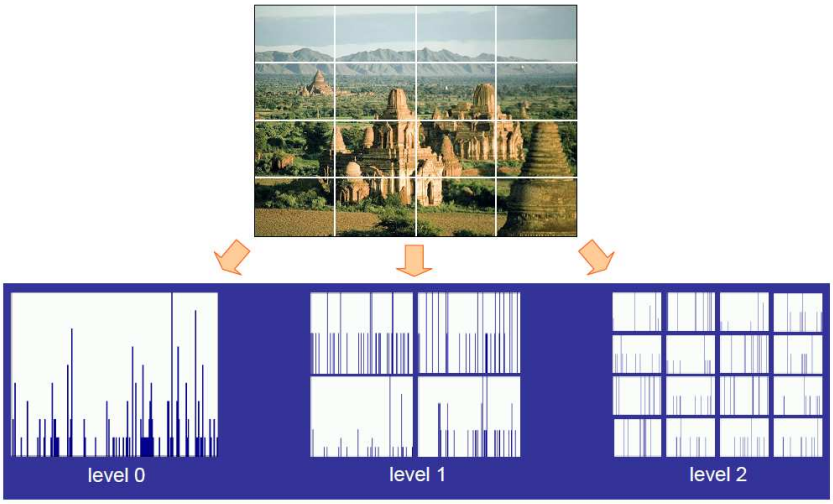


FIGURE 3.6 – *Construction d'un descripteur par Spatial Pyramid Matching : découpage de l'image et extraction des histogrammes à différents niveaux. Figure tirée de [113] (présentation).*

couleur n'apporte pas d'information significative. Le recalage est calculé grâce au code Matlab fourni par Baker [10] et disponible sur internet [3].

3.2.1.5 Spatial Pyramid Matching

Nous implémentons la méthode des Spatial Pyramid Matching, introduite par Lazebnik et al. [113], qui s'est imposée comme référence pour la reconnaissance de scènes, et qui étend le schéma sacs-de-mots. L'idée est de construire une pyramide spatiale, en découpant successivement l'image, de construire l'histogramme sac-de-mots correspondant à chacune des régions ainsi définies, et de combiner ces histogrammes en un seul descripteur. La figure 3.6 illustre ce principe.

Plus précisément, la construction des SPM est inspirée du *pyramid match kernel* proposé par Grauman et Darrell [80]. Ils travaillent sur l'espace des caractéristiques de dimension d, et suivent le schéma suivant :

1. Construction d'une suite de grilles de résolutions de plus en plus fines $0, \ldots, L$, telle que la grille de niveau ℓ découpe l'espace en 2^ℓ cellules selon chaque dimension,
2. Pour un vecteur x donné, un histogramme H_x^ℓ est calculé à chaque étage ℓ de la pyramide tel que pour chaque cellule i, $H_x^l(i)$ est le nombre de composantes de X tombant dans i. Le nombre de correspondances entre deux vecteurs x_1 et x_2 au niveau ℓ est donné par l'intersection d'histogrammes :

$$\mathcal{I}^\ell = \mathcal{I}(H_{x_1}^\ell, H_{x_2}^\ell) = \sum_{i=1}^{d} \min(H_{x_1}^\ell(i), H_{x_2}^\ell(i)). \tag{3.1}$$

   A noter, il y a un recouvrement entre les correspondances de niveau ℓ et celui des grilles de degré supérieur $\ell + 1$ (la grille i contiendra 4 grilles). Les points comptés dans l'histogramme de niveau ℓ seront donc retranchés de l'histogramme de niveau $\ell - 1$.
3. Chaque niveau est associé à un poids inversement proportionnel à la largeur des cellules, égal à $\frac{1}{2^{L-\ell}}$,

3. Recalage : code disponible à l'adresse suivante : http://www.ri.cmu.edu/research_project_detail.html?project_id=515&menu_id=261

4. En regroupant tous les niveaux, on obtient le *pyramid match kernel*, défini par :

$$\mathsf{k}^L(x_1, x_2) = \mathcal{I}^L + \sum_{\ell=0}^{L} \frac{1}{2^{L-\ell}}(\mathcal{I}^\ell - \mathcal{I}^{\ell+1}) \quad (3.2)$$

$$= \frac{1}{2^L}\mathcal{I}^0 + \sum_{\ell=1}^{L} \frac{1}{2^{L-\ell+1}}(\mathcal{I}^\ell), \quad (3.3)$$

et qui est un noyau de Mercer au même titre que le noyau intersection d'histogrammes (voir annexe B.4).

Pour les SPM, les descripteurs locaux extraits sont regroupés de la même manière que pour les sacs-de-mots pour donner un nombre M de canaux (correspondant à la taille du vocabulaire). Le SPM combine les *pyramid match kernel* obtenus pour les différents canaux, en divisant l'espace géométrique de l'image (de dimension 2) plutôt que l'espace des caractéristiques, ce qui revient à concaténer les histogrammes de type sacs-de-mots obtenus sur toutes les cellules avec les poids de l'équation (3.3). Les auteurs obtiennent ainsi un histogramme de dimension $M\sum_{\ell=0}^{L} 4^\ell = \frac{M}{3}(4^{L+1} - 1)$. Par exemple, pour $M = 200$ et $L = 2$, les caractéristiques finales sont de dimension $d = 4200$.

Les SPM offrent des performances supérieures aux approches globales. Pour Caltech-101 par exemple, là où le schéma sac-de-mots classique donne environ 41% de réussite, le schéma Spatial Pyramid Matching avec une pyramide à 4 niveaux (L=3) donne plus de 64%, avec des descripteurs SIFT [113]. Mieux, avec des descripteurs beaucoup plus simples, pour lesquels les sacs-de-mots donnent 15% de réussite, les auteurs montrent que les SPM montent à 54%. Cette efficacité est largement due au fait que la division en pyramide permet d'introduire une composante géométrique : les descripteurs locaux sont groupés en différents histogrammes selon leur position dans l'image. Sans pyramide, les auteurs montrent que découper l'image en 16 cases permet déjà d'augmenter beaucoup la réussite.

Pour nos expériences, nous avons implémenté les SPM avec les paramètres de l'article original, $M = 200$ et $L = 2$. Nous avons extraits les SIFT sur une grille dense assez fine (patchs de taille 17×17 tous les 4 pixels, à 3 échelles). Avec notre implémentation, nous obtenons des résultats proches de ceux obtenus par Griffin et al. [81] sur Caltech-101.

3.2.2 Performances sur la base de voitures

Méthode

Un banc de test est mis en place de la manière suivante :

- Choix d'un nombre d'exemples d'apprentissage N_{train}, fixé en déterminant le nombre minimal d'exemples par catégorie (ou modèle) N_{min} puis $N_{train} = N_{min} - 1$,
- entraînement d'un classifieur sur N_{train} exemples par catégorie,
- test sur les N_{test} images restantes, $N_{test} = \sum_{k=1}^{K} N_{test}^k$: pour chaque image, on note y_i le label vérité terrain, et $\hat{y}_i$ le label estimé.

Le taux de classification est alors calculé en faisant la moyenne de la diagonale de la matrice de confusion, soit $a = \frac{1}{K}\sum_{k=1}^{K} \frac{1}{N_{test}^k} \sum_{i=1}^{N_{test}^k} \{y_i = \hat{y}_i\}$. Entraînement et tests sont répétés (10 fois) pour avoir une estimation plus fiable de ce taux. Pour nos expériences, on a $N_{train} = 11$.

Remarquons que pour ces tests, nous avons remplacé la catégorie Clio II-1 3 portes par des Smart, dans le but d'avoir un nombre minimum d'images plus grand. En prenant les Smart, nous avons $N_{min} = 12$, soit $N_{train} = 11$. Les résultats

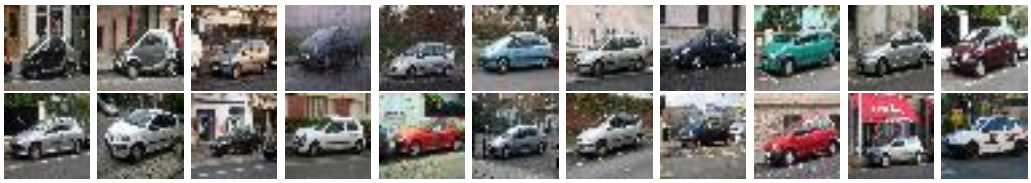


FIGURE 3.7 – *Vignettes de voitures : certaines voitures ont une forme typique et peuvent encore être reconnues.*

FIGURE 3.8 – *Voitures propres ou* eigencars *: quelques composantes principales obtenues dans la décomposition d'images de voitures (dont les 4 premières).*

obtenus ici sont donc a priori meilleurs que ceux que nous aurions pu obtenir avec la base originale : non seulement les Smart risquent moins d'être confondues avec les autres modèles, mais il y a plus d'exemples à disposition pour l'apprentissage.

Une base d'images difficile...

Nous présentons les résultats obtenus sur la base de voitures pour les différents algorithmes de référence dans le tableau 3.4. Globalement, ces résultats montrent que la base d'images est difficile : distinguer des modèles de voitures n'est pas une tâche aisée.

La figure 3.7 montre quelques vignettes de voitures sélectionnées au hasard dans la base. A l'œil nu, malgré la très mauvaise qualité des images, on peut reconnaître certaines voitures suffisamment typiques, par exemple les Smart, certaines Twingos... Des informations moins précises sont plus accessibles : on peut déterminer si une voiture est une berline ou un monospace. Cependant les faibles performances obtenues montrent que ces informations ne sont pas suffisantes.

La reconnaissance par distance dans l'espace des *eigencars* donne des résultats très mauvais. Décomposer dans l'espace des couleurs n'améliore pas ; décomposer dans l'espace des gradients n'améliore que marginalement. Il y a plusieurs explications : (a) les voitures ne sont pas bien alignées ; (b) les arrière-plans, très divers, occupent une trop grande proportion de l'image ; (c) les variations de couleurs des voitures dominent les variations de forme. L'hypothèse (a) n'est pas une explication suffisante : ajouter un recalage n'améliore pas beaucoup les résultats. Nous avons également testé l'hypothèse (b), grossièrement, en supprimant les parties hautes et basses des images, sans amélioration visible. La figure 3.8 montre les premières composantes obtenues pour l'algorithme de décomposition en composantes principales (ACP). On remarque que les différences entre les images fantômes correspondent plus à des variations de couleurs que de formes. Ces images suggèrent également que la forme des voitures est capturée de manière très approximative.

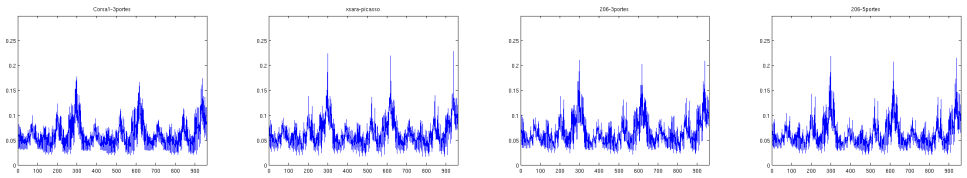


FIGURE 3.9 – *Descripteur GIST moyen pour quelques modèles de voitures : Corsa 1 3 portes, Xsara Picasso, 206 3 portes, 206 5 portes. Que les modèles soient très proches ou très différents, le descripteur GIST reste similaire.*

	TINY (K=5)	EIGEN	EIGEN-R	EIGEN-R-GRAD	GIST	SPM
20 Voitures	6,2	7,4	8,5	9,1	5,6	**29,2**
Caltech 101	8,2	-	-	-	-	**67,1**
Caltech 101 (20)	23,6	29,0	-	-	13,9	**72,7**

TABLE 3.4 – *Performances obtenues sur la base de voitures et sur la base Caltech-101 pour les algorithmes de référence (taux de réussite en %).*

Le *gist* capture des informations différentes de ces deux premiers descripteurs : on s'attend à ce qu'il soit moins sensible aux variations de luminosité et de couleurs, et capture mieux l'information de forme. Cependant les résultats montrent que ces informations globales ne sont pas assez précises pour une reconnaissance spécifique. Quelques descripteurs GIST sont présentés figure 3.9. Pour examiner le caractère discriminant du descripteur, nous comparons le descripteur moyen de quelques classes, ce qui montre que le GIST capture l'information commune aux voitures plutôt que l'information discriminante. Le GIST est un descripteur global par excellence : il n'est pas étonnant qu'il ne distingue pas les détails, et qu'il donne des résultats encore moins bon que les vignettes, tout juste meilleurs que le hasard.

Enfin, comme on pouvait s'y attendre, l'algorithme Spatial Pyramid Matching donne des résultats largement meilleurs que les autres méthodes. Ils restent cependant très insuffisants.

... plus difficile que Caltech-101

Les taux de classification sont comparés avec ceux obtenus pour la base Caltech-101 dans le tableau 3.4, et les matrices de confusions sont données figure 3.10, dans le cas SPM. La figure 3.11 donne les taux de classification par catégories pour la base de voiture et permet de visualiser quelles sont les catégories les plus performantes.

Les catégories de Caltech 101 sont plus génériques et plus faciles à reconnaître : malgré le nombre supérieur de catégories (101 pour Caltech, 20 pour les voitures), on obtient de meilleures performances. En reprenant les mêmes conditions d'apprentissage (20 catégories, 11 images d'apprentissage par catégorie), on obtient des performances encore meilleures.

3.2.3 Conclusion

Ces quelques expériences nous permettent de tirer plusieurs conclusions. Tout d'abord, les informations globales de couleur sont inexploitables directement : la couleur d'une voiture renseigne peu sur son modèle[4]. Les informations de forme, telles qu'elles sont capturées par le *gist*, sont également peu fiables : la forme est un renseignement important, mais il est nécessaire de la décrire de manière très précise pour qu'elle soit utile. L'analyse des gradients en composante principale

4. On pourrait dire "pas du tout", mais ce serait ignorer que les modèles paraissent en un nombre fixe de couleurs bien précises. Par exemple, une voiture bleu ciel a plus de probabilité d'être une C3 qu'une Twingo, et inversement pour une voiture jaune or.

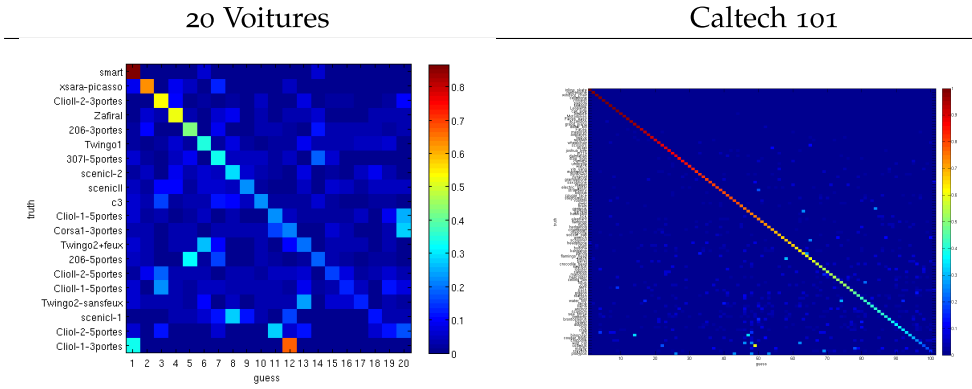


FIGURE 3.10 – *Matrices de confusion obtenues pour le Spatial Pyramid Matching.*

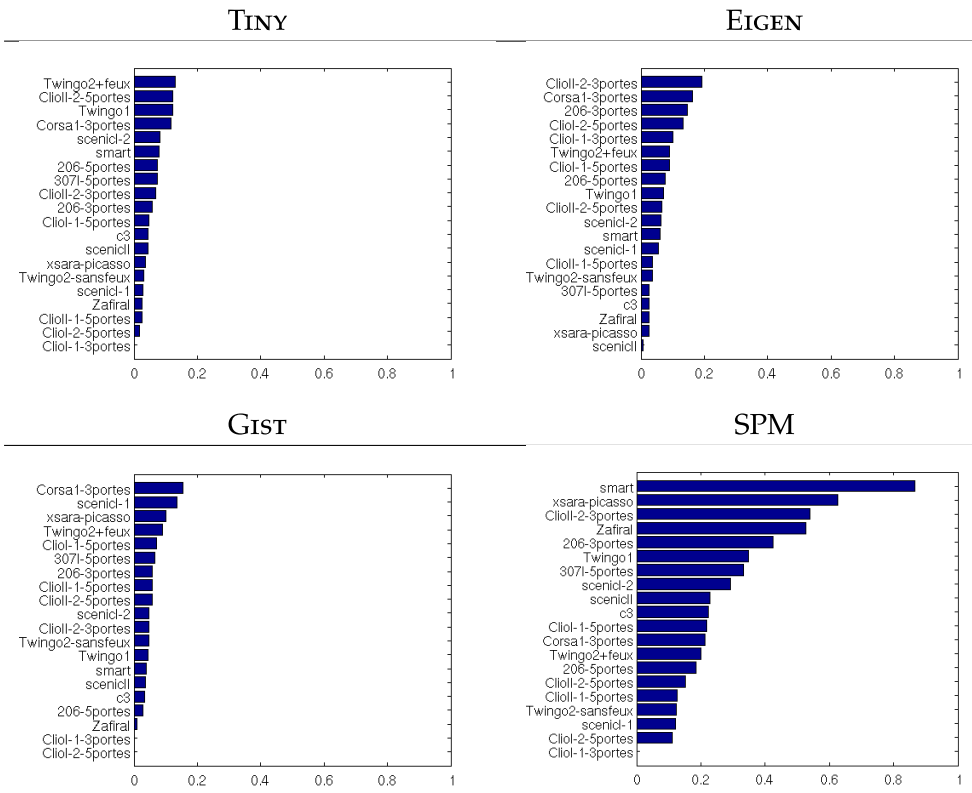


FIGURE 3.11 – *Taux de classification par catégorie sur la base de voitures.*

semble légèrement plus intéressante de ce point de vue, sans être suffisante. Enfin, les statistiques d'apparences locales ne permettent pas non plus de discriminer entre les modèles. Intuitivement, on devine que l'information locale utile se trouve en quelques points seulement. Les SPM capturent l'information sur toute l'image et intègrent beaucoup de bruit. Le vocabulaire, a priori, va se focaliser sur les caractéristiques communes à toutes les voitures avant de détecter celles utiles à la discrimination.

Il est donc nécessaire de développer des algorithmes plus spécialisés, capables de découvrir les détails discriminants pour la catégorisation de modèles. C'est l'objet de la section suivante.

3.3 Extraction de signatures spécifiques

3.3.1 Introduction

Partant d'une image $\mathcal{I}$, l'objectif est d'extraire une signature $X \in \mathbb{R}^d$, où d est le nombre de caractéristiques extraites. Dans le schéma sacs-de-mots, par exemple, d serait le nombre de mots visuels. Nous nous intéressons ici au cas plus particulier où cette signature permet de décrire un type d'image donné, c'est-à-dire appartenant à une catégorie hypernyme connue d'avance. La signature cherchée doit permettre une caractérisation plus précise des images et doit pouvoir être utilisée dans différents types de problèmes, par exemple :

Particularités de notre problème

- pour catégoriser à un niveau plus spécifique (voiture vs. minibus,...)
- pour identifier des instances d'objets (la voiture de M.Dupont, le visage de Barack Obama),
- pour repérer des paires d'objets identiques vus dans différentes conditions (une même actrice dans différents films, une même voiture sous différents angles...).

Comme nous venons de le constater, les caractéristiques globales utilisées pour la reconnaissance de scènes ou d'objets génériques s'avèrent insuffisantes pour ce genre de tâches. Par ailleurs, nous avons vu au chapitre 2 que la reconnaissance humaine se fait selon une hiérarchie. Des études à ce sujet montrent que (a) les objets du niveau fondamental sont reconnus par leurs parties, et que (b) les variations des parties permettent l'identification plus précise, c'est-à-dire à un niveau inférieur de la hiérarchie ([155, 102, 185, 90]). En vision par ordinateur, le cas (a) est représenté par les approches basées composantes, souvent utilisées pour la catégorisation ou la détection d'objets génériques, i.e. au niveau fondamental ([186, 56, 71, 99, 192, 114]).

Importance des parties

Dans notre cas, il s'agit de caractériser des objets appartenant à une même catégorie de base, visuellement très similaires, ce qui correspond au cas (b). La figure 3.12 permet de mieux situer le problème. Au niveau fondamental, les objets sont définis par les mêmes parties, ce qui permet une analyse plus spécifique. Par exemple, pour les visages, on pourra chercher les yeux, la bouche, ou d'autres parties permettant une reconnaissance plus fine (par ex., jeune femme) ou l'identification (par ex., Keira Knightley). Ces caractéristiques peuvent parfois être dites "sémantiques", dès lors que ce sont des parties d'objets pouvant être nommées. Elles correspondent aux méronymes. Alors que pour la reconnaissance au niveau fondamental, les **parties** sont reconnues de manière globale (par ex., des yeux, une bouche), au niveau inférieur, ces mêmes parties sont vues comme des **détails** de l'objet, utiles à la différenciation (yeux bleus/marrons).

Exploitation des parties dans l'état de l'art

L'idée de s'appuyer sur les parties communes est exploitée par Bar-Hillel et

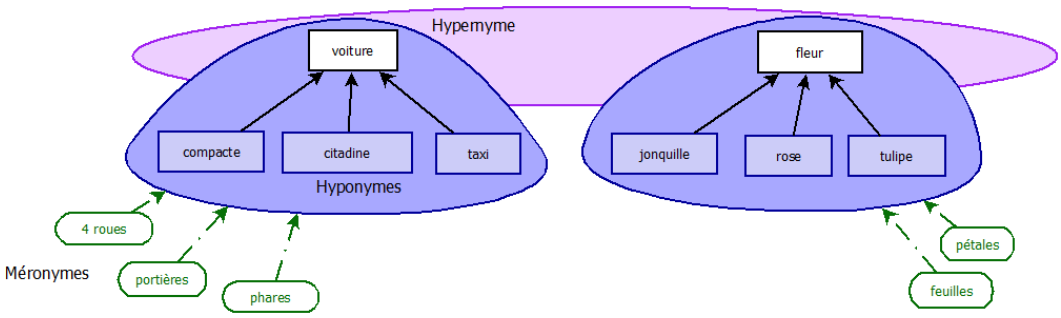


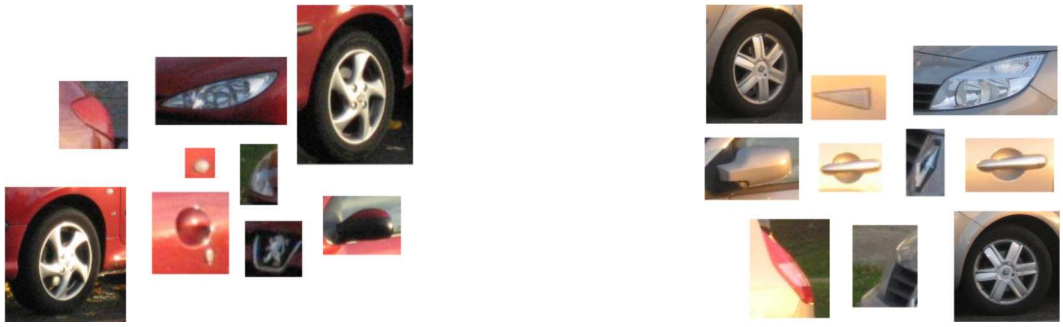
FIGURE 3.12 – *Positionnement du problème dans la taxonomie (cf figures 2.4 et 2.7). Les méronymes sont des parties d'objets avec une caractérisation sémantique. On catégorise des classes hyponymes d'un hypernyme situé au niveau fondamental. Les méronymes peuvent se rattacher à toutes les classes hyponymes.*

Weinshall [12], qui proposent d'apprendre un modèle de parties pour la catégorie, et de discriminer les sous-catégories par rapport aux variations de ces parties. Epshtein et Ullman [55] proposent de s'intéresser plutôt à des détails "satellites", localisés précisément à partir de détails communs à tous les objets de la catégorie de base. Les détails peuvent encore être utilisés pour recaler les objets, dans le but de faire une comparaison plus fine. Ferencz et al. [68] s'intéressent également à trouver des patchs distinctifs, après recalage, sur des voitures ou des visages, et dans un but plus particulier : ils cherchent à identifier si deux images représentent le même objet ou non. Dans un contexte d'identifications de personnes, Arandjelovic et Zisserman [7] extraient les positions des deux yeux et de la bouche pour recaler les images. Ils notent l'importance du contexte pour la détection, d'autant plus nécessaire que ces petites parties ont généralement une faible résolution. Dans le contexte de la classification de fleurs, Nilsback et Zisserman [136] proposent une segmentation fine, adaptée à la tâche, utilisant la notion de parties. Par contre, les parties ne sont pas directement exploitées pour la description : des descripteurs locaux et globaux sont extraits de la zone d'intérêt uniquement (histogramme HSV, SIFT, HOG).

3.3.2 Méthode

Exploitation des détails

Le fait que les objets d'une même catégorie partagent des parties a donc déjà été exploité. Cependant, l'idée a surtout été de déterminer automatiquement ces parties communes, ou alors, elles ont été utilisées pour raffiner le calcul des descripteurs (recalage, segmentation). Le possible contenu sémantique des parties n'a pas été étudié à ce niveau de reconnaissance. Nous proposons de sélectionner manuellement certaines caractéristiques sémantiques discriminantes entre les modèles, et de construire des détecteurs spécifiques pour chacune. Pour une meilleure automatisation, il aurait pu être intéressant de concevoir un algorithme découvrant automatiquement les détails utiles. Cependant, (1) cela aurait nécessité une démarche bien plus complexe que celle que nous avons adoptée (et qui déjà est complexe) ; (2) le choix manuel permet de faire l'étude de faisabilité : est-ce qu'il est *vraiment* intéressant de s'intéresser à quelques détails ? Et (3), dans le cas où les détails choisis manuellement sont efficaces, ils sera intéressant de faire évoluer l'algorithme pour que ces détails soient choisis automatiquement. En terme de robustesse et d'interprétabilité, également, le choix manuel est plus avantageux que le choix automatisé. En effet, la possibilité de nommer les détails discrimants permet d'avoir une explication plus intuitive du comportement du système. Nous avons donc déterminé des éléments, ou "indices sémantiques",



(a) (b)

FIGURE 3.13 – *Parties de voitures. Avec quelques détails et sans aucune information géométrique, une personne peut reconnaître les modèles facilement.*

(a) (b) (c)

FIGURE 3.14 – *Même avec 3 ou 4 détails, un spécialiste peut encore reconnaître les modèles de voitures.*

par rapport à leur pouvoir discriminant au moment de la reconnaissance. Une personne tâchant de reconnaître un modèle de voiture aura tendance à considérer certains détails, outre la silhouette générale du véhicule. Par exemple, le logo, la forme des phares, la forme des poignées de portes sont souvent particuliers à un modèle. Nous précisons bien que l'aspect "manuel" n'est introduit que dans la phase de conception, pour l'apprentissage de l'algorithme. En aucun cas une intervention humaine ne sera nécessaire lors de la phase de test.

L'approche choisie est du type sacs-de-mots, dans le sens où (a) on s'appuie sur des "mots" visuels, qui ici ont en plus une signification symbolique, et (b) les liens géométriques ne sont pas pris en compte. Cependant, à la différence des sacs-de-mots, nous n'utilisons pas des statistiques sur les mots visuels, mais directement l'information qui leur est liée. Dans le cas idéal, où les détecteurs trouveraient toutes les parties, et uniquement les bonnes, on aurait des résultats du type de la figure 3.13. Pour quelqu'un qui connaît un peu les modèles de voitures, ce genre d'information est largement suffisant[5]. La figure 3.14 montre qu'on peut encore reconnaître un modèle de voiture avec très peu de détails.

Principe de calcul de la signature

La signature est calculée en détectant ces détails ou "mots" visuels, et nécessite donc la construction d'autant de détecteurs que sa dimension d. On note $[d] = \{1, \ldots, d\}$. Nous utilisons une procédure de fenêtre glissante multi-échelle pour détecter chaque détail. Le calcul se déroule en trois étapes, illustrées par la figure 3.15 et que l'on peut formaliser de la manière suivante :

1. Extraction des caractéristiques sur toute l'image $\mathcal{I}$, aux échelles Σ, $\mathbf{F}(\mathcal{I}) = \bigcup_{\sigma \in \Sigma} \mathbf{F}_\sigma(\mathcal{I})$,
2. Pour chaque détecteur δ_i, $i \in [d]$:
   (a) sélection des caractéristiques dans une zone de recherche $\mathcal{I}_i \subset \mathcal{I}$, et aux échelles $\Sigma_i \subset \Sigma$, $\mathbf{F}_i(\mathcal{I}) = \bigcup_{\sigma \in \Sigma_i} \mathbf{F}_\sigma(\mathcal{I}_i)$,

5. 3.13a : Peugeot 206-3portes ; 3.13b : Renault Mégane Scénic II.

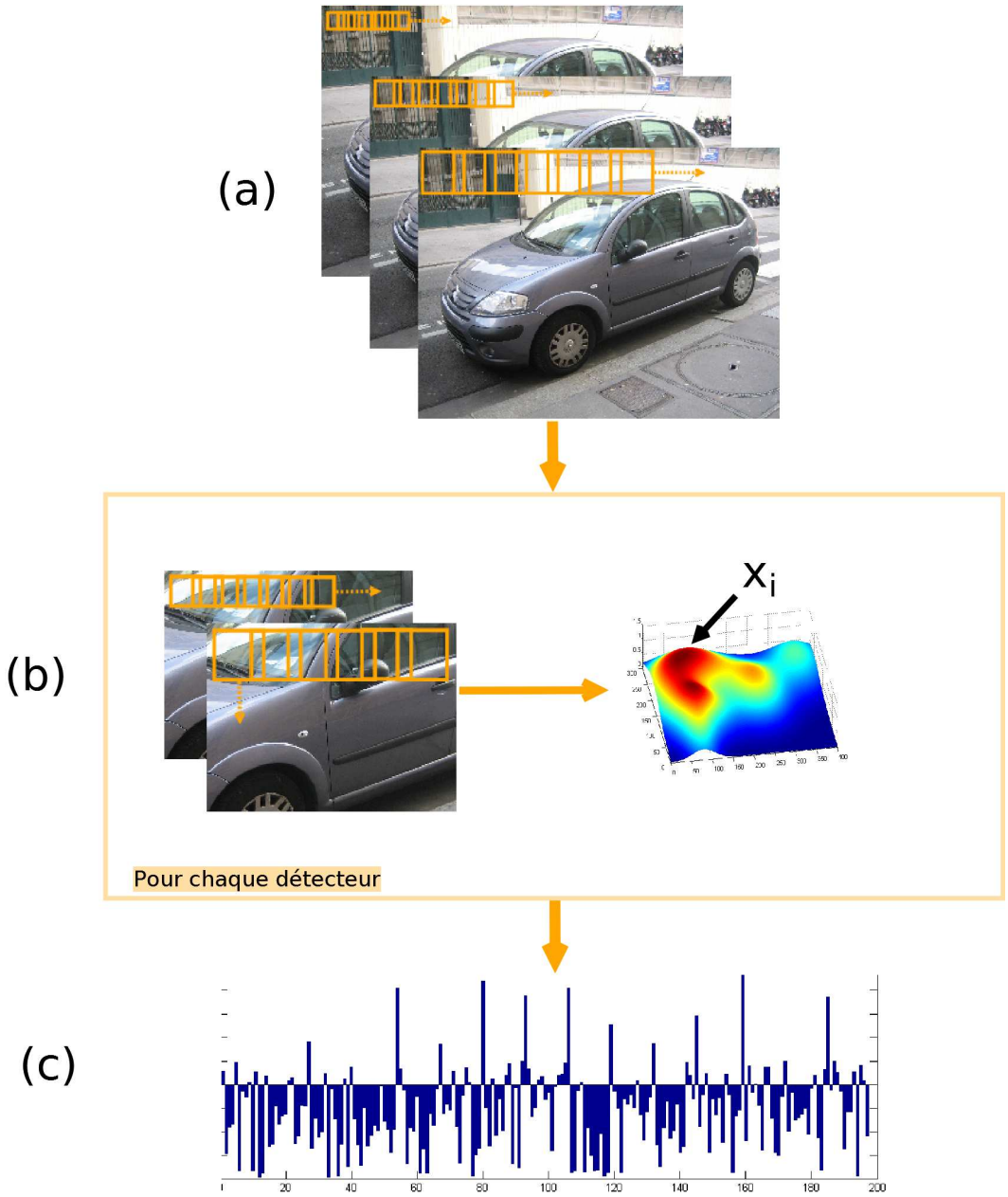
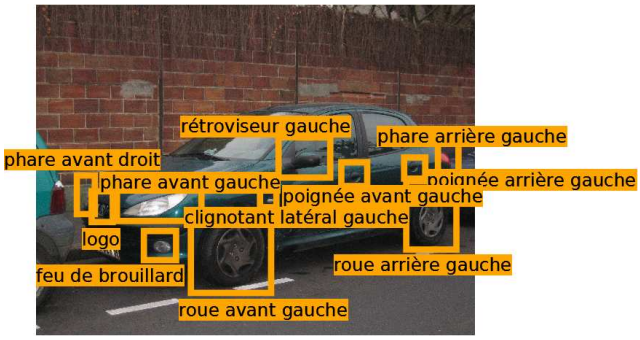


FIGURE 3.15 – *Principe des détecteurs : (a) pré-calcul des descripteurs sur toute l'image ; (b) pour chaque détecteur δ_i, extraction des descripteurs correspondants, classification, calcul de la composante x_i (score de détection maximum) ; (c) obtention de la signature $x = \{x_1, \ldots, x_d\}$.*



Détails
logo
phare avant droit
phare avant gauche
roue avant gauche
roue arrière gauche
rétroviseur gauche
clignotant latéral gauche
poignée avant gauche
poignée arrière gauche
phare arrière gauche
feu de brouillard

FIGURE 3.16 – *Annotation des détails sur une image : exemple de rectangles annotés sur une image de 206 5 portes. Au total, 11 détails sont annotées.*

(b) classification par la fonction f_i, $\forall p \in \mathbf{F}_i(\mathcal{I}),\ p \rightarrow f_i(p)$,

(c) calcul du score de détection, $\{(p, f_i(p)),\ p \in \mathbf{F}_i(\mathcal{I})\} \rightarrow x_i$,

3. Signature : $x = \{x_1, \ldots, x_d\}$.

Construction d'une base de patchs

Afin de pouvoir construire des détecteurs spécifiques, chaque image de la base $\mathcal{L}_a$ a été précisément annotée par des polygones contenant les éléments caractéristiques. La figure 3.16 montre l'annotation associée à une image de la catégorie "206 5 portes" : chaque polygone est associé à un label dénotant le type de détail. Les détails issus de l'annotation sont étiquetés par une paire de labels indiquant le type de détail, et la catégorie (le modèle de voiture). Une paire $(type, modèle)$ décrit une catégorie de patchs.

Comme nous l'avons expliqué au chapitre 2, l'annotation manuelle est un travail laborieux. Pour une annotation aussi précise des détails, il nous a fallu en moyenne 3-4 minutes par image. Ceci explique le petit nombre d'images annotées. Chaque image de $\mathcal{L}_a$ est associée à un fichier d'annotation écrit dans un langage de type XML, dont on peut voir un exemple figure 3.17. Après annotation de la base d'images $\mathcal{L}_a$, nous obtenons une base d'images de patchs contenant un ensemble d'images pour chacune des caractéristiques et pour chaque modèle. Si l'on note K le nombre de modèles, et $N_p(k)$ le nombre de caractéristiques pour un modèle $k \in [K]$, la base contient donc $d = \sum_{k=1}^{K} N_p(k)$ catégories de patchs. On note $\mathbf{\Pi} = \{\mathbf{\Pi}_i, i \in [d]\}$ la base de patchs, composée des bases de patchs de chaque catégorie.

Dans le reste du chapitre, nous détaillons la conception des détecteurs. Dans un premier temps, nous allons nous intéresser à la manière de décrire les patchs, pour trouver des caractéristiques discriminantes (3.3.3), puis pour les calculer de manière efficace (3.3.4). Dans un second temps, nous détaillerons le processus d'apprentissage des détecteurs (3.3.5). Enfin, nous présenterons les expériences effectuées et les résultats obtenus (3.4).

3.3.3 Sélection de descripteurs discriminants

3.3.3.1 Objectif

La première étape de la conception d'un détecteur est l'extraction de caractéristiques visuelles pertinentes. On appelle descripteur une fonction F qui associe un vecteur de caractéristiques à un patch : $F : \pi \rightarrow F(\pi)$. En général, un descripteur est associé à un paramètre d'échelle σ. On notera donc le descripteur F_σ,

```
<Annotation>
<Region>
<Vertex>[395,751]</Vertex>
<Vertex>[393,841]</Vertex>
<Vertex>[635,806]</Vertex>
<Vertex>[623,716]</Vertex>
<Vertex>[395,751]</Vertex>
</Region>
<Comment>phare_avant_gauche</Comment>
</Annotation>
<Annotation>
<Region>
<Vertex>[300,758]</Vertex>
<Vertex>[298,811]</Vertex>
<Vertex>[350,815]</Vertex>
<Vertex>[354,761]</Vertex>
<Vertex>[300,758]</Vertex>
</Region>
<Comment>logo</Comment>
</Annotation>
...
```

FIGURE 3.17 – *Format des annotations : exemple décrivant deux patchs.*

avec $F_\sigma(x,y) = F(\mathcal{V}(x,y,\sigma))$, où $\mathcal{V}(x,y,\sigma)$ définit un voisinage autour du pixel (x,y) Le but est de pouvoir extraire les descripteurs d'une image. Dans ce cas, on note $\mathbf{F}_\sigma(\mathcal{I}) = \{F_\sigma(x,y),(x,y) \in \mathcal{I}\}$. Le descripteur étant extrait à plusieurs échelles $\sigma \in \Sigma$, on notera $\mathbf{F}(\mathcal{I}) = \{\mathbf{F}_\sigma(\mathcal{I}),\sigma \in \Sigma\}$. L'objectif de cette partie est de sélectionner un descripteur répondant aux critères suivants :

1. **simplicité :** pour pouvoir être calculé rapidement sur toute l'image,
2. **information :** pour permettre un assez bon taux de classification patch/fond,
3. **généralisation :** le descripteur doit être discriminant pour un maximum de patchs.

3.3.3.2 Méthode

Le descripteur répondant à tous ces critères à la fois n'existe pas : l'idéal pour répondre aux deux dernières conditions serait d'avoir une combinaison de descripteurs, ce qui donnerait un descripteur complexe. Nous cherchons donc un compromis.

Pour cela, nous testons un certain nombre de descripteurs simples. Pour chaque classe de patchs, nous définissons une classe négative permettant de faire les tests préliminaires de sélection. Ces tests nous permettront de retenir quelques descripteurs intéressants qui seront testés dans les conditions réelles (c'est-à-dire avec le classifieur non-supervisé). Parmi les exemple négatifs, nous distinguons :

1. les patchs de même type, mais d'un modèle différent, par exemple logo de 206 vs. logo de Clio,
2. les patchs de type différent, par exemple logo de 206 vs. rétroviseur (de 206 ou de Clio),

3. des patchs négatifs, correspondant à des patchs sélectionnés au hasard sur les images de $\mathcal{L}_a$, en dehors des zones d'intérêt,

définissant trois "types" de négatifs.

Nous mesurons le taux d'efficacité d'un descripteur pour la classification de deux manières. La première méthode correspond à un algorithme de plus proches voisins : on observe l'évolution du taux de classification en fonction du nombre de voisins K. Les différents types de négatifs sont traités de la même manière. La deuxième méthode consiste à observer la matrice noyau obtenue avec différents noyaux. Elle permet d'observer plus précisément quels sont les patchs négatifs les mieux traités.

Nous testons les descripteurs niveaux de gris, gradients, ondelettes de haar, variance, et SIFT, calculés de la manière suivante :

niveaux de gris filtrage par un filtre moyenneur, réduction de l'image à la taille $N \times M$, égalisation d'histogramme (par la fonction de répartition).

gradients réduction de l'image à la taille $N \times M$, filtrage gaussien, calcul du gradient.

ondelettes de Haar réduction de l'image à la taille $N \times M$, égalisation d'histogramme, coefficients des ondelettes de Haar (image d'approximation et détails diagonaux).

variance réduction de l'image à la taille $N \times M$, calcul de la variance des niveaux de gris dans un voisinage donné autour de chaque pixel.

SIFT réduction de l'image à la taille $N \times M$, division du patch en $n_h \times n_w$ cases. Dans chaque case, on calcule alors un histogramme des orientations pondérées par la norme du gradient, en divisant les orientations en r classes. Pour chaque plan P de l'imagette couleur, on calcule m_P, la norme du gradient en chaque point, et θ_P, l'orientation correspondante. On représente chaque pixel i par :

$$G(i) = \max_{P \in \{R,G,B\}} m_P(i) \tag{3.4}$$

$$\Theta(i) = \theta_{\text{argmax}_{P \in \{R,G,B\}} m_P(i)}(i) \tag{3.5}$$

3.3.3.3 Expériences et résultats préliminaires

Nous effectuons des expériences à plusieurs échelles. Pour simplifier, nous prenons $N = M$ pour tous les types de patchs, avec $N = 20, 40$. Pour les SIFT, les paramètres sont fixés à $n_h = n_w = 4$ et $r = 8$. Les résultats des K-plus-proches-voisins sont présentés figure 3.18 et montrent que l'efficacité des descripteurs varie fortement d'une classe de patchs à une autre. Globalement, les SIFT se dégagent malgré tout comme un descripteur retenant bien l'information (mais au prix de la simplicité). Les simples niveaux de gris sont également capables de bonnes performances dans un certain nombre de cas.

La difficulté de séparation des patchs d'intérêt d'avec une classe négative apparaît lorsque l'on observe les matrices noyaux calculées sur un ensemble de descripteurs issus d'une base π_i donnée, et de patchs négatifs sélectionnés comme précédemment. La figure 3.19 montre ces matrices pour quelques cas intéressants. Globalement, pour les descripteurs SIFT, on peut remarquer que les confusions se font d'abord avec les catégories de patchs du même type, mais de modèles différents, puis avec des patchs du fond (sur la voiture ou non). Les confusions avec d'autres parties d'intérêt sont les moins fréquentes. On ne note pas ce même comportement pour les descripteurs niveaux de gris : les confusions sont beaucoup moins prévisibles.

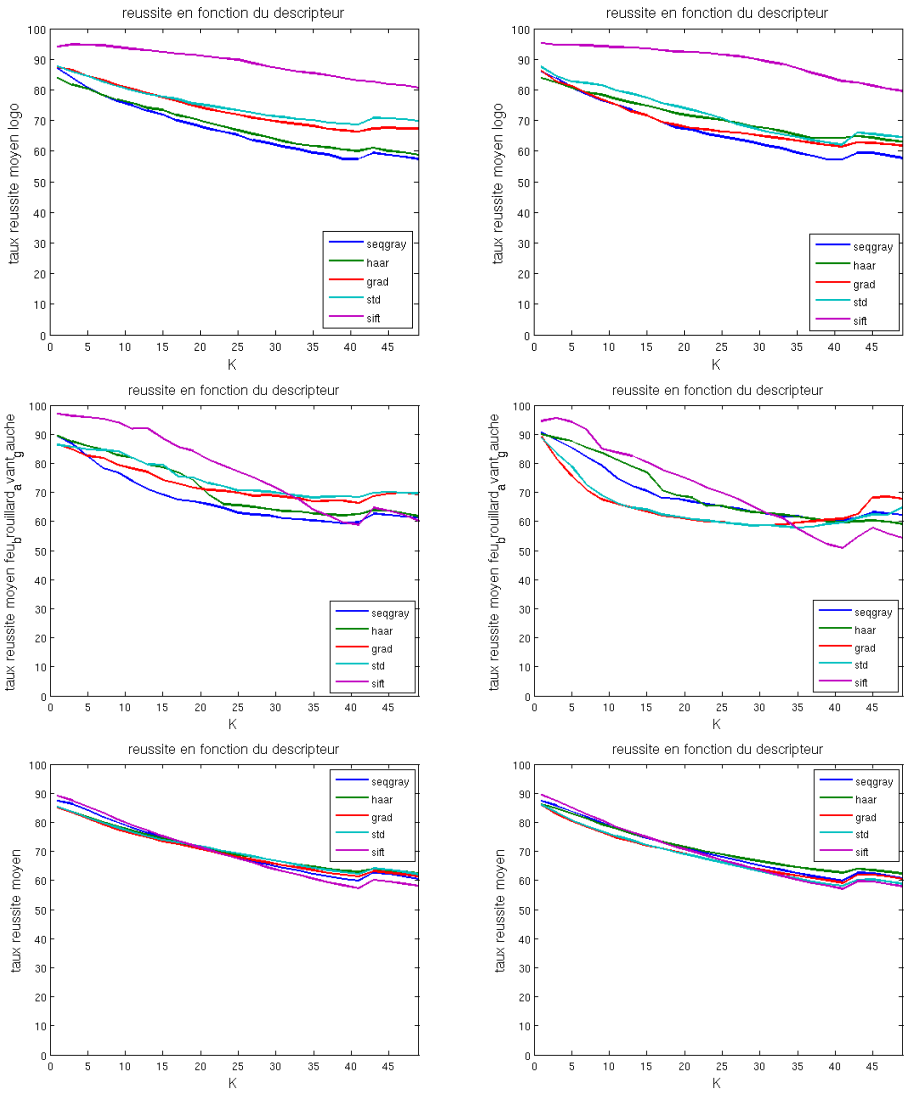
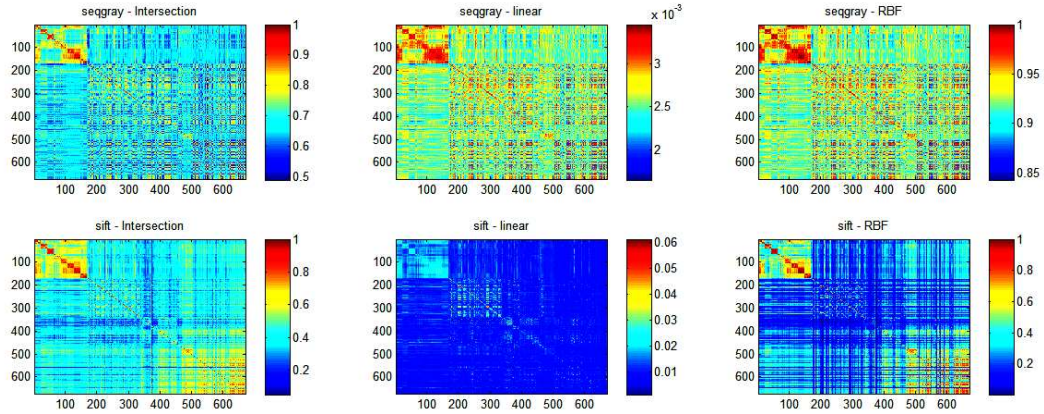
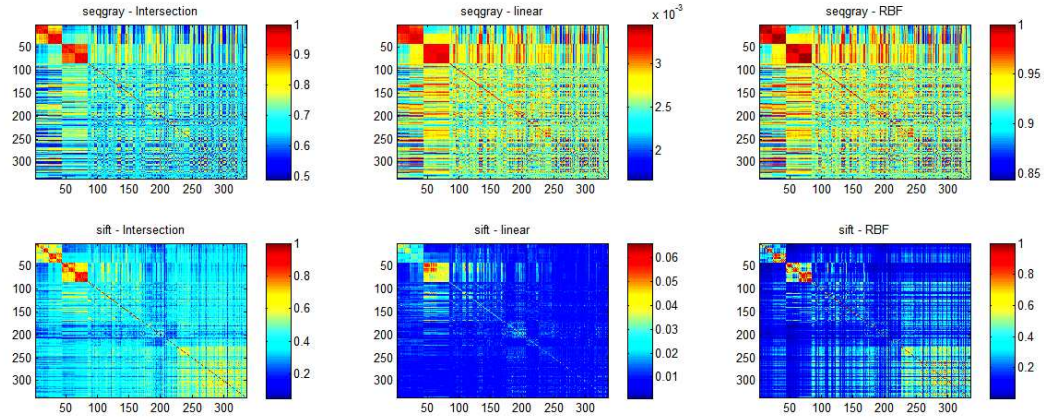


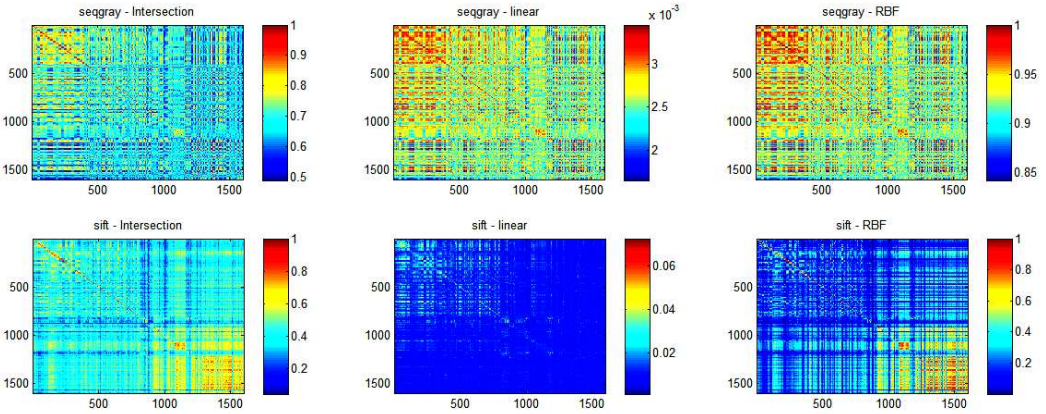
FIGURE 3.18 – *Résultats de classification de patchs obtenus avec l'algorithme des K-plus-proches-voisins : taux de réussite moyen pour K variant de 3 à N_{train} et pour différents types de patchs (moyenne sur les différentes classes pour le même type). Première colonne : avec N=20 ; deuxième colonne : avec N=40 ; Dernière ligne : résultats moyens sur tous les types de patchs.*



(a) *logo 307-I, 5 portes*



(b) *logo Clio-I-1, 3 portes*



(c) *logo Clio-I-1, 5 portes*

FIGURE 3.19 – *Matrices de Gram calculées sur les bases de patchs. Une colonne correspond à un type de noyau (intersection d'histogramme, linéaire, RBF). En ligne, différentes catégories de patchs et différents descripteurs. Une matrice est composée de quatre parties égales entre positifs, négatifs de type 1, 2, et 3 (dans l'ordre). (a) : Exemple ou la séparation positif/négatif se fait bien, pour tous les noyaux. (b) : exemple où la catégorie présente plusieurs modes, et où des confusions apparaissent avec des patchs négatifs de type 1. (c) : exemple où la classe semble difficile à discriminer, et où les niveaux de gris permettent une meilleure séparation. Ces expériences montrent par ailleurs l'importance des exemples d'apprentissage : en effet, (b) et (c) correspondent, en pratique, au même type de données.*

Pour la suite des expériences, nous retiendrons deux types de descripteurs : un complexe, le SIFT, et un simple, le niveau de gris normalisé.

Le choix d'un même descripteur pour tous les types de détails a été fait dans le but de mutualiser l'extraction des caractéristiques : celles-ci sont extraites une fois pour tous les détecteurs. Nous décrivons maintenant comment cette opération est faite de manière rapide dans le cas des histogrammes SIFT.

3.3.4 Extraction efficace des caractéristiques

Par défaut, l'extraction de descripteurs SIFT de manière dense sur toute l'image pour un grand nombre d'échelles est une opération coûteuse. Nous proposons une version simplifiée des descripteurs, permettant un calcul rapide des descripteurs sur toute l'image. Nous calculons des histogrammes des orientations de gradients de la manière décrite précédemment : chaque pixel participe pour une orientation, avec un poids donné par l'amplitude du gradient. Le gradient est calculé directement sur l'image, sans lissage. D'une part, cela évite de multiplier les lissages et d'avoir un gradient dépendant de l'échelle, et d'autre part, Dalal et Triggs [41] ont montré que le lissage dégradait plutôt les performances. Les calculs sont faits globalement en utilisant soit des histogrammes glissants, soit des histogrammes intégraux, dérivés des images intégrales.

Nous ne rappellerons pas ici la méthode des images intégrales. Elle est donnée pour rappel en annexe A.2.5. La notion d'image intégrale a été adaptée pour le calcul d'histogrammes [151, 202]. Le descripteur est composé $n_h \times n_w$ histogrammes (dits "unitaires") sur une fenêtre localisée en (x, y) et de taille $n_h\sigma \times n_w\sigma$. Un histogramme unitaire s'écrit :

$$h_\theta(x, y, \sigma) = \sum_{(x_i,y_i)\in\mathcal{V}(x,y,\sigma)} [\![\Theta(x_i, y_i) = \theta]\!] G(x_i, y_i), \tag{3.6}$$

où Θ est l'image des orientations quantifiées et G l'image des amplitudes de gradient.

On calcule une image intégrale par orientation. La figure 3.20 illustre ce principe. Soit les images des gradients par orientations O_θ définies par $O_\theta(x, y) = [\![\Theta(x, y) = \theta]\!] G(x, y)$ (on a donc $G = \sum_\theta O_\theta$). À chaque O_θ, on associe une image intégrale Ω_θ.

Un histogramme unitaire à l'échelle σ est obtenu par le calcul suivant, pour $i \in [n_w], j \in [n_h]$ définissant le découpage de la fenêtre :

$$\begin{aligned} h_\theta(x_i, y_j, \sigma) =& \Omega_\theta(x_i, y_j) + \Omega_\theta(x_i + \sigma, y_j + \sigma) \\ & - \Omega_\theta(x_i + \sigma, y_j) - \Omega_\theta(x_i, y_j + \sigma). \end{aligned} \tag{3.7}$$

On a $x = x_1, y = y_1$, et $x + (n_w - 1)\sigma = x_{n_w}, y + (n_h - 1)\sigma = y_{n_h}$. Les histogrammes unitaires sont concaténés sur les différentes parties de la fenêtre, ce qui donne r histogrammes H_θ :

$$H_\theta(x, y, \sigma) = [h_\theta(x_1, y_1, \sigma), \ldots, h_\theta(x_{n_w}, y_1, \sigma), \ldots, \ldots, h_\theta(x_{n_w}, y_{n_h}, \sigma)], \tag{3.8}$$

L'histogramme final en un point de l'image est obtenu par la concaténation des histogrammes H_θ aux r différentes orientations,

$$H(x, y, \sigma) = [H_{\theta_1}(x, y, \sigma), \ldots, H_{\theta_r}(x, y, \sigma)]. \tag{3.9}$$

suivie d'une normalisation L1 :

$$F_\sigma(x, y) = \frac{H(x, y, \sigma)}{||H(x, y, \sigma)||_1}, \tag{3.10}$$

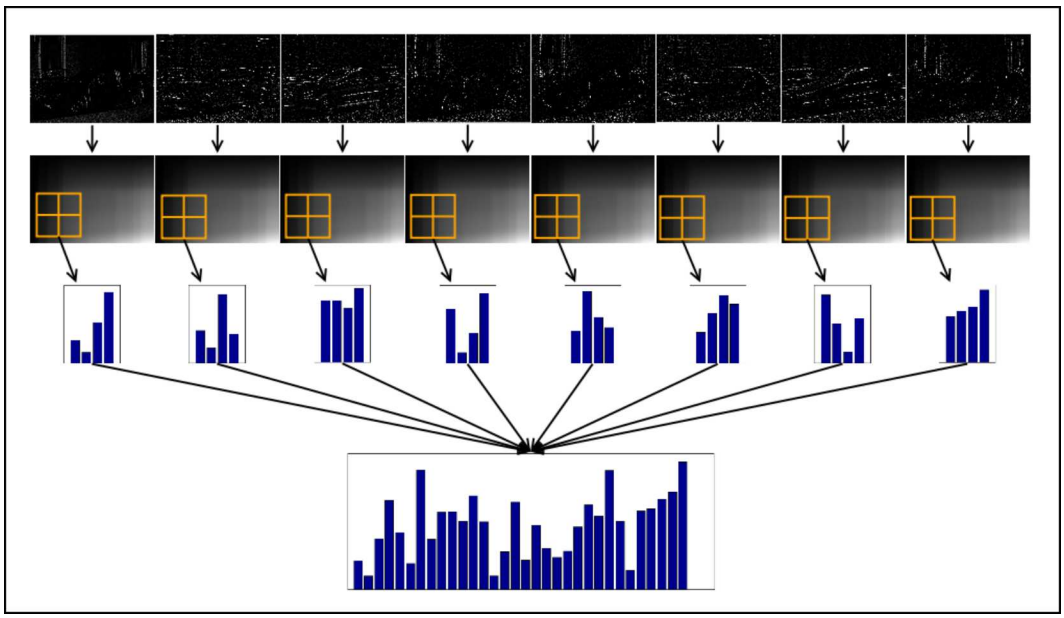


FIGURE 3.20 – *Calcul des SIFT denses par images intégrales. Première ligne : images des gradients par orientations O_θ, avec $r = 8$. Deuxième ligne : images intégrales et points de calculs pour $n_w = n_h = 2$ en un pixel (x, y, σ) donné. Troisième ligne : histogrammes H_θ obtenus en (x, y, σ). Dernière ligne : histogramme final $H(x, y, \sigma)$.*

ce qui définit le descripteur au point (x, y) à l'échelle σ.

Nous testons également le calcul des descripteurs par l'intermédiaire d'un histogramme glissant. Dans ce cas, l'ordre des calculs n'est pas tout à fait le même. En effet, on concatène directement les histogrammes unitaires aux différentes orientations :

$$h(x_i, y_i, \sigma) = [h_{\theta_1}(x_i, y_i, \sigma), \ldots, h_{\theta_r}(x_i, y_i, \sigma)]. \tag{3.11}$$

Le calcul de l'histogramme glissant doit être répété pour chaque échelle, ce qui le rend moins efficace a priori. Par contre, il ne nécessite pas de calcul supplémentaire en chaque point : il suffit de concaténer les histogrammes $h(x_i, y_i, \sigma)$ et de normaliser. Ils peuvent donc présenter un léger avantage au moment de récupérer les histogrammes.

Soit n le nombre de fenêtres à extraire, n_σ le nombre d'échelles, r le nombre d'orientations. La complexité des pré-calculs est $C_{pre}(n) = O(rn)$ pour les histogrammes intégraux, et $C_{pre}(n) = O(n_\sigma n)$ pour les histogrammes glissants. La complexité de calcul au moment de l'extraction individuelle des caractéristiques est de $C_{ext}(n) = \alpha r n n_\sigma$ pour les images intégrales, et de $C_{ext}(n) = \beta r n n_\sigma$ pour les histogrammes glissants, avec $\beta < \alpha$. La complexité totale valant $C(n) = C_{pre}(n) + C_{ext}(n)$, la sélection de l'algorithme le plus rapide dépend donc de n_σ.

Une fois les caractéristiques extraites, il est nécessaire d'évaluer leur appartenance à la catégorie d'intérêt. Nous nous intéressons maintenant à la classification des détails.

3.3.5 Classification de détails

3.3.5.1 Méthode

La signature est basée sur les résultats obtenus par les différents détecteurs appliqués à l'image. Pour pouvoir calculer une signature, il est donc d'abord nécessaire de concevoir l'ensemble des détecteurs. Un détecteur est défini par rapport à sa base d'apprentissage correspondante : il permet d'associer à chaque point

d'une image une mesure de confiance sur l'appartenance ou non du point à la classe de patchs testée.

Par rapport aux détecteurs classiques, les nôtres possèdent plusieurs particularités :

- les objets recherchés sont petits par rapport à la taille de l'image,
- ces objets sont des parties d'un objet d'une même catégorie,
- on recherche des parties spécifiques plutôt que des parties génériques (logo de 206 plutôt que logo),
- les fausses alarmes sont moins pénalisables que dans un schéma classique, puisque la quantité de détecteurs permet de réduire globalement leur effet.

Chaque détecteur δ_i est associé à un classifieur f_i. Les fonctions de classification f_i sont apprises à partir des imagettes de détails extraites des images annotées.

La plupart des méthodes de détection de l'état de l'art se basent sur des classifieurs binaires objet/fond. C'est le cas, par exemple, du détecteur de Viola et Jones [189], détectant les visages avec un classifieur adaboost entraîné de manière à contrôler le ratio entre fausses alarmes et taux de détection. Par ailleurs, en détection/localisation, la plupart des travaux étudient un problème où il s'agit de trouver un objet d'intérêt, plutôt que simplement une partie d'objet, ce qui permet d'utiliser ses composantes [130, 3, 47]. Dans notre cas, nous cherchons des parties d'objets : (a) la définition d'un fond est d'autant plus délicate (à la fois les autres parties de l'objet, les mêmes parties d'un objet d'une autre sous-catégorie, et tout autre fond) ; (b) l'utilisation de systèmes basés composantes est impossible.

Nous avons donc opté pour un système où les patchs sont décrits globalement, et où les classifieurs sont non-supervisés : ils permettent simplement l'estimation de la distribution de la classe d'intérêt. Les patchs utilisés pour l'apprentissage des détecteurs sont obtenus en annotant la base d'images. Pour cela, la base d'images est tout d'abord séparée en deux base notées $\mathcal{L}_a$ et $\mathcal{L}_b$, et seuls les détails de la base $\mathcal{L}_a$ seront annotés. La figure 3.21 reprend la figure 3.4 est montre la répartition des images entre les deux bases : en tout, 326 images pour $\mathcal{L}_a$ et 318 images pour $\mathcal{L}_b$.

La figure 3.22 montre l'approche adoptée : la base $\mathcal{L}_a$ est divisée en $N_f = 5$ parties, formant une partition $\mathcal{P}_a$. La même partition $\mathcal{P}_a$ est conservée tout au long du processus d'apprentissage. Les patchs sont extraits à partir des annotations. Le classifieur est entraîné en optimisant les paramètres par validation croisée sur la partition $\mathcal{P}_a$. Les classifieurs utilisés sur la base $\mathcal{L}_b$ sont appris sur l'ensemble de la base $\mathcal{L}_a$.

Il est nécessaire d'agrandir la base d'apprentissage artificiellement

En outre, chaque base de patchs est étendue. En effet, une des caractéristiques de notre base est le petit nombre d'images à disposition. L'apprentissage non-supervisé nécessite un plus grand nombre de données. Afin d'augmenter ce nombre à partir des données existantes, nous pratiquons de petites transformations affines sur chacun des patchs, ce qui permet de multiplier le nombre de points dans l'espace d'apprentissage. Il est important d'éviter tout sur-apprentissage. En augmentant ainsi la base d'apprentissage, il est clair que certaines images sont très corrélées, et ne répondent plus à l'hypothèse de tirages aléatoires indépendants parmi les données. Les paramètres d'apprentissage sont fixés par validation croisée, et il est donc nécessaire de garder les images liées dans les mêmes jeux de données. Dans chacune des parties de la partition $\mathcal{P}_a$, on effectue n_A petites transformations affines choisies aléatoirement pour chaque image. Comme nous l'avons déjà remarqué, ce sont ces mêmes divisions qui seront utilisées pour la validation croisée des paramètres. L'amplitude maximale $A_{\max}$ des transformations aléatoires est calculée à l'intérieur de chaque base de patch de la manière suivante :

Pour chaque patch i,

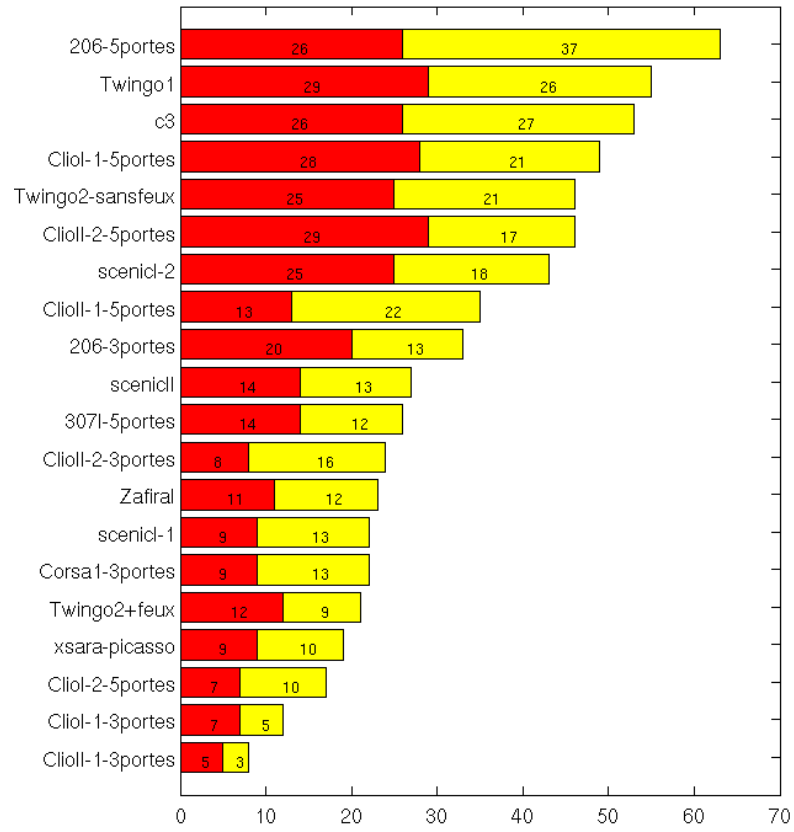


FIGURE 3.21 – *Nombre d'images par modèle pour $\mathcal{L}_a$ (en rouge) et $\mathcal{L}_b$ (en jaune). Les inégalités de répartition ne sont pas les mêmes entre les deux bases.*

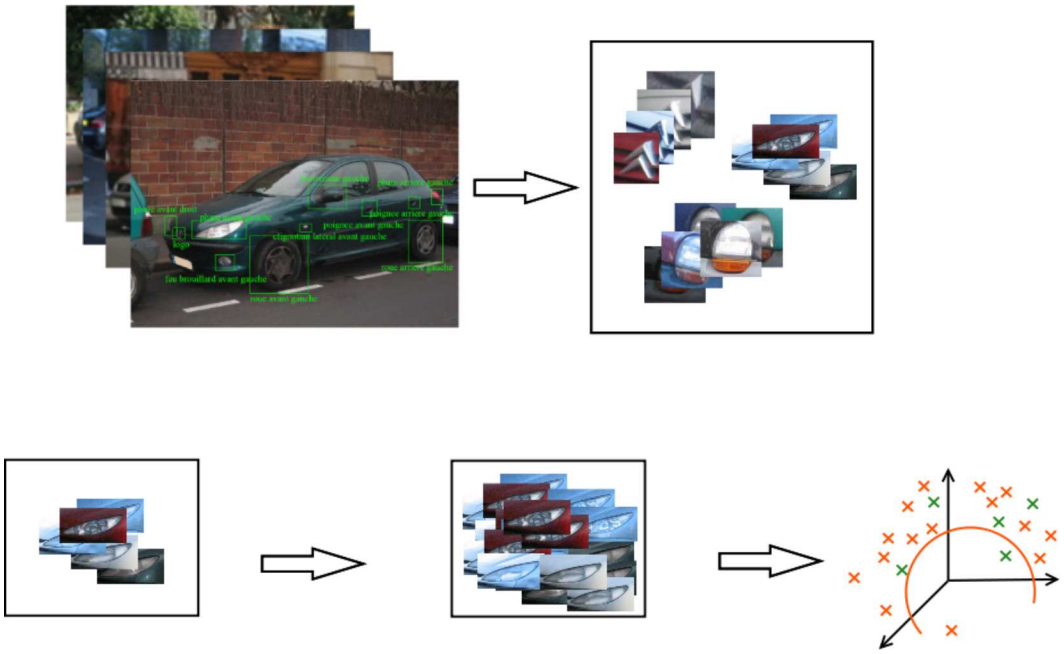


FIGURE 3.22 – *Entraînement des classifieurs pour les détecteurs. Les patchs sont extraits de la base partitionnée par $\mathcal{P}_a$. Chaque base de patchs est étendue, et utilisée pour entraîner un classifieur SVM one-class.*

1. calculer le patch le plus proche par recalage affine,
2. $A_{\max}$ = amplitude de ce recalage multiplée par un facteur α.

L'extension de la base de patchs est effectuée une fois pour toute. Nous avons choisi $n_A = 20$, et $\alpha = 1,2$.

3.3.5.2 Le SVM one-class

Les SVMs classiques sont des classifieurs supervisés. Schölkopf et al. [164] introduisent les SVM one-class pour faire de la classification non-supervisée : en estimant le support de distributions sur un espace de très grande dimension, le but est de prédire, pour une donnée test, si elle suit ou non la même distribution. Soit les données d'apprentissage :

$$x_1, \ldots, x_n \in \mathcal{X}, \tag{3.12}$$

où $n \in \mathbb{N}$ est le nombre d'observations et $\mathcal{X}$ est l'ensemble de représentation, en général $\mathcal{X} \subset \mathbb{R}^d$.

Le but est de définir une fonction g qui prend la valeur $+1$ dans une région la plus petite possible contenant la plupart des observations, et -1 partout ailleurs. De la même manière qu'avec les SVM binaires, les points sont transformés par une fonction Φ dans un espace de Hilbert à noyau reproduisant, puis ils sont séparés de l'origine par un hyperplan en maximisant la marge. Séparer les données d'avec l'origine correspond au problème quadratique suivant :

$$\min_{w \in F, \Xi \in \mathbb{R}^n, \rho \in \mathbb{R}} \quad \frac{1}{2}||w||^2 + \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i - \rho \tag{3.13}$$

$$\text{tq} \quad (w \cdot \Phi(x_i)) \geq \rho - \xi_i,\ \xi_i \geq 0. \tag{3.14}$$

Le paramètre $\nu \in]0,1[$ permet de contrôler un compromis entre le nombre de vecteurs supports et la fraction de points aberrants. Pour une observation x, la fonction de prédiction g prend la forme :

$$g(x) = \text{sgn}((w \cdot \Phi(x)) - \rho). \tag{3.15}$$

La fonction définie par $f(x) = (w \cdot \Phi(x)) - \rho$ est appelée fonction de décision, ou encore fonction score.

Le problème peut être reformulé dans le dual de la même manière que pour les SVMs classiques.

L'influence de ν est résumée par la proposition 3.1 [164] :

Proposition 3.1
- *ν est une borne supérieure sur la fraction de points aberrants.*
- *ν est une borne inférieure sur la fraction de vecteurs supports.*

Autrement dit, une petite valeur de ν permet de réduire le nombre de points aberrants, donc de donner une meilleure estimation de la distribution, et permet d'avoir moins de vecteurs supports (sans garantie). Réduire trop la valeur de ν peut mener à du sur-apprentissage.

3.3.5.3 Calcul des signatures

Dans notre cadre, les données à classifier sont les descripteurs de détails extraits de l'image, notés $p \in \mathbf{F}_i(\mathcal{I})$. Pour les histogrammes SIFT, nous testons l'utilisation pour chaque détecteur d'un SVM one-class avec un noyau intersection

d'histogramme. Nous avons déjà utilisé ce noyau avec les descripteurs SPM (cf équation (3.1)). Il est défini par :

$$K(x, x') = \sum_{i=1}^{d} min(x_i, x'_i), \tag{3.16}$$

et se prête particulièrement bien à nos données. Nous comparons ses performances avec un noyau linéaire et un noyau RBF.

Le classifieur SVM one-class nous permet de définir la fonction f_i, telle que $\forall \delta_i, \forall p \in \mathbf{F}_i(\mathcal{I}),\ p \rightarrow f_i(p)$. Le score de détection x_i est donné par le maximum des scores sur l'image : $x_i = \max_p f_i(p)$. On aurait pu chercher une meilleure heuristique pour optimiser la détection, mais (a) au vu des tests, la fenêtre de score maximum coïncide le plus souvent avec la cible, lorsque la fenêtre contenant la cible est correctement classifiée, et (b) le but n'est de toute façon pas d'avoir une localisation précise des détails (bien que ce soit un avantage). Certes, un calcul plus élaboré du score de détection permettrait éventuellement d'éliminer des fausses alarmes. Nous avons préféré utiliser des heuristiques plus simples (normalisation, seuillage) et donc moins coûteuses.

Plus précisément, nous proposons plusieurs formes de signatures :

1. directe : $x = \{x_1, \ldots, x_d\}$,
2. binaire : $x = \{[\![x_i > \theta]\!], i \in [d]\}$, où θ est un seuil de détection fixé, et en notant $[\![f]\!]$ où f est une proposition, la fonction binaire qui vaut 1 si f est vraie, 0 sinon.
3. normalisée : $x = \{\frac{x_i - \theta_i - m_X}{\sigma_X}, i \in [d]\}$, où m_X et σ_X sont calculés en lançant les détecteurs sur la base d'apprentissage $\mathcal{L}_a$.

3.4 Evaluation et discussion

3.4.1 Evaluation

L'évaluation des détecteurs peut se faire à plusieurs niveaux. Tout d'abord, nous évaluons de manière individuelle les performances de chaque détecteur. Dans un second temps, nous les évaluons globalement, en utilisant la signature pour faire la catégorisation des modèles de voitures, de manière comparable à l'évaluation faite au paragraphe 3.2.

Les performances d'un algorithme de détection sont données à la fois par le taux de fausses alarmes et par le taux de reconnaissance (le nombre de positifs correctement trouvés). Nous reprenons les définitions données au paragraphe 2.4. Pour un détecteur donné, soit N_t le nombre d'images contenant effectivement cet objet. Soit N^+ le nombre de détections, et N_t^+ le nombre d'objets correctement détectés. La précision P et le rappel R sont définis par :

$$P = \frac{N_t^+}{N^+} \quad ; \quad R = \frac{N_t^+}{N_t}. \tag{3.17}$$

Le taux de fausses alarmes correspond alors à $1 - P$, et le taux de détection à R. Les performances du détecteur sont généralement présentées sous la forme de courbes $P = f(R)$. Nous les obtenons en faisant varier un seuil de détection θ : il y a détection pour le détecteur δ_i si le score $x_i > \theta$ (c'est aussi le seuil de binarisation).

Pour un détecteur simple, un objet sera correctement détecté dès lors que le détecteur répond positivement sur une zone de l'image. Si l'on souhaite une évaluation plus précise du détecteur, intégrant sa capacité à localiser les objets, il faut

de plus intégrer une mesure dépendant de la position et de la taille de la fenêtre de détection. En reprenant la mesure utilisée pour tracer les courbes précision/rappel dans le Challenge Pascal, on évalue une détection comme correcte si la surface de chevauchement a_0 est supérieure à 50%. Notant B_p la boîte englobante prédite, et B_{gt} la vérité terrain, et $\mathcal{A}(B)$ l'aire d'une surface B, a_0 est défini par :

$$a_0 = \frac{\mathcal{A}(B_p \cap B_{gt})}{\mathcal{A}(B_p \cup B_{gt})}. \tag{3.18}$$

Les performances des signatures sont évaluées d'une manière légèrement différente que dans les expériences précédentes. En effet, compte tenu de la disparité de répartition des exemples dans les classes, l'hypothèse d'une distribution uniforme ne correspond pas aux données. Par ailleurs, étant donné que les détecteurs sont entraînés à partir des images de la base $\mathcal{L}_a$, seule la base $\mathcal{L}_b$ est utilisée dans la suite des expériences : sur un petit nombre d'images, les disparités se font d'autant plus ressentir. L'estimation des performances se déroule donc comme suit :

- pour chaque classe, prendre α% des éléments pour constituer une base d'apprentissage,
- tester sur les N_{test} éléments restants et estimer le taux de classification a,

en répétant ces étapes 10 fois. A chaque étape, le taux de classification est estimé directement par $a = \sum_{i=1}^{N_{test}} [\![y_i = f(x_i)]\!]$. Nous reportons le taux de classification moyen sur les 10 tests.

3.4.2 Expériences

données

La base d'images de voitures contient 20 modèles, avec une dizaine de détails sélectionnés par modèle, pour un total de $d = 197$ détails à détecter. Chaque détecteur δ_i est associé à un descripteur pouvant être extrait à différentes échelles $\sigma \in \Sigma_i$, et sur une zone limitée de l'image, déterminées à partir de la base $\mathbf{\Pi}_i$.

zone de recherche

Pour une classe de patchs donnée, la zone de recherche est définie par rapport aux positions des patchs annotés. Plus précisément, pour chaque base de patchs $\mathbf{\Pi}_i$, on récupère les positions extrêmes $x^i_{min}, y^i_{min}, x^i_{max}, y^i_{max}$. La zone de recherche minimale est un rectangle de dimension $w_i \times h_i$ avec $w_i = x^i_{max} - x^i_{min} + 1$ et $h_i = y^i_{max} - y^i_{min} + 1$. Ce rectangle est centré en (x^i_0, y^i_0), avec $x^i_0 = x^i_{min} + w_i/2$ et $y^i_0 = y^i_{min} + h_i/2$. Il est agrandi d'un facteur $\alpha = 20\%$ pour donner la zone de recherche finale.

échelles

Les échelles sont ajustées de manière à réduire globalement le nombre d'échelles auxquelles les descripteurs sont calculés (sachant qu'il y a des recoupements entre les différentes classes de patchs). Pour cela, nous adoptons une procédure en deux temps. Premièrement, on extrait N_s échelles décrivant les variations de taille à l'intérieur de chaque base de patchs, et on les regroupe dans un même ensemble L. Deuxièmement, on sous-échantillonne L_s de manière à réduire sa taille en dessous d'un seuil fixé (on obtient Σ_i) . Une deuxième méthode de sélection des échelles peut se faire conjointement avec la sélection des paramètres n_w et n_h des descripteurs SIFT. Le produit $n_w \times n_h$ étant gardé inférieur à un seuil donné (typiquement égal à 19 ou 20), le découpage optimal d'un patch de taille (w, h) à une échelle donnée σ est obtenu en minimisant le nombre de pixels résiduels sur les patchs pour chaque base d'apprentissage $\mathbf{\Pi}_i$.

Nous comparons une méthode où $n_h = n_w = 4$ sont fixés pour tous les détecteurs avec une méthode où n_h et n_w sont estimés séparément pour chaque catégorie de patchs. Dans les deux cas, nous limitons le nombre d'échelles par patchs à $n_\sigma = 5$, et le nombre d'échelles total à $4n_\sigma$.

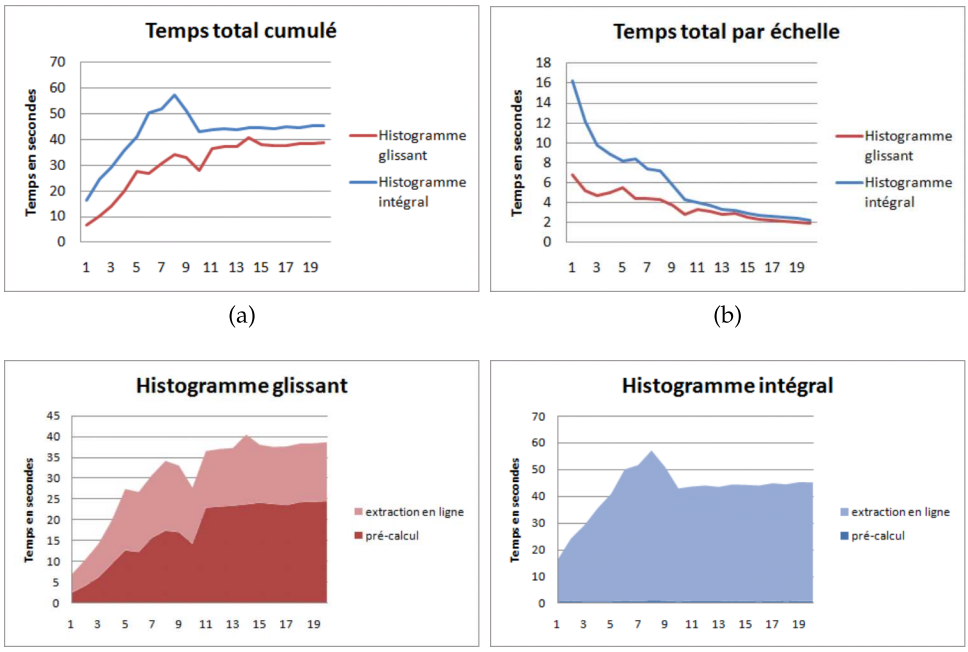


FIGURE 3.23 – *Évolution du temps de calcul des histogrammes en fonction du nombre d'échelles. Le calcul se déroule en 2 phases : pré-calcul, commun à tous les détecteurs, des histogrammes unitaires ou des images intégrales suivant les cas, et calcul en ligne, donnant l'histogramme final. Les figures (c) et (d) donnent la décomposition du temps total de la figure (a) entre pré-calcul et calcul en ligne. On constate que le temps de pré-calcul des histogrammes glissants tend à devenir un handicap lorsque le nombre d'échelle augmente. L'apparition d'un palier s'explique par le fait que nos données ne s'étendent pas assez en échelle. Le temps de calcul moyen par échelle diminue plus rapidement avec l'histogramme intégral (figure (b)).*

Extraction efficace des histogrammes

Nous comparons également les méthodes d'extraction rapide des histogrammes en terme de temps de calcul, par rapport au nombre d'histogrammes à classifier. Pour cela, nous calculons une signature sur la même image, avec les deux méthodes, et avec des détecteurs entraînés avec des valeurs de n_σ variant entre 1 et 20, et $n_h = n_w = 4$ fixés, de manière à ce qu'aucune variations de découpage ne perturbent les calculs. Les variations de temps de calcul sont donnés figure 3.23. Il est intéressant de constater que les variations ne sont pas uniformes. Cela peut s'expliquer par le fait qu'on cherche à appliquer des détecteurs avec une grande variation dans les échelles, alors que les données d'apprentissage présentent des variations limitées : au bout d'un certain temps, le nombre optimal d'échelles est atteint, et bien que le nombre d'échelles théoriquement possible augmente, le nombre utilisé en pratique se stabilise (aux alentours de $n_\sigma = 10$). Pour mieux évaluer la différence entre les deux calculs d'histogrammes, il faudrait donc une base d'images plus variées. Globalement, on constate que le temps de calcul investi au départ avec les histogrammes glissants est largement compensé par le temps gagné au moment de l'extraction en ligne des histogrammes. Ceci est dû au fait que l'on extrait un très grand nombre de fenêtres de l'image pour chaque détecteur. Le nombre d'échelle augmentant, il semble que les pré-calculs deviennent malgré tout un handicap pour la méthode par histogrammes glissants, ce qui serait à confirmer avec d'autres données. Par contre, il apparaît clairement que l'histogramme intégral est plus intéressant pour calculer quelques détecteurs seulement.

sélection des paramètres SVM

Les paramètres définissant le classifieur SVM one-class pour chacun des dé-

tecteurs sont sélectionnés par validation croisée, en respectant la partition de la base utilisée dans la phase d'extension (cf. figure 3.22). Pour classifier les descripteurs SIFT, nous utilisons le noyau intersection d'histogramme : le seul paramètre à sélectionner est donc le paramètre ν des SVM. Lorsque nous utilisons d'autres descripteurs, nous utilisons un noyau RBF : un deuxième paramètre vient donc s'ajouter, définissant la taille du noyau.

On peut avoir une première estimation de la qualité des détecteurs en observant l'erreur de validation croisée des classifieurs correspondants. Ceci permet de faire une première sélection des paramètres, avant d'effectuer les tests complets (courbes précision/rappel, "PR"), coûteux en temps de calcul.

courbes précision/rappel

La figure 3.25 montre l'ensemble des courbes PR obtenues, ainsi que la courbe moyenne sur tous les détecteurs (en vert). Globalement, elle permet de constater qu'il y a une grande disparité dans les performances des différents détecteurs : quelques uns réussissent très bien, tandis qu'une bonne partie présentent des résultats très mauvais. La figure 3.26 et le tableau 3.5 donnent la précision moyenne (*average precision* AP) de chacun des détecteurs et permet de mieux visualiser les performances de chaque détecteur : moins de 10% des détecteurs ont une précision moyenne supérieure à 50%.

Une faible précision est souvent observée, mais elle peut s'expliquer assez facilement, car les fausses alarmes observées ne sont pas toujours des aberrations : il peut y avoir des détections justes en pratiques (il est normal qu'un détecteur de phare de 206-3 portes trouve les phares de 206-5 portes), ou du moins parfaitement compréhensibles (confusions entre différentes poignées de portes, ou différentes roues). La figure 3.27 montre la courbe PR moyenne par type de détail, et permet de renforcer cette intuition : les détails les plus confondus, entre autres, sont les roues et le clignotant latéral. La figure 3.29 en donne la confirmation : nous affichons, pour quelques catégories de patchs, les 3 premières fausses alarmes (selon le score SVM). Dans la plupart des cas, les détecteurs se trompent sur des détails de même type que ceux recherchés. Nous pouvons quantifier ce phénomène en dénombrant deux types de fausses alarmes. Les fausses alarmes seront dites *de type I*, si leur localisation correspond à un détail du même type que le détail recherché. Sinon, elles seront dites *de type II*. Nous calculons ainsi le rapport r^i du nombre de fausses alarmes de type I par rapport au total (types I et II) pour un détecteur δ_i (en utilisant l'équation 3.18, avec un seuil de 0,5, pour tester la différence de localisation entre le détail trouvé et le détail annoté). Nous calculons la moyenne de ce rapport $\bar{r}^i$ pour différentes valeurs de seuillage (parrallèlement à la courbe PR). Selon les détecteurs, la valeur de $\bar{r}^i$ varie beaucoup. En moyenne, on a $\bar{r} = 49.4\%$ des fausses alarmes qui sont de type I.

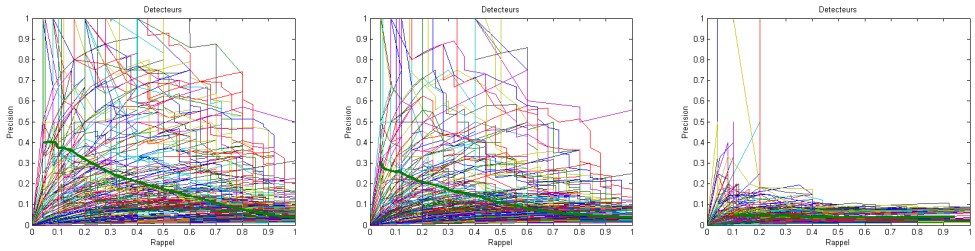
étude des fausses alarmes

La figure 3.24 affiche les détections obtenues pour tous les détecteurs dont le score est supérieur à 0.05. On constate que les signatures sont effectivement bruitées par de nombreuses fausses alarmes. A priori, un petit nombre seulement des détections semblent pouvoir être utiles. Il sera intéressant de vérifier si cette intuition se confirme par la suite, i.e. quelles sont les détections aidant le mieux l'annotation.

Mise en relation avec le nombre de données d'apprentissage

La figure 3.28 donne la courbe PR moyenne par modèle. Étant donné les disparités de répartition des images entre les modèles, il est intéressant de mettre en relation les performances des détecteurs par rapport au nombre d'exemples d'apprentissage : on constate en effet que les modèles ayant les moins bonnes performances sont appris avec moins de 10 exemples. Avec quelques exceptions (par exemple, la Corsa), les détecteurs les plus efficaces correspondent au modèles les mieux représentés. Ainsi, toutes les performances que nous montrerons plus loin

FIGURE 3.24 – *Exemples de détections. Les boîtes englobantes correspondantes sont affichées lorsque le score est supérieur à 0.05. L'affichage est muet pour éviter la surcharge. On constate qu'il y a une forte proportion de fausses alarmes, mais presque toujours plusieurs bonnes détections. Dans ces conditions, le score de détection peut s'avérer un renseignement essentiel pour éliminer des fausses alarmes.*



(a) (b) (c)

FIGURE 3.25 – *Courbes précision/rappel obtenues pour tous les détecteurs. (a) : Descripteurs SIFT et noyau intersection, (b) : Descripteurs SIFT et noyau RBF, (c) : Niveaux de gris. En vert et trait épais, la courbe moyenne.*

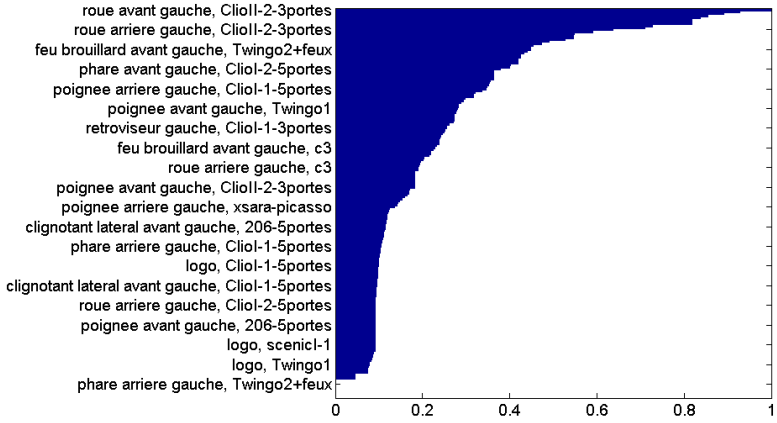


FIGURE 3.26 – *Précision moyenne (AP) des détecteurs. Pour plus de clarté, seules certaines valeurs sont étiquetées en ordonnée. Les résultats complets sont données dans le tableau 3.5.*

seront à considérer avec l'idée qu'en ajoutant simplement quelques images sur certains modèles, nous pourrions faire encore mieux.

Pour les SIFT, le noyau intersection d'histogramme donne de meilleures performances que le noyau RBF, comme on pouvait s'y attendre. Le descripteur niveau de gris, utilisé ici avec un noyau RBF, donne des résultats très décevants et ne sera donc pas exploité.

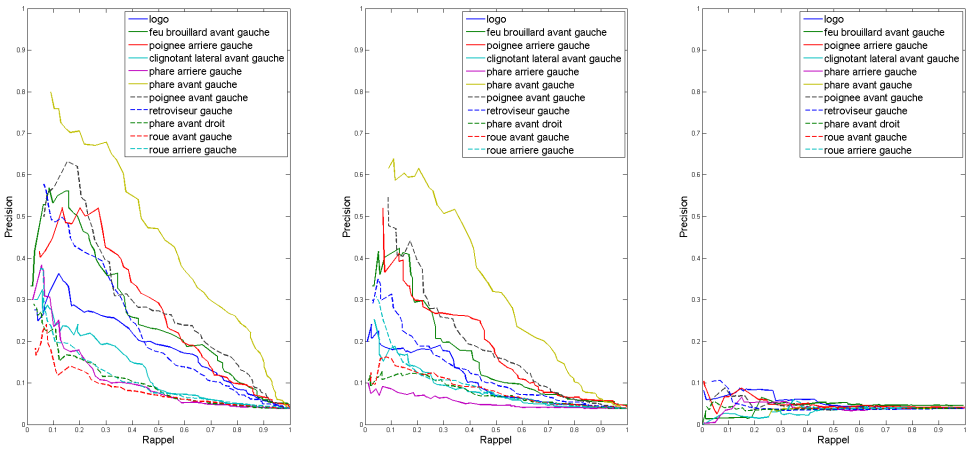
Enfin, nous comparons les performances des signatures obtenues par ces détecteurs de détails par rapport aux signatures obtenues par SPM. Nous utilisons un classifieur SVM un-contre-tous avec un noyau intersection d'histogramme pour les SPM, et un simple noyau linéaire pour les détecteurs. Les taux de classification obtenus dans chaque cas sont rapportés dans le tableau 3.6. Pour des vecteurs de dimension nettement inférieure, et malgré le bruit des détecteurs (erreurs de détection) on observe des performances nettement meilleures, ce qui démontre la pertinence de notre approche. Nous comparons également les différentes variantes de signatures.

3.4.3 Discussion

Les résultats obtenus montrent que l'utilisation de détecteurs de détails permet d'avoir de bien meilleures performances que les caractéristiques classiques. Nous prévoyons d'ailleurs que ces performances peuvent encore progresser largement avec la qualité des détecteurs. En effet, nombre de nos détecteurs ont des performances individuelles assez mauvaises. Quelques pistes possibles sont :

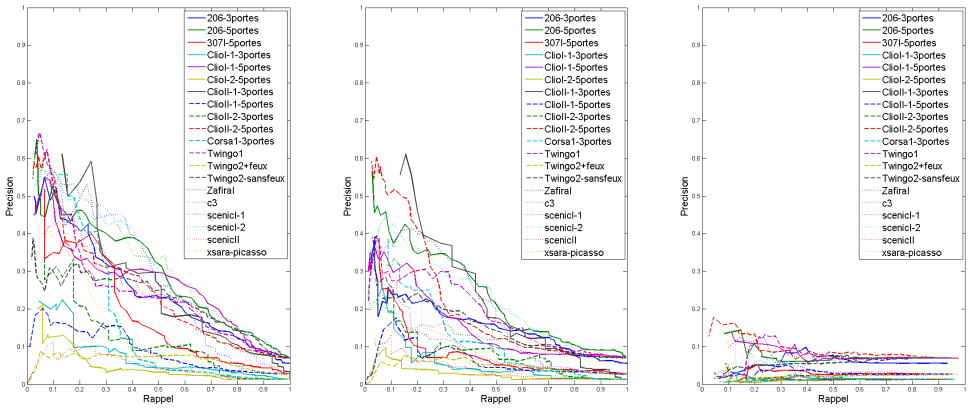
	logo	feu brouillard avant gauche	poignee arriere gauche	clignotant lateral avant gauche	phare arriere gauche	phare avant gauche	poignee avant gauche	retroviseur gauche	phare avant droit	roue avant gauche	roue arriere gauche
206 3 portes	31,66	-	-	4,55	7,42	35,64	10,83	12,45	7,87	36,36	34,69
206 5 portes	23,89	-	43,19	11,45	11,39	**44,83**	9,09	11,25	10,10	18,90	18,18
307I-5portes	9,60	-	9,51	10,00	9,33	**54,82**	11,63	10,24	-	**49,09**	28,18
Clio-I-1 3 portes	9,09	-	-	0,00	4,55	**45,45**	40,26	25,45	9,09	**47,27**	18,18
Clio-I-1 5 portes	9,83	-	34,59	9,48	10,47	39,95	19,36	23,85	10,40	20,36	18,18
Clio-I-2 5 portes	9,09	-	0,00	9,36	0,00	37,88	42,42	41,82	9,09	**81,82**	9,09
Clio-II-1 3 portes	10,73	-	-	**81,82**	29,79	**89,09**	27,27	21,82	36,36	**70,91**	41,82
Clio-II-1 5 portes	9,09	-	9,71	9,36	9,09	41,82	23,74	16,78	10,27	21,82	24,55
Clio-II-2 3 portes	33,64	-	-	0,00	9,09	36,36	18,18	15,91	11,65	**100,00**	**63,64**
Clio-II-2 5 portes	11,02	27,56	9,09	8,60	8,72	26,14	17,15	12,30	21,43	31,79	29,35
Corsa-1 3 portes	12,10	-	-	16,88	14,29	**92,73**	9,09	43,94	9,09	18,18	36,36
Twingo-1	7,79	-	-	8,17	4,59	15,24	28,06	23,90	9,76	36,36	27,92
Twingo-2 avec feux	9,22	44,73	-	0,00	0,00	9,24	0,00	9,85	9,09	9,09	11,82
Twingo-2 sans feux	11,89	-	-	9,30	9,09	25,55	23,14	35,17	8,36	18,18	19,64
Zafira-I	7,46	14,71	9,82	7,68	10,65	**83,64**	10,50	10,15	9,59	27,27	**59,09**
c3	9,71	22,74	14,10	9,77	22,39	35,31	9,14	19,14	7,47	32,00	18,91
scenic-I-1	9,09	-	0,00	9,16	9,09	**85,45**	**54,55**	10,00	10,01	9,09	18,18
scenic-I-2	11,48	25,00	42,51	8,53	9,28	**52,73**	0,00	18,19	27,02	25,09	24,15
scenic-II	9,09	-	11,94	9,69	20,05	**72,73**	28,16	11,01	16,46	18,18	27,27
xsara picasso	9,09	9,09	13,64	9,09	11,02	28,90	9,09	**54,55**	9,09	27,27	**81,82**

TABLE 3.5 – *Précision moyenne des détecteurs (AP, en %). Les détecteurs les plus performants sont mis en évidence.*



(a) (b) (c)

FIGURE 3.27 – *Courbes précision/rappel moyenne par type de détail. (a) : Descripteurs SIFT et noyau intersection, (b) : Descripteurs SIFT et noyau RBF, (c) : Niveaux de gris*



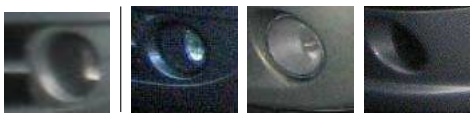
(a) (b) (c)

FIGURE 3.28 – *Courbes précision/rappel moyenne par modèle. (a) : Descripteurs SIFT et noyau intersection, (b) : Descripteurs SIFT et noyau RBF, (c) : Niveaux de gris*

Méthode	d	$a(\%)$
SPM	4200	$31,4 \pm 3,1$
Détails directs	197	$50,13 \pm 4,76$
Détails binarisés	197	$37,94 \pm 5,52$
Détails normalisés	197	$\mathbf{51,05 \pm 3,68}$

TABLE 3.6 – *Comparaison des performances sur la base Lb des détecteurs de détails par rapport aux SPM. d est la dimension de signatures utilisées pour la classification. a est le taux de classification. Malgré des détecteurs fortement bruités, nous observons un net gain de performances.*

(a) *Logo - 206, 3 portes*

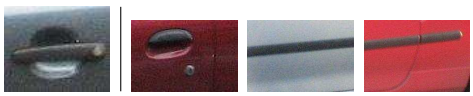


(b) *Feu de brouillard - Clio II-2, 5 portes*

(c) *Phare arrière gauche - Zafira I*

(d) *Logo - Twingo 1*

(e) *Clignotant latéral - Twingo 2, sans feux*



(f) *Poignée avant gauche - C3*

FIGURE 3.29 – *Exemples de fausses alarmes. Pour chaque type de patch, nous donnons d'abord un exemple de positif, suivi des 3 fausses alarmes renvoyées par l'algorithme avec les plus forts scores. On constate que les erreurs les plus fréquentes ne sont pas aberrantes : dans la majorité des cas, c'est le même type de détail qui est trouvé, sur un autre modèle. Dans le dernier cas, la confusion peut s'expliquer par la similarité entre la forme de la poignée et de la baguette latérale. La confusion entre les logos de Renault et de Peugeot paraît plus étonnante. Elle peut s'expliquer par la présence dans les deux cas du pare-choc et par une similarité de couleurs du logo, ainsi que par la difficulté intrinsèque de reconnaître les logos Renault, qui présentent une bi-modalité (selon la couleur de la carrosserie) et une forme simple.*

- Introduction de détecteurs supplémentaires, associés à différents niveaux de précisions. Ceci permettrait d'avoir des détecteurs plus robustes dans certains cas (par exemple en mélangeant phares de 206-3 portes et de 206-5 portes, identiques). Cependant, il n'est pas évident que tous les détecteurs ainsi introduits soient meilleurs : un détecteur de "poignées de Peugeot", par exemple, serait multimodal et probablement peu robuste.
- Introduction de critères géométriques. En utilisant comme amers les patchs associés aux détecteurs les plus robustes, on pourrait contraindre les autres détections.
- Introduction de nouvelles caractéristiques. Les méthodes à noyaux permettent de concaténer différents descripteurs (cf. par exemple Nilsback et Zisserman [136]).
- Apprentissage plus robuste, par exemple par l'utilisation d'une cascade de détecteurs : les fausses alarmes obtenues par les classifieurs actuels peuvent servir d'exemples négatifs pour apprendre un classifieur binaire, permettant une meilleure sélectivité de l'algorithme.

Nous n'avons pas plus approfondi ce problème ici, car il s'écarte de notre objectif initial, qui est avant tout de proposer une méthode d'annotation multifacette hiérarchique. Cependant, la définition de signatures plus robutes sera essentielle pour augmenter les performances.

Un désavantage qui paraît important est la part de supervision, notamment la nécessité d'annoter précisément une grande quantité d'images. Cependant, nous avons montré que même avec un petit nombre d'images annotées par classe (une vingtaine), on peut déjà avoir des résultats intéressants. D'une part, l'investissement dans le travail humain paye, et d'autre part, le développement d'une interface adaptée permet de réduire cet investissement. Si l'automatisation semble attrayante, elle ne permet pas toujours les mêmes performances. Une méthode intermédiaire pourrait être d'exploiter les premières annotations pour faire des propositions d'annotations sur les nouvelles images, introduisant ainsi une méthode d'apprentissage en ligne, ou mieux, d'apprentissage actif [2, 144].

Un autre reproche que l'on pourrait faire vis-à-vis de notre méthode est la difficulté qu'il y aurait à l'étendre si la base intégrait de nouvelles catégories. Certes, en l'état, les classifieurs basés sur ces signatures devraient être entièrement ré-entraînés. En contrepartie, l'utilisation de classifieurs SVM one-class pour les détecteurs permet de ne pas avoir à tout recalculer : seuls les détecteurs associés aux nouvelles catégories sont à entraîner, ce qui serait de toute façon inévitable. Sans même ajouter de nouveaux modèles de voitures, les signatures que nous proposons sont facilement extensibles : pour ajouter un nouveau détecteur, il suffit d'ajouter une base de patchs à $\mathbf{\Pi}$, l'entraînement est entièrement indépendant. Au final, ce type de signature se révèle plutôt souple. Les difficultés liées à l'ajout de catégorie se rencontreront plutôt au niveau de la catégorisation des modèles. Nous en rediscuterons plus loin.

Dans ce chapitre, nous avons introduit une base de donnée et les descripteurs associés. Nous allons maintenant les exploiter pour développer une méthode d'annotation multifacette hiérarchique, objet du chapitre 4.

4 Annotation multifacette hiérarchique d'images

L'objectif de cette thèse est de développer une méthode d'annotation multifacette hiérarchique des images. Dans le chapitre 2, nous avons justifié cet objectif par deux raisons essentiellement : (a) en montrant que c'était le type de description le plus adapté aux utilisateurs, et (b) parce que c'est une approche qui permet une plus grande souplesse que les approches proposées par l'état de l'art. Dans le chapitre 3, nous avons introduit une base d'image annotée de manière suffisamment précise et suffisamment variée pour étudier les aspects multifacette et hiérarchique. Dans le but d'améliorer les performances de référence, nous avons vu comment extraire des caractéristiques visuelles informatives de l'image. Nous avons testé ces caractéristiques pour la catégorisation, au niveau de précision sémantique le plus spécifique.

Dans ce chapitre, nous arrivons au cœur de notre étude : nous allons maintenant montrer comment il est possible d'annoter les images de manière souple et structurée, en intégrant en particulier ces deux aspects de multifacette et hiérarchique. L'idée est de construire une annotation sous la forme d'une distribution de multilabels associés à des niveaux de confiance. Dans ce but, nous verrons comment adapter le problème multilabel classique pour intégrer les contraintes liées à la structure du vocabulaire.

Nous commençons par donner une formulation précise du problème (section 4.1). Nous reprenons ensuite l'état de l'art sur les méthodes d'annotation hiérarchiques (section 4.2), en insistant davantage sur les aspects techniques qu'au chapitre 2. Nous pouvons alors introduire la démarche choisie, que nous décrivons pas-à-pas (4.3). Nous introduisons une méthode d'évaluation des performances adaptée à notre problématique (4.4), que nous utilisons dans nos expériences (4.5).

4.1 Introduction et formalisation du problème

4.1.1 Objectif

Dans le chapitre 2, nous avons analysé le problème de l'annotation d'images, et nous avons fait ressortir entre autres trois aspects :

1. une image peut être annotée à plusieurs niveaux, et selon différents points de vue,
2. différents utilisateurs pourront user de différentes interprétations pour une même image,
3. pour un même utilisateur, l'interprétation qu'il fait d'une image peut évoluer au cours du temps.

Multilabel	Confiance
Berline	0.78
Citadine	0.66
Berline, Citadine	0.6
Citroën	0.55
Berline, Citroën	0.5
Berline, Citadine, Citroën	0.45
Berline, Citadine, Citroën, C3	0.4
Berline, Citadine, Citroën, C3, 5 portes	0.2

FIGURE 4.1 – *Exemple d'annotation multifacette hiérarchique, sous la forme d'une distribution de multilabels, associés à des indices de confiance. Un multilabel peut contenir différents types de descriptions (aspect multifacette). Les différents multilabels présentent des relations hiérarchiques cohérentes. Les indices de confiance associés diminuent avec la précision sémantique des multilabels.*

Pour constuire des métadonnées plus proches de l'utilisateur, il est donc naturel d'adapter la forme des annotations d'une image, de manière à offrir une multiplicité d'interprétations. Pourtant, dans la littérature, la manière d'annoter la plus courante consiste à associer *un* label à *une* image. La notion de **multilabel** a certes été introduite, lorsque plusieurs labels sont associés à une image. Mais souvent, ces labels correspondent en réalité à différents objets contenus dans l'image (par ex. voiture, chaussée, maison...). Nous nous intéressons ici à une notion plus précise de multilabel : notre objectif est d'associer *plusieurs labels* à *un seul objet* (par ex. Citroën, citadine,...). Nous considérons donc qu'une image contient un seul objet. Par ailleurs, notre but est aussi de donner des multilabels respectant la structure du vocabulaire naturelle pour l'utilisateur.

Le type d'annotation que nous proposons d'associer à une image prend la forme d'une distribution de multilabels, associés à des indices de confiance. La figure 4.1 montre le type de résultat attendu pour une C3, 5 portes. Différentes caractérisations sont possibles : soit très précises (par ex. C3), soit génériques (par ex. citadine). Les multilabels proposés recouvrent différents types de descriptions (facettes), et différents niveaux de précision sémantique (hiérarchie). Selon ce qu'il cherche, l'utilisateur pourra jouer sur un seuil de confiance pour sélectionner différentes annotations.

4.1.2 Formalisation

L'annotation que nous proposons d'associer à l'image est caractérisée par la notion de **multilabel**, c'est-à-dire d'ensemble de labels. Un **label** $l \in \mathcal{L}$ est un mot du vocabulaire, où $\mathcal{L} = \{l_1, \ldots, l_{N_l}\}$, i.e. il y a $|\mathcal{L}| = N_l$ labels possibles. Un **multilabel** M est un sous-ensemble de $\mathcal{L}$. Suivant la notation habituelle ([175, 183, 11]), on note un multilabel sous forme de vecteur binaire $\mathbf{y} \in \{0,1\}^{N_l}$ où $y_j = 1$ si et seulement si $l_j \in M$.

On suppose de plus que le vocabulaire est structuré, c'est-à-dire qu'il est associé à un ensemble de contraintes, définissant :

1. des relations de hiérarchie : $l_1 \prec l_2$ si l_1 est un (Is-A) l_2,
2. des relations d'exclusion, sous forme de paires l_1, l_2 telles que les deux labels ne peuvent être attribués simultanément.

Un multilabel $\mathbf{y}$ est dit **consistant** par rapport à des contraintes de hiérarchie et d'exclusion s'il respecte ces contraintes.

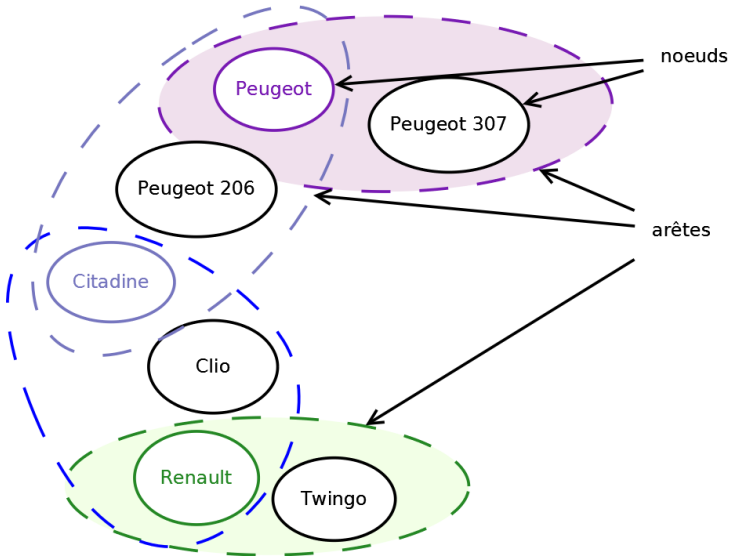


FIGURE 4.2 – *Modélisation des multilabels sous la forme d'un hypergraphe. Les arêtes correspondent à des labels. Les multilabels sont des ensembles de labels plus ou moins imbriqués.*

En combinatoire, la structure permettant de modéliser différentes combinaisons d'éléments est l'hypergraphe [20]. Nous proposons d'utiliser une telle structure pour décrire les multilabels consistants.

Dans cet hypergraphe, noté $\mathcal{H}$, l'ensemble des sommets est $\mathcal{L}$ et l'ensemble des arêtes $\mathcal{M}$ est formé par tous les multilabels consistants $\mathcal{M} = \{M_1, \ldots, M_{N_m}\}$, avec $|\mathcal{M}| = N_m$. La figure 4.2 montre un exemple avec quelques labels décrivant des voitures. Occasionnellement, on notera aussi un multilabel sous forme d'ensemble $M = \{l_{i_1}, \ldots, l_{i_{|M_j|}}\}$, lorsque cette notation sera plus naturelle.

Par abus de notation, on notera $|\mathbf{y}| = |M| = \sum_{i=1}^{N_l} y_i$. L'ensemble des multilabels sous forme binaire est noté $\mathcal{Y}$. Il est isomorphe à $\mathcal{M}$. Nous utiliserons de préférence la notation binaire.

On définit la **complexité** d'un multilabel $\mathcal{C}(\mathbf{y}) = |\mathbf{y}|$. La complexité permet de traduire la notion de **précision sémantique** d'un multilabel : plus un multilabel est complexe, i.e. plus il contient de labels, plus il est spécifique.

Les relations de hiérarchie entre les labels se traduisent dans $\mathcal{H}$ par la relation $\prec$, telle que $\mathbf{y}_i \prec \mathbf{y}_j$ si $M_j \subseteq M_i$. En munissant l'hypergraphe de cette relation d'ordre partiel entre arêtes, on peut le généraliser à un *poset* (ensemble partiellement ordonné), et le représenter par un diagramme de Hasse [20], dont les nœuds sont les arêtes de $\mathcal{H}$ (le nœud j correspond au multilabel $\mathbf{y}_j$), et où une arête entre deux nœuds, de i vers j existe si :

1. $\mathbf{y}_j \prec \mathbf{y}_i$,
2. $\mathcal{C}(\mathbf{y}_j) = C(\mathbf{y}_i) + 1$.

Nous identifierons l'hypergraphe et sa représentation sous forme de diagramme de Hasse, en notant encore $\mathcal{H}$ le graphe obtenu. Nous travaillerons sur les relations de $\mathcal{H}$ en utilisant les notations $fils(i)$, $par(i)$, $desc(i)$, $anc(i)$, pour désigner respectivement les ensembles de nœuds fils, parents, descendants et ancêtres de i.

Soit un label $l \in \mathcal{L}$. On notera $\mathbf{y}(l)$, où $M(l)$, le **multilabel minimal** M contenant l, c'est-à-dire :

$$\exists! M \in \mathcal{H} \text{ tel que } l \in M \text{ et } \forall M' \in \mathcal{H},\ M' \subseteq M \Rightarrow l \notin M. \tag{4.1}$$

À tout label de $\mathcal{L}$, on peut associer un multilabel minimal dans $\mathcal{H}$. Le multilabel "vérité terrain" d'une image $\mathcal{I}_i$ est noté $\mathbf{t}_i = \mathbf{y}(k_i)$.

Les catégories *de base*, ou *atomiques*, correspondant par exemple aux catégories définies par les modèles de voitures au chapitre 3, correspondent aux feuilles de l'hypergraphe : ce sont les éléments maximaux. Soit $\mathbf{y} \in \mathcal{H}$, $\mathbf{y}$ est une feuille si et seulement si $\forall \mathbf{y}' \in \mathcal{H},\ \mathbf{y}' \prec \mathbf{y} \Rightarrow \mathbf{y}' = \mathbf{y}$. On note $\hat{\mathcal{H}}$ l'ensemble des feuilles de $\mathcal{H}$.

Dans la suite, nous travaillerons essentiellement avec les multilabels en tant que nœuds de $\mathcal{H}$.

Le but est de proposer une interprétation multi-niveaux : contrairement au chapitre 3, les catégories en sortie sont tous les nœuds du graphe $\mathcal{H}$, et ne sont pas limitées aux feuilles. Notre objectif est d'évaluer la pertinence de l'association d'un multilabel à une image, de manière à pouvoir gérer un compromis entre fiabilité et précision : intuitivement, plus un multilabel est complexe, moins l'annotation sera fiable. Dans certaines applications, il peut être intéressant de réduire l'information apportée (moins de labels), pour donner une information plus sûre.

Nous avons introduit au chapitre 2 la notion de vocabulaire contrôlé. L'idée est d'utiliser la structure du vocabulaire pour en déduire les multilabels consistants et l'hypergraphe $\mathcal{H}$. Nous allons donc adjoindre à la base d'images introduite au chapitre 3, une structure de graphe Is-A, décrivant ces relations entre labels.

Le système d'annotation que nous allons présenter utilise les données suivantes :

1. une base d'images $\mathcal{I}_1, \ldots, \mathcal{I}_n$, n étant la taille de la base, contenant des objets de K classes distinctes. Chaque image contient un et un seul objet. Les images sont décrites par des vecteurs caractéristiques $x_1, \ldots, x_n$ de dimension d, également appelés signatures.
2. des contraintes de hiérarchie et d'exclusion entre les labels. Dans notre cas, ces contraintes seront définies par un graphe Is-A de représentation des connaissances, noté $\mathcal{G}$, contenant N_l nœuds ou labels, que nous allons décrire un peu plus loin. Ce graphe a K feuilles, correspondant aux K catégories, et permet de construire l'hypegraphe $\mathcal{H}$ qui servira à l'annotation,
3. la vérité-terrain, sous forme d'annotations des objets : à une image $\mathcal{I}$ correspond une annotation $k \in [K]$, décrivant l'objet qu'elle contient. Ce label correspond à un multilabel $\mathbf{t} = \mathbf{y}(k) \in \mathcal{H}$.

Le but est de déterminer, pour une image test, quels sont les labels qui la caractérisent le mieux, à différents niveaux de précision, et suivant différentes facettes.

4.2 Remarques bibliographiques

À notre connaissance, l'apprentissage avec des multilabels structurés par un hypergraphe n'a pas encore fait l'objet d'études. Cependant, un certain nombre de travaux en apprentissage statistique utilisent des sorties avec des structures arborescentes ou, plus généralement, des graphes, et se rapprochent de notre problème. Comme nous l'avons montré au chapitre 2, l'exploitation de vocabulaires structurés pour la reconnaissance est un sujet suscitant un intérêt croissant.

4.2.1 Exploitation de hiérarchies

Les hiérarchies sémantiques sont couramment utilisées pour l'annotation de documents textuels. Ainsi, la plupart des travaux traitant de classification hiérarchique sont appliqués à ce domaine, où la définition d'un vocabulaire est plus naturelle, puisque les mots eux-mêmes sont les données. On peut distinguer principalement deux types d'approches de la classification hiérarchique : les approches

locales, où les nœuds sont traités de manière individuelle, dans un premier temps au moins, et les approches globales.

4.2.1.1 Approches locales

classification multilabel hiérarchique de documents,...

Cesa-Bianchi et al. [35] proposent un algorithme de classification hiérarchique où la structure est une forêt $\mathcal{G}$, et où les multilabels sont associés à des chemins multiples ou partiels. Un chemin relie un nœud quelconque à une racine. Les multilabels autorisés (i.e. respectant $\mathcal{G}$) sont des unions de chemins. L'algorithme proposé est incrémental : les vecteurs d'apprentissage sont présentés un à un, et permettent d'ajuster les paramètres itérativement. La classification se fait de manière descendante : partant de chaque racine, si le nœud est évalué positivement, on peut évaluer ses fils, sinon tous sont considérés négativement. Par conséquent, le classifieur de chaque nœud est appris uniquement sur les instances positives pour ce nœud. Les classifieurs appris itérativement sont de simples seuils linéaires, appris par une variante hiérarchique de la régression aux moindres carrés. Les mêmes auteurs, dans Cesa-Bianchi et al. [34], sur le même type de données, utilisent des classifieurs SVM en chaque nœud, toujours de manière descendante. Ils montrent, de plus, que cette classification peut être améliorée en estimant les probabilités a posteriori en chaque nœud (en transformant les sorties des SVM selon l'algorithme de Platt [146]), et en les combinant de manière ascendante. Cette procédure permet de s'assurer que tous les multilabels obtenus en sortie sont consistants. Un des inconvénients majeurs de ce type de méthode est qu'elle ne peut pas se généraliser facilement à des graphes non arborescents : en particulier, la combinaison des probabilités repose sur le fait que la probabilité d'un nœud parent est la somme des probabilités de ses nœuds fils. Lorsqu'un nœud peut avoir plusieurs parents, cette propriété n'est pas toujours vérifiée.

...de gènes

Pour le même type de structure, mais pour une application de classification de gènes, Barutcuoglu et al. [19] proposent une méthode plus souple, en repoussant à la décision finale la contrainte d'avoir des multilabels consistants en sortie. Des classifieurs SVM sont appliqués de manière indépendante sur tous les nœuds de la hiérarchie. Les probabilités sont estimées indépendamment sur tous les nœuds de la hiérarchie. Le lien avec la hiérarchie est introduit en utilisant un modèle graphique pour combiner les scores de classification des SVM : les probabilités sont calculées en estimant la distribution des scores sur chaque nœud.

4.2.1.2 Approches globales

approches théoriques de la classification avec des sorties structurées

D'autres approches de la classification avec des sorties structurées, basées sur des modèles discriminatifs, sont proposées entre autres par Taskar et al. [175] et Cai et Hofmann [31]. L'une comme l'autre sont des généralisations des SVM multiclasse. Tsochantaridis et al. [183] présentent les deux approches selon un formalisme commun. L'avantage de ces approches est qu'elles gèrent directement les liens hiérarchiques au cours de l'apprentissage, et qu'elles permettent d'intégrer une fonction de coût dépendante de la structure. Il existe encore d'autres approches, sur lesquelles nous ne nous étendrons pas ici. Pour une étude plus approfondie de la théorie sur la prédiction de données structurées, nous référons le lecteur à Bakir et al. [11].

Nous nous sommes plus particulièrement intéressés à la façon dont les vocabulaires hiérarchiques pouvaient être utilisées pour l'annotation d'images ou la reconnaissance d'objets.

4.2.2 Exploitation de hiérarchies en vision

Nous avons vu dans le chapitre 2 comment l'introduction de hiérarchies sémantiques se faisait naturellement pour l'analyse d'images. Nous avons également remarqué que les vocabulaires structurés étaient essentiellement utilisés pour enrichir le vocabulaire, et éviter certaines ambiguïtés : peu d'articles abordent réellement le problème de l'annotation multilabel hiérarchique. Nous examinons ici plus précisément les approches existantes, que nous avons déjà évoquées dans la section 2.3.5.

Remarquons que nous nous intéressons d'abord aux relations de type Is-A, c'est-à-dire d'hypernymie/hyponymie. Des travaux comme ceux de Gao et Fan [74] intègrent les deux relations Is-A et Is-A-Part-Of. Ils utilisent donc deux hiérarchies : l'une représente les liens de méronymie/holonymie entre les objets d'intérêt (*salient object taxonomy*), l'autre représente les liens d'hypernymie/hyponymie entre concepts (*concept ontology*). Chacune est construite de manière semi-automatique, à partir de WordNet et des annotations de la base d'images utilisée (LabelMe).

Ils utilisent une approche Bayésienne hiérarchique pour estimer des probabilités sur chacun des concepts de manière ascendante (*bottom-up*). La probabilité de chaque concept au niveau atomique (c'est-à-dire des feuilles) est d'abord estimée en fonction des objets d'intérêt. En notant $O_1, \ldots, O_{N_o}$ les variables aléatoires binaires indiquant la présence de chaque objet, et $\mathcal{S}$ la variable aléatoire discrète dénotant un concept atomique, à valeurs dans $\{S_1, \ldots, S_{N_s}\}$, ils estiment donc la probabilité a posteriori $P(\mathcal{S}|O_1, \ldots, O_{N_o})$. Pour un concept donné C_k, dont les nœuds fils sont $fils(C_k) = \{S_1, \ldots, S_l\}$, ils estiment la probabilité a posteriori par :

$$P(C_k|O_1, \ldots, O_{N_o}) = \sum_{S_i \in fils(C_k)} P(C_k|S_i) \cdot P(S_i|O_1, \ldots, O_{N_o}). \tag{4.2}$$

Dans Fan et al. [59; 61], le même type de structure est utilisé, et les classifieurs sont appris en deux temps, toujours selon une démarche ascendante. Au niveau des feuilles, ils utilisent l'apprentissage multi-tâche pour avoir un terme de régularisation commun entre les fils d'un même nœud C_k. Ce même terme de régularisation est utilisé pour entraîner un *classifieur biaisé* sur C_k. Ils utilisent ensuite une procédure de *boosting hiérarchique* pour combiner les classifieurs multi-tâche : le classifieur au niveau de C_k est obtenu par régression logistique et combine le classifieur biaisé avec les classifieurs multi-tâche de ses fils.

Zweig et Weinshall [203] cherchent à utiliser la hiérarchie sémantique dans l'optique d'améliorer la catégorisation, et en particulier d'effectuer un transfert de connaissances entre les catégories. Il n'y est donc pas question de multilabel ou de multi-niveaux. Leurs expériences consistent à faire des classification binaires objet/fond, où les objets correspondent aux feuilles. Les classifieurs utilisés peuvent être simplement entraînés à partir d'exemples appartenant à la classe feuille, mais aussi avec des exemples des catégories voisines ou parentes. Ils testent diverses combinaisons de ces classifieurs pour observer leur comportement.

Les auteurs remarquent que les classifieurs correspondant à des objets plus spécifiques sont caractérisés par une forte précision et un faible rappel, tandis que pour des classes plus génériques, ils sont caractérisés par une faible précision et un fort rappel. Ils remarquent aussi que, moins il y a d'images à disposition pour l'apprentissage au niveau de la feuille, plus il est intéressant d'intégrer l'information des nœuds parents.

Marszałek et Schmid [126] cherchent à améliorer la catégorisation d'objets en exploitant les liens de hiérarchie. Ils construisent des classifieurs SVM binaires

associés aux arêtes de la manière suivante : pour une arête $i \prec k$, le classifieur est entraîné sur les images associées au label k uniquement, et différencie i de ses frères $j \prec k, j \neq i$. La classification se fait de manière descendante : on passe au nœud suivant i si le classifieur de l'arête $i \prec j$ donne une réponse positive. Une fonction de décision est calculée, pour un concept, en tenant compte des résultats sur tous les chemins menant de ce nœud à la racine.

4.2.3 Conclusion

On peut remarquer que les méthodes globales pour la classification structurée sont peu utilisées en pratique en vision : tous les exemples que nous avons vus utilisent l'approche locale et traitent les nœuds séparément (ou les arêtes). On peut trouver plusieurs raisons à cela. Par exemple, l'apprentissage global sur tous les nœuds peut poser un problème si la base d'apprentissage évolue pour intégrer de nouvelles classes : auquel cas, le classifieur doit entièrement être ré-appris. Au contraire, les approches traitant séparément les nœuds dans un premier temps permettent de n'apprendre que les nouveaux nœuds. Les méthodes locales permettent aussi plus de souplesse, par exemple pour traiter différemment les nœuds minimaux (comme dans [61]). Un autre problème lié aux approches globales est la complexité algorithmique : contrairement aux méthodes locales, les approches globales ne se prêtent pas facilement à une parallélisation. Enfin, il n'est pas évident que l'approche globale permette naturellement la gestion d'un compromis entre complexité et fiabilité.

4.3 Méthode

La structure sur laquelle nous allons nous appuyer pour construire l'algorithme d'annotation est celle de l'hypergraphe $\mathcal{H}$. On souhaite encore avoir une mesure de confiance pour chaque annotation. Pour cela, nous proposons d'estimer les probabilités a posteriori de chaque multilabel. Soit une entrée x dont on cherche à estimer le multilabel $\mathbf{t}$. On veut estimer :

$$\forall \mathbf{y}_j \in \mathcal{Y},\ p_j = p(\mathbf{t} \prec \mathbf{y}_j | x). \tag{4.3}$$

Suivant la tendance de l'état de l'art, nous avons opté pour une approche locale, dans laquelle un classifieur est appris pour chaque nœud de la structure. L'hypergraphe n'étant pas donné directement, il est déduit des connaissances sémantiques (i.e. des contraintes) à disposition. L'algorithme est le suivant :

- à partir du graphe Is-A, construction du diagramme de Hasse représentant $\mathcal{H}$ (section 4.3.1),
- soit une image $\mathcal{I}$ décrite par un vecteur x. Pour chaque nœud de $\mathcal{H}$, un classifieur local donne une première estimation de la probabilité p_j (section 4.3.2),
- les liens entre les nœuds sont introduits pour donner une estimation des probabilités globalement cohérente $\tilde{p}_j$ en chaque nœud (section 4.3.3).

Ces probabilités peuvent ensuite être utilisées de différentes manières, selon l'application. Pour de l'annotation d'images, on pourra sélectionner les nœuds de plus grande probabilité, selon le compromis fiabilité/précision déjà évoqué. Nous détaillerons comment dans la section 4.3.4.1. On peut également utiliser les probabilités de chaque image sur un nœud dans un contexte de recherche d'image : si on cherche les images de multilabel $\mathbf{y}_j$, on peut les retourner dans l'ordre des $\tilde{p}_j$ décroissants. Il est possible aussi de faire de la catégorisation selon le schéma classique en restreignant les décisions aux feuilles.

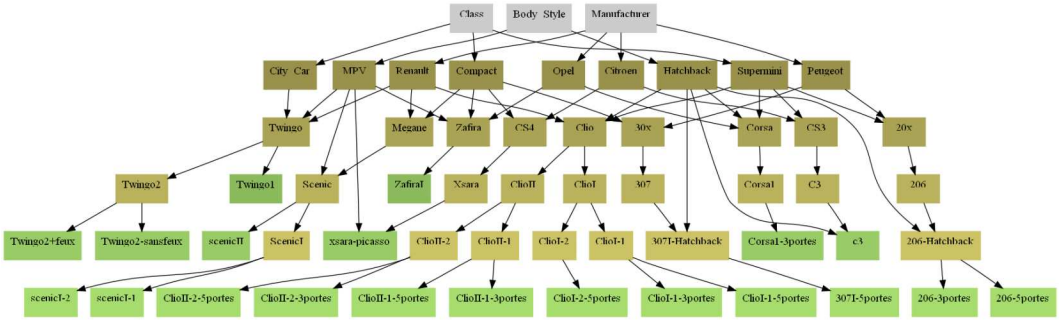


FIGURE 4.3 – *Graphe sémantique représentant les relations de type* Is-A *entre les modèles de voitures. Les nœuds racines sont des nœuds "abstraits", appelés* nœuds facettes, *qui ne désignent pas des concepts, mais des types de description.*

4.3.1 De la hiérarchie sémantique à l'hypergraphe

4.3.1.1 La hiérarchie sémantique : présentation

Pour construire l'hypergraphe $\mathcal{H}$, nous avons utilisé un graphe représentant les contraintes de hiérarchie et d'exclusion entre labels, c'est-à-dire les relations entre les différents modèles de voitures. Il a fallu construire ce graphe manuellement. En effet, à notre connaissance, il n'existe pas actuellement d'ontologie décrivant assez précisément ce domaine. L'exemple de WordNet donné par le tableau 2.3 montre une partie du vocabulaire lié aux voitures. Même en cherchant plus profond dans les hyponymes de "voiture", on constate que WordNet n'est pas assez spécifique pour effectuer une annotation précise de notre base. Par ailleurs, comme nous l'avons déjà relevé, WordNet ne tient pas compte des différents types de descriptions (facettes) et n'exprime pas les contraintes d'exclusion.

Pour construire le graphe Is-A, nous nous sommes intéressés à la manière de classer les automobiles. Les voitures peuvent être caractérisées selon plusieurs aspects. Seuls les critères purement visuels nous intéressent (nous excluons les caractéristiques type moteur etc.). Sans rentrer dans les détails techniques, nous en avons sélectionné trois : le constructeur, le type de carrosserie (la forme), et le segment automobile (décrivant, *grosso modo*, la taille et l'usage de la voiture). Ces critères correspondent aux différents points de vue descriptifs nécessaires pour illustrer l'approche multifacette. Dans la nomenclature des spécialistes de l'automobile, il y a des divergences de classification par rapport aux segments ; nous avons préféré garder une certaine neutralité, et simplifier le problème. Nous avons au moins deux bonnes raisons pour cela : (a) notre problème sera surtout de voir comment gérer ces différents modes de description ; (b) nous avons peu de modèles à caractériser, et la plupart des catégorisations "divergentes" n'apparaissent pas dans notre base.

Le graphe construit est donné en figure 4.3. Les différents points de vue apparaissent par l'intermédiaire des différentes racines. Les racines sont des nœuds particuliers : elles ne correspondent pas à des concepts (ou labels, c'est-à-dire qu'elles ne décrivent pas les voitures), mais à des types de descriptions. Ces racines sont à mettre en relation avec les différents "points d'accès" préconisés par Jörgensen [104], ou encore avec l'idée de facettes (quoiqu'ils ne correspondent pas aux facettes de Shatford Layne [166]). Par rapport à une taxonomie classique, ce graphe présente donc une structure particulière, où plusieurs arbres sont mélangés et ont toutes leurs feuilles en commun. Il ne présente pas de cycles. Enfin, la notion d'exclusion apparaît de manière logique : si une Peugeot est une 206, elle ne peut être aussi une 307, c'est-à-dire que des nœuds ayant au moins un parent commun sont mutuellement exclusifs.

Nous utilisons le terme de **hiérarchie sémantique** pour désigner ce type de structure, se voulant plus général que la taxonomie. Toute structure permettant de décrire les relations sémantiques IS-A sur les labels pourra être nommée hiérarchie sémantique, sans être limitée à un arbre (hiérarchie multiple). Nous proposons une description plus précise de cette structure.

4.3.1.2 La hiérarchie sémantique : définitions et propriétés

La hiérarchie sémantique décrivant le domaine de l'automobile, ou graphe Is-A, présente des particularités, qui demandent une approche adaptée. Contrairement aux taxonomies classiques, la représentation des différents points de vue nous amène à introduire deux types de nœuds, et deux types de relations entre les nœuds.

Définition 4.1 (nœud label) *Un nœud est dit* label *lorsqu'il caractérise un objet.*

Par exemple, les nœuds "Peugeot", "Clio", et "Citadine" sont des nœuds labels. On notera ces nœuds par la lettre l ou par un index i, j.

Définition 4.2 (nœud facette) *Un nœud est dit de type* facette *lorsqu'il définit un point de vue descriptif, c'est-à-dire qu'il caractérise un label.*

Par exemple, les nœuds "carrosserie", "segment", et "constructeur", sont des nœuds facettes. On notera ces nœuds par la lettre f.

La différence entre les types de liens vient du type des nœuds liés. Les seules relations apparaissant dans le graphe sont les liens nœud label→nœud label et nœud label→nœud facette. Seuls les premiers correspondent à une relation Is-A.

Tout nœud label du graphe est relié directement ou indirectement par une relation Is-A au nœud label racine "Voiture". Celui-ci n'est pas représenté, pour éviter la confusion avec les nœuds facettes, qui sont aussi des racines. Les liens Is-A définissent les liens de hiérarchie entre les labels. Cependant, la présence de deux types de nœuds permet d'introduire une propriété supplémentaire, l'exclusion. En effet, les différentes descriptions liées à un même point de vue descriptif, c'est-à-dire une même facette, sont mutuellement exclusives. Si l'on excepte le nœud "Voiture", implicite, et auquel tous les autres nœuds sont liés, cette propriété peut être formulée de la manière suivante :

Propriété 4.1 (Exclusion) *Deux nœuds labels ayant un parent commun sont mutuellement exclusifs (ou contradictoires).*

Si cette propriété est vérifiée par définition dans une taxonomie, elle ne l'est pas dans toutes les hiérarchies sémantiques. Comme nous l'avons déjà vu, par exemple, elle n'est pas vérifiée par WordNet. Elle est pourtant essentielle pour que la catégorisation ait un sens.

Par la suite, nous appellerons simplement *nœuds* les nœuds labels.

4.3.1.3 Application à l'interprétation multilabel

La notion de consistance est la même que celle vue au paragraphe 4.1.2. Le graphe $\mathcal{G}$ sert à expliciter les contraintes de hiérarchie et d'exclusion. Un multilabel $\mathbf{y}$ est donc dit **consistant** s'il respecte la structure de $\mathcal{G}$:

$$\forall i, j \in [N_l], \begin{cases} l_j \prec l_i \Rightarrow y_j \leq y_i, \\ l_i \text{ et } l_j \text{ ont un parent commun } \Rightarrow y_i + y_j \leq 1. \end{cases} \tag{4.4}$$

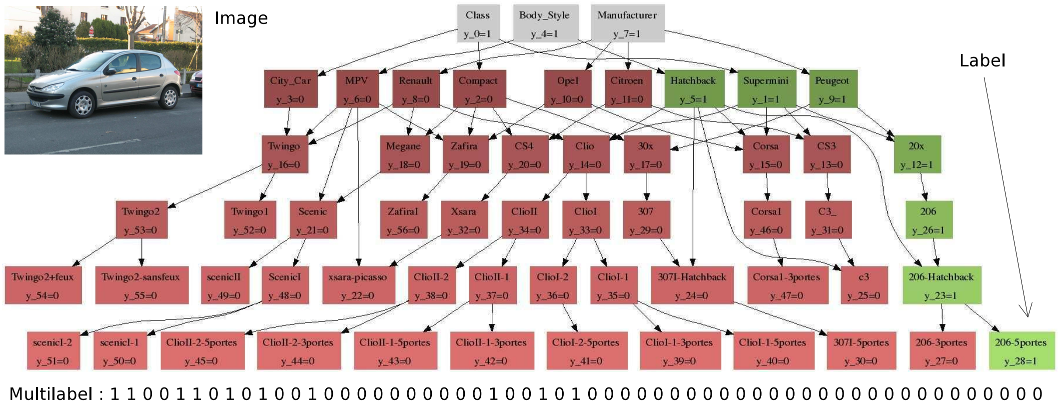


Figure 4.4 – *Exemple de vérité terrain pour une image de 206 5 portes. L'image est classée par son label de complexité maximale. Le multilabel est donné sous forme binaire, et sous forme graphique : les labels inclus dans le multilabel sont en vert, les autres en rouge. Les nœuds facettes sont représentés en gris.*

Le diagramme de Hasse $\mathcal{H}$ sera construit directement à partir de $\mathcal{G}$.

La figure 4.4 donne un exemple d'annotation pour une image de 206 5 portes et permet de voir plus clairement le lien entre le graphe $\mathcal{G}$ et la représentation binaire du multilabel.

Le graphe $\mathcal{H}$ n'a pas les propriétés du précédent graphe $\mathcal{G}$. En particulier, deux nœuds issus d'un même parent ne sont pas forcément exclusifs.

Les K classes données en entrée correspondent à des feuilles du graphe $\mathcal{G}$. Les multilabels minimaux correspondants $\mathbf{y}(k), k = 1 \dots K$ sont exactement les feuilles de $\mathcal{H}$. Les feuilles de $\mathcal{H}$ forment donc toujours la même partition de l'ensemble des objets.

4.3.1.4 Construction du diagramme de Hasse

La première étape de l'algorithme consiste à construire $\mathcal{H}$. Le diagramme de Hasse permet de représenter les relations d'ordre existant entre les éléments d'un ensemble fini. Dans le cas de l'hypergraphe, ces éléments sont les ensembles de labels. Autrement dit, les nœuds du diagramme de Hasse sont les multilabels $\mathbf{y}_i$. Une arête entre $\mathbf{y}_i$ et $\mathbf{y}_j$ représente la relation d'ordre $i \in fils(j) \Leftrightarrow \mathbf{y}_i \prec \mathbf{y}_j$ et $C(\mathbf{y}_i) = C(\mathbf{y}_j) + 1$.

Le nombre de multilabels consistants dépend du nombre de nœuds dans le graphe Is-A et de son degré de connectivité. Étant donné que les graphes étudiés ont une connectivité faible, il est facile d'énumérer l'ensemble des multilabels formant l'hypergraphe $\mathcal{H}$. $\mathcal{H}$ est donc construit en deux temps :

1. extraction des multilabels consistants pour construire $\mathcal{Y}$, l'ensemble des nœuds de $\mathcal{H}$,
2. construction des arêtes.

Chaque multilabel est noté sous sa forme binaire. On note $\mathbf{y}_0$ le multilabel minimal, qui correspond au label "voiture", c'est-à-dire , en pratique, à la liste de labels vide. Il est de complexité nulle. L'ensemble des multilabels de $\mathcal{H}$ est initialisé : $\mathcal{Y} = \{\mathbf{y}_0\}$, puis, pour chaque nœud i du graphe Is-A,

1. traiter les parents de i et mettre à jour $\mathcal{Y}_i$, la liste des multilabels ancêtres de i, et $\mathcal{Y} = \mathcal{Y} \cup \mathcal{Y}_i$,
2. ajouter $\mathbf{y}(i)$ le multilabel minimal correspondant à i : $\mathcal{Y} = \mathcal{Y} \cup \{\mathbf{y}(i)\}$,

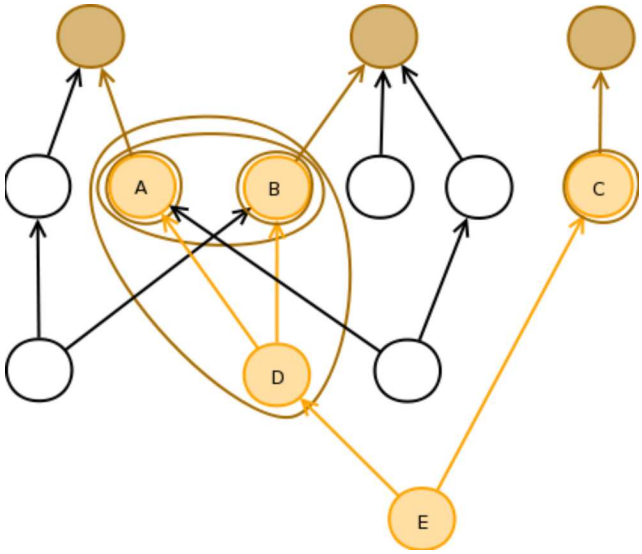


FIGURE 4.5 – *Combinaisons de multilabels. En marron, les nœuds facettes. En orange, les nœuds labels concernés par l'opération sur le nœud E, regroupés en multilabels, et marqués par des cerclages marrons. On s'intéresse au nœud E. Les ancêtres ayant été traités, ont donné les multilabels suivants : A :* $\mathbf{y}(A), \mathcal{Y}_A = \varnothing$ *; B :*$\mathbf{y}(B), \mathcal{Y}_B = \varnothing$ *; C :* $\mathbf{y}(C), \mathcal{Y}_C = \varnothing$ *; D :* $\mathbf{y}(C), \mathcal{Y}_D = \{\mathbf{y}(A), \mathbf{y}(B), \mathbf{y}(A) \vee \mathbf{y}(B)\}$*. Les multilabels issus des parents de E sont donc* $\mathcal{Y}_E = \{\mathbf{y}(C)\} \cup \{\mathbf{y}(D)\} \cup \mathcal{Y}_C \cup \mathcal{Y}_D$*. On ajoute* $\mathbf{y}(E)$*, puis les combinaisons des labels de* $\mathcal{Y}_E$*, combinant ceux issus de C avec ceux issus de D, soit* $\{\mathbf{y}(D) \vee \mathbf{y}(C), \mathbf{y}(A) \vee \mathbf{y}(C), \mathbf{y}(B) \vee \mathbf{y}(C), \mathbf{y}(A) \vee \mathbf{y}(B) \vee \mathbf{y}(C)\}$*.*

3. ajouter à $\mathcal{Y}$ les combinaisons des multilabels de $\mathcal{Y}_i$ issus des différents parents. La combinaison de deux labels $\mathbf{y}_1$ et $\mathbf{y}_2$ est obtenue par l'union des ensembles de labels, i.e. $comb(\mathbf{y}_1, \mathbf{y}_2) = \mathbf{y}_1 \vee \mathbf{y}_2$.

La figure 4.5 donne un exemple détaillé précisant ces opérations. L'ordre de traitement des labels n'a pas d'importance : la seule condition pour pouvoir traiter un nœud est d'avoir déjà traité tous ses parents. "Ajouter" est une opération sans répétition, c'est-à-dire que si un multilabel est déjà présent dans la liste, il n'est pas ajouté une seconde fois. Une combinaison est formée simplement par l'union de deux multilabels. Pour i donné, toutes les combinaisons de multilabels de $\mathcal{Y}_i$ sont possibles. En effet, si une combinaison était impossible, cela signifierait qu'il existe au moins deux nœuds ancêtres de i contradictoires. Auquel cas, ils ne pourraient être tous deux ensemble ancêtres de i.

L'ensemble des nœuds étant ainsi construit, et notant un nœud par son index, on a :

Proposition 4.1 *Pour tout* i, j *tels que* $i \prec j$*, il existe une suite de nœuds* $j_1, \ldots, j_k$ *vérifiant :*

(i) $j_1 = i$ *et* $j_k = j$,

(ii) $\forall l \in [k],\ j_l \in \mathcal{H}$,

(iii) $\forall l \in [k-1],\ \mathcal{C}(j_l) = \mathcal{C}(j_{l+1}) + 1$.

Si le graphe Is-A est un arbre, c'est évident, et la suite de nœuds est même unique. S'il y a plusieurs racines, il peut y avoir plusieurs suites qui mènent d'un nœud à la racine de $\mathcal{H}$. La croissance régulière de la complexité vient de la construction de nœuds : à chaque étape, on enlève un seul label en remontant au nœud parent. Si un nœud a plusieurs parents dans le graphe Is-A, on peut retrouver la différence d'un seul label en prenant une combinaison de ses parents.

Cette relation d'ordre partiel est utilisée pour construire les arêtes. Pour les construire, on parcourt la liste des nœuds obtenue. Soit (i, j), une arête est ajoutée

de i vers j si :

$$(i \prec j) \wedge (\mathcal{C}(i) = \mathcal{C}(j) + 1) \tag{4.5}$$

Le diagramme de Hasse obtenu à partir d'un arbre est aussi un arbre dont les multilabels peuvent être mis en bijection avec les labels. Dans ce cas, le multilabel $\mathbf{y}_j$ correspond au label l_j s'il contient l_j et tous les labels ancêtres de l_j. C'est également le multilabel minimal $\mathbf{y}(l_j)$. Le figure 4.6 présente le diagramme de Hasse $\mathcal{H}$ obtenu à partir du graphe $\mathcal{G}$ de la figure 4.4.

4.3.2 Probabilité d'un multilabel

Le calcul de la probabilité p_j associée à un multilabel consistant $\mathbf{y}_j$ se fait en deux étapes. Chaque multilabel est d'abord considéré de manière indépendante. Dans un second temps, les probabilités ainsi obtenues sont régularisées de manière à être cohérentes globalement (section 4.3.3).

La probabilité du nœud j est calculée par l'intermédiaire d'un algorithme de classification binaire un-contre-tous appliqué sur les signatures. Une image sera considérée comme positive si sa vérité terrain $\mathbf{t}$ contient tous les labels contenus dans le multilabel $\mathbf{y}_j$ associé au nœud, i.e. si $\mathbf{t} \prec \mathbf{y}_j$. On introduit le label binaire $z^{(j)}$ correspondant au nœud j et à l'entrée x, défini par :

$$z^{(j)} = \begin{cases} 1 & \text{si } \mathbf{t} \prec \mathbf{y}_j, \\ -1 & \text{sinon.} \end{cases} \tag{4.6}$$

Par exemple, une image étiquetée avec le label "206-5 portes" sera un exemple positif pour les classifieurs correspondant aux nœuds "206", "Peugeot", "berline", mais un exemple négatif pour les classifieurs de "206-3 portes", "Clio", "Compacte", etc. (voir figure 4.6).

On estime ainsi autant de classifieurs qu'il y a de nœuds dans $\mathcal{H}$.

Nous comparons plusieurs types de classifieurs. La régression logistique paraît une solution logique au problème, étant donné qu'elle permet une estimation directe des probabilités a posteriori. Cependant, nous avons voulu profiter de l'efficacité des SVM pour la classification. Nous avons donc aussi testé des classifieurs SVM, en estimant les probabilités a posteriori par l'algorithme de Platt [146].

Nous proposons une courte introduction aux SVMs en annexe, section B.4. Nous donnons ici une formulation adaptée au problème multilabel. Plus les nœuds sont de grande complexité, plus les ensembles d'apprentissage sont déséquilibrés entre positifs et négatifs. Ce déséquilibre de répartition des données est géré en faisant varier le paramètre C des SVM entre positifs et négatifs, et en adaptant le calcul de l'erreur. Le problème SVM au nœud j s'écrit donc :

$$\begin{aligned} &\min_{w_j, b_j} ||w_j||^2 + C_j^+ \sum_{i, z_i^{(j)}=1} \xi_{i,j} + C_j^- \sum_{i, z_i^{(j)}=-1} \xi_{i,j} \\ &\text{tel que } \forall i \in [n],\ z_i^{(j)}(\langle w_j, x_i \rangle + b_j) \geq 1 - \xi_{i,j},\ \xi_{i,j} \geq 0. \end{aligned} \tag{4.7}$$

Les paramètres C_j^+, C_j^- sont optimisés indépendamment pour chaque classifieur par validation croisée.

Selon le modèle classique, on estime les scores en chaque nœud par $f_j(x) = \sum_{i=1}^n \alpha_{j,i} \mathsf{k}(x_{j,i}, x)$. L'algorithme proposé par Platt [146] permet d'ajuster une sigmoïde sur les sorties SVM en estimant les paramètres A_j et B_j tels que :

$$\forall i \in [n], P(\mathbf{t}_i \prec \mathbf{y}_j | x_i) = P(z_i^{(j)} = 1 | f_j(x_i)) = \frac{1}{1 + e^{A_j f_j(x_i) + B_j}}. \tag{4.8}$$

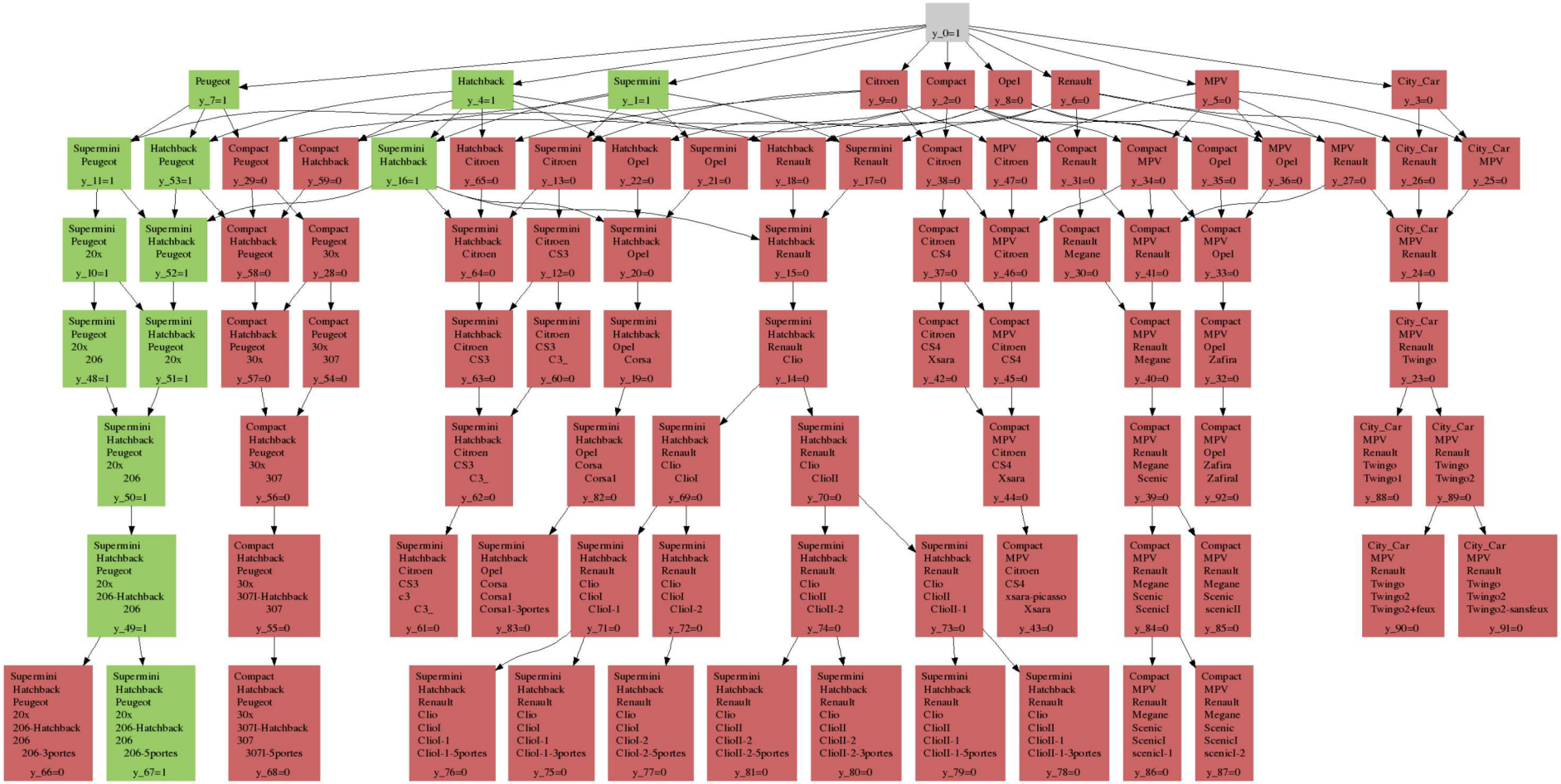


FIGURE 4.6 – *Représentation de $\mathcal{H}$ correspondant au graphe $\mathcal{G}$ de la figure 4.4. On retrouve le même exemple de vérité terrain, donnant l'appartenance de l'image correspondante aux ensembles d'apprentissage des différents classifieurs. En vert, les multilabels pour lesquels l'image est un positif. En rouge, les multilabels pour lequels elle est un négatif.*

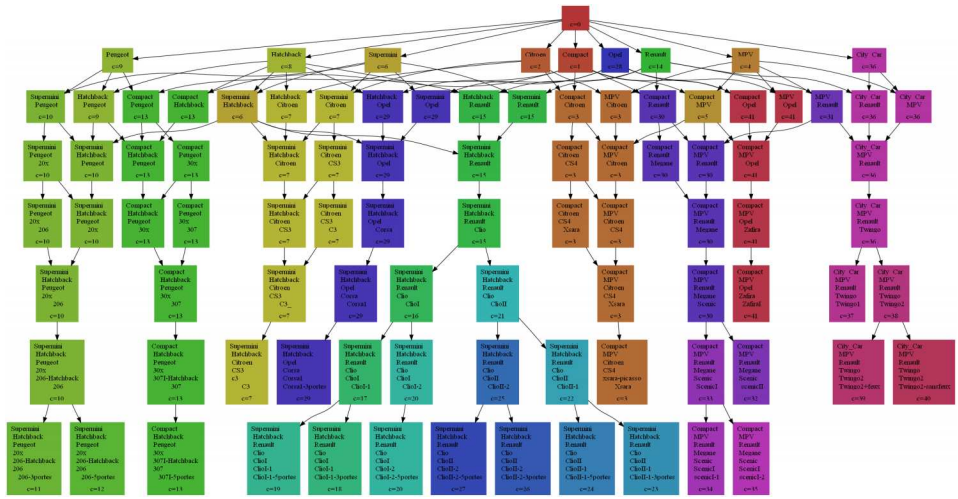


FIGURE 4.7 – *Visualisation des consets de l'hypergraphe : chaque conset est associé à une couleur et un index.*

Comme le notent Marszałek et Schmid [126], certains nœuds sont équivalents : en effet, si un nœud est lié à un seul parent, il est lié aux mêmes données. Ceci est lié au fait que le graphe Is-A est construit de manière à ce que les feuilles correspondent aux catégories. Les nœuds internes correspondent à des connaissances sémantiques connues indépendamment de ces catégories, et permettant d'en regrouper un plus grand nombre. Dans notre graphe, par exemple, une "Citadine, Opel" est forcément une "Corsa-1-3portes" (Notons que dans la réalité, il y a d'autres types de Corsa, et d'autres "Citadines, Opel", et ces nœuds ne pourraient donc pas être groupés). Pour traduire ces regroupements de nœuds, Marszałek et Schmid [126] introduisent la notion de *conset* :

Définition 4.3 (Conset) *Un* conset *est un ensemble de nœuds partageant exactement les mêmes exemples d'apprentissage.*

Si les feuilles définissent une partition des données, un conset peut encore être défini comme un ensemble de nœuds liés aux mêmes feuilles.

La notion de conset peut être utilisée pour réduire le nombre de classifieurs : par exemple, notre hypergraphe contient 92 nœuds, pour seulement 41 consets (sans compter la racine, à laquelle aucun classifieur n'est associé). Ils sont représentés figure 4.7. Cela peut représenter un gain intéressant, en particulier lors de la phase d'apprentissage. Dans nos expériences, nous constatons un gain de temps de près de 56%.

4.3.3 Régularisation des probabilités

Pour chaque nœud du graphe on a un classifieur, qui nous permet d'avoir en sortie une probabilité : $\forall i \in [n],\ \forall j \in \mathcal{H},\ p_j = p(\mathbf{t} \prec \mathbf{y}_j | x_i)$. Chaque p_j est calculée de manière indépendante pour chaque nœud. Les liens hiérarchiques ne sont pas explicitement utilisés, bien que les classifieurs soient liés par leurs ensembles d'apprentissage.

L'objectif est d'avoir des probabilités qui soient cohérentes vis-à-vis de la structure. Pour cela, on exploitera le fait que les feuilles représentent une partition des données, et donc que le multilabel de chaque nœud i doit vérifier :

$$p_i = \sum_{j \in \hat{\mathcal{H}}} [\![\mathbf{y}_j \prec \mathbf{y}_i]\!] \cdot p_j, \tag{4.9}$$

en notant $\hat{\mathcal{H}}$ l'ensemble des feuilles de $\mathcal{H}$ et, pour rappel, $[\![f]\!]$ où f est une proposition, la fonction binaire qui vaut 1 si f est vraie, 0 sinon.

Typiquement, on peut s'attendre à ce que les nœuds de faible complexité, disposant d'un plus grand nombre d'exemples positifs pour l'apprentissage, soient plus fiables. En pratique, on constate une certaine instabilité des classifieurs. Cette instabilité s'explique à la fois par le faible nombre de données, et par la forme des signatures. En effet, les signatures elles-mêmes sont issues de détecteurs appris sur peu de données, donc de détecteurs dont une bonne partie sont très peu fiables. En conséquence, ces signatures sont très bruitées. L'exploitation des liens de hiérarchie pour avoir des probabilités cohérentes permet de limiter les instabilités.

4.3.3.1 Lissage par minimisation d'un problème quadratique

L'ajout de la contrainte de régularisation 4.9 est fait en optimisant le problème quadratique suivant :

$$\min_{\tilde{\mathbf{p}}} \sum_{i \in \mathcal{H}} w_i \cdot (p_i - \sum_{j \in \hat{\mathcal{H}}} [\![\mathbf{y}_j \prec \mathbf{y}_i]\!] \cdot \tilde{p}_j)^2, \tag{4.10}$$

$$\text{tel que} \quad \sum_{j \in \hat{\mathcal{H}}} \tilde{p}_j = 1, \quad \text{et } \forall j, \tilde{p}_j \geq 0, \tag{4.11}$$

où $\tilde{\mathbf{p}}$ est le vecteur des probabilités estimées sur les feuilles seulement, et w_i est un poids sur le nœud i permettant par exemple de favoriser l'influence de nœuds à faible complexité. Le problème 4.10 estime des probabilités se rapprochant de la contrainte 4.9 tout en restant proche des résultats des classifieurs. Les contraintes 4.11 assurent que les valeurs estimées sont des probabilités. Ces probabilités estimées sur les feuilles sont ensuite propagées aux autres nœuds en utilisant l'équation 4.9.

Ce problème quadratique peut être réécrit sous forme matricielle de la manière suivante. Soit $\mathbf{P}$ la matrice regroupant les p_j obtenus pour chaque image, autrement dit telle que $\mathbf{P}_{i,j} = p(\mathbf{y} = \mathbf{y}_j | X = x_i)$, où x_i décrit l'image i. Pour chaque image on cherche à estimer le vecteur $\tilde{\mathbf{p}} \in [0,1]^{\mathcal{N}_f}$, où $\mathcal{N}_f = K$ est le nombre de feuilles.

Soit la matrice $\mathbf{L} \in \{0,1\}^{N_m \times \mathcal{N}_f}$ telle que

$$\mathbf{L}_{j_1,j_2} \triangleq \begin{cases} 1 & \text{si } \mathbf{y}_{j_2} \in feuilles(\mathbf{y}_{j_1}), \\ 0 & \text{sinon,} \end{cases} \tag{4.12}$$

en notant $feuilles(\mathbf{y})$ l'ensemble des feuilles descendantes de $\mathbf{y}$, i.e. $feuilles(\mathbf{y}) = \hat{\mathcal{H}} \cap desc(\mathbf{y})$. Enfin, soit $W = diag(w_j)$ et $A = \mathbf{P}(i,.)$ le vecteur des probabilités estimées par les classifieurs.

Soit

$$\Lambda = \sum_{i \in \mathcal{H}} w_i \cdot (p_i - \sum_{j \in \hat{\mathcal{H}}} [\![\mathbf{y}_j \prec \mathbf{y}_i]\!] \cdot \tilde{p}_j)^2 \tag{4.13}$$

On peut réécrire pour toute image i :

$$\begin{aligned} \Lambda &= \sum_j w_j (P_{i,j} - L(j,.)^\top \tilde{\mathbf{p}}_i)^2 && (4.14) \\ &= (A - L\tilde{\mathbf{p}}_i)^\top W (A - L\tilde{\mathbf{p}}_i) && (4.15) \\ &= A^\top W A - 2\tilde{\mathbf{p}}_i^\top L^\top W A + \tilde{\mathbf{p}}_i^\top L^\top W L) \tilde{\mathbf{p}}_i && (4.16) \\ &= A^\top W A + 2 * f^\top \tilde{\mathbf{p}}_i + \tilde{\mathbf{p}}_i^\top H \tilde{\mathbf{p}}_i && (4.17) \end{aligned}$$

où l'on a défini : $H \triangleq L^\top W L$ et $f \triangleq -L^\top W A$.

Le problème permet de gérer les liens entre les multilabels en utilisant leurs feuilles communes plutôt que les arêtes, ce qui permet d'éviter les difficultés liées aux boucles. Par exemple, si on utilisait les arêtes, la question de l'ordre de traitement se poserait. Avec notre méthode, l'ordre n'a plus d'importance.

4.3.3.2 Lissage par modèles de Bradley-Terry

Les modèles de Bradley-Terry permettent de formuler des probabilités multiclasses par rapport à des probabilités binaires entre des individus. Huang et al. [94] proposent de les généraliser à des groupes d'individus. Nous présentons ici le modèle dans ses grandes lignes. Nous proposons une description plus détaillée de l'approche en annexe, section B.4.2.2.

Des "parties" ont lieu entre deux équipes I_i^+ et I_i^- et donnent lieu à des résultats enregistrés par r_i et r'_i, où r_i est le nombre de fois que I_i^+ bat I_i^- et r'_i le nombre de fois que I_i^- bat I_i^+. Soit m le nombre de parties. Les équipes doivent vérifier pour $i \in [m]$:

$$I_i = I_i^+ \cup I_i^-,\ I_i^+ \neq \emptyset,\ I_i^- \neq \emptyset \text{ et } I_i^+ \cap I_i^- = \emptyset. \tag{4.18}$$

La solution est obtenue en minimisant l'opposé de la log-vraisemblance selon une procédure itérative. En classification, ce schéma peut être repris directement en interprétant différemment les quantités. Le but est d'estimer pour une entrée x les probabilités $P(t \prec \mathbf{y}(k))$, pour chaque catégorie $l_k, k \in [K]$. La probabilité $P(I_i^+ \text{ bat } I_i^-)$ correspondra à $P(\mathcal{L}_i^+ | \mathcal{L}_i)$, où $\mathcal{L}_i = \{l_k,\ k \in I_i\}$ et $\mathcal{L}_i^+ = \{l_k,\ k \in I_i^+\}$.

Pour reproduire le problème de classification multilabel, nous avons repris comme classes les feuilles de notre hypergraphe. Un classifieur en un nœud correspond à un match. Le modèle de Bradley-Terry généralisé peut être utilisé directement si pour tout $\mathbf{y}_i \in \mathcal{H}$, les ensembles I_i^+ et I_i^- sont définis de la manière suivante :

$$\forall k \in \hat{\mathcal{H}}, \mathbf{y}_k \in \begin{cases} I_i^+ & \text{si } \mathbf{y}_k \prec \mathbf{y}_i, \\ I_i^- & \text{sinon.} \end{cases} \tag{4.19}$$

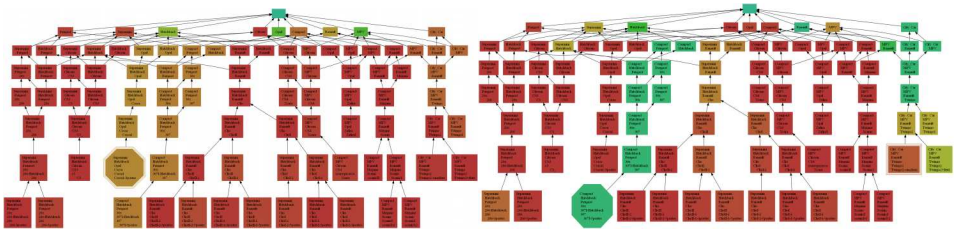
Les capacités estimées par le modèle de Bradley-Terry correspondent aux probabilités de chaque feuille, et sont obtenues à partir des mêmes probabilités binaires sur chaque nœud. Les liens hiérarchiques sont introduits implicitement au travers des matches : la capacité d'une équipe est égale à la somme des capacités de ses joueurs. De même que précédemment, la probabilité d'un multilabel quelconque est la somme des probabilités de ses feuilles.

La figure 4.8 montre sur un exemple les probabilités obtenues pour chacun des lissages.

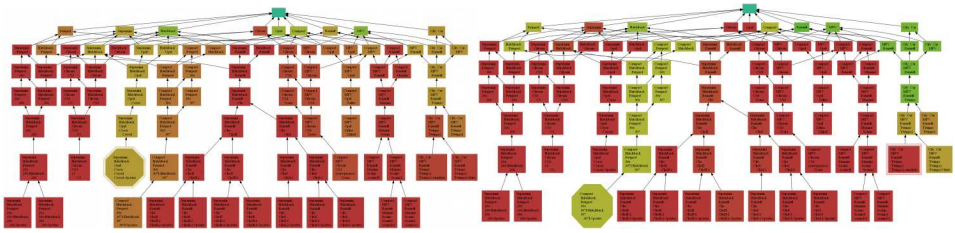
Dans les expériences, nous appellerons H-QP les méthodes régularisant les probabilités par un problème quadratique, et H-BT la méthode de régularisation par modèle de Bradley-Terry.

4.3.4 Annotation multifacette hiérarchique

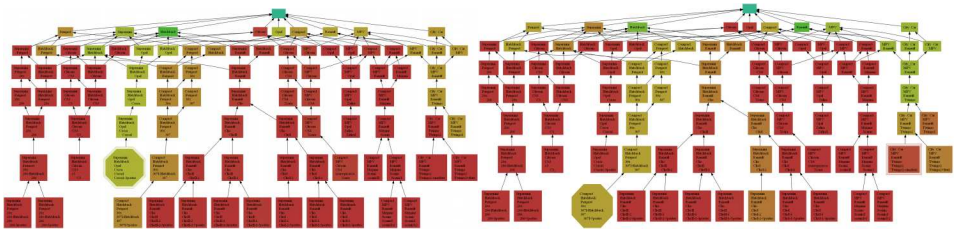
Nous avons vu comment associer à une image une probabilité en chaque nœud de $\mathcal{H}$. L'annotation d'images consiste à associer à l'image un ou plusieurs multilabels. Nous présentons ici comment obtenir des annotations en contrôlant un compromis précision/fiabilité, ainsi que d'autres manières d'exploiter ces probabilités.



(a) *Sans lissage*



(b) *Lissage* H-QP-ICOMP



(c) *Lissage* H-BT

FIGURE 4.8 – *Lissage des probabilités sur deux exemples. Première colonne : exemple 1, où la feuille de plus forte probabilité (nœud octogonal) correspond à l'annotation (double contour). Deuxième colonne : exemple 2, où les deux ne correspondent pas. (a) : Sans régularisation, les probabilités ne sont pas cohérentes. La régularisation par problème quadratique (b) ou par modèles de Bradley-Terry (c) donnent des résultats similaires. Dans l'exemple 1, la régularisation permet d'augmenter la probabilité de la bonne réponse. Dans l'exemple 2, la régularisation réduit suffisamment la probabilité sur la feuille fausse pour que l'on puisse récupérer un multilabel juste en réduisant la complexité. Une feuille ayant une probabilité non nulle alors que la majorité de ses ancêtres ont une probabilité nulle (exemple 2, extrême gauche) reçoit une probabilité nulle après régularisation.*

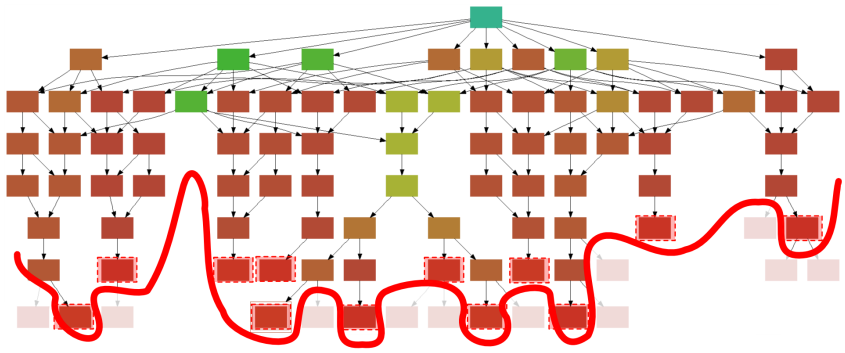


FIGURE 4.9 – *Annotation multilabel par contrôle d'un seuil de précision. Pour un seuil p, la frontière, marquée par une ligne rouge, est telle que tous les nœuds de probabilité supérieure à p, i.e. les nœuds de $\mathcal{Y}^p$, sont au-dessus. Les nœuds de probabilité inférieure à p sont grisés. Les nœuds de $\partial\mathcal{Y}^p$ sont encadrés en pointillés rouges.*

4.3.4.1 Annotation multifacette hiérarchique par gestion du compromis précision/fiabilité

L'algorithme présenté dans la section précédente renvoie un ensemble de probabilités tel que pour toute chaîne de multilabels $\mathbf{y}_1 \prec \cdots \prec \mathbf{y}_k$, on a $p_1 \leq \cdots \leq p_k$. Cette décroissance est assurée par l'équation (4.9).

Soit une image test dont la vérité terrain est le multilabel $\mathbf{t}$, tout multilabel tel que $\mathbf{t} \prec \mathbf{y}$ est une annotation correcte. On contrôle la précision de l'annotation par un paramètre de confiance p choisi par l'utilisateur. On souhaite que l'algorithme donne la description la plus précise possible avec un niveau de confiance au moins égal à p. Ce choix est fait grâce aux étapes suivantes :

1. Réduction de l'ensemble des multilabels $\mathcal{Y}$ par seuillage des probabilités par rapport à p :
$$\mathcal{Y}^p = \{\mathbf{y} \in \mathcal{Y} | p(\mathbf{t} \prec \mathbf{y}|x) \geq p\}, \tag{4.20}$$
2. Sélection de l'ensemble "frontière" constitué des multilabels de complexité maximale dans $\mathcal{Y}^p$:
$$\partial\mathcal{Y}^p = \{\mathbf{y} \in \mathcal{Y}^p | \forall \mathbf{y}' \in \mathcal{Y}, \text{ s.t. } \mathbf{y}' \prec \mathbf{y}, \mathbf{y}' \neq \mathbf{y}, \mathbf{y}' \notin \mathcal{Y}^p\}, \tag{4.21}$$
3. Sélection du multilabel estimé $\mathbf{y}^p$ de probabilité $p(\mathbf{y})$ maximale dans $\partial\mathcal{Y}^p$.

La figure 4.9 illustre ce procédé sur un exemple.

En faisant varier $p \in [0, 1[$, on peut constituer une liste de multilabels associés à des niveaux de confiance croissants, toujours supérieurs à p. Soient $p_1 < \cdots < p_R$, R seuils de confiance, la liste associée sera $\mathbf{y}_1 \prec \cdots \prec \mathbf{y}_R$, associée aux probabilités $p(\mathbf{t} \prec \mathbf{y}_1|x) \leq \cdots \leq p(\mathbf{t} \prec \mathbf{y}_R|x)$.

4.3.4.2 Annotation non hiérarchique

Pour comparer les résultats de classification aux différents niveaux de $\mathcal{H}$, proposons une méthode d'annotation utilisant la catégorisation au niveau des feuilles uniquement. Les résultats de classification pour des nœuds supérieurs sont obtenus de la manière suivante. Le coût est calculé aux différents niveaux de complexité, en faisant la moyenne des coûts des nœuds $\varepsilon(c), c \in [\mathcal{C}_{max}]$, où $\mathcal{C}_{max}$ est la complexité maximale d'un multilabel. Pour un nœud i, une image annotée par $\mathbf{t}$,

et un multilabel estimé $\mathbf{y}$, le coût 0/1 est donné par :

$$\ell^j_{0/1}(\mathbf{t},\mathbf{y}) = \begin{cases} 0 & \text{si } (\mathbf{t} \prec \mathbf{y}_j \wedge \mathbf{y} \prec \mathbf{y}_j) \vee (\mathbf{t} \not\prec \mathbf{y}_j \wedge \mathbf{y} \not\prec \mathbf{y}_j) \\ 1 & \text{sinon.} \end{cases} \tag{4.22}$$

Autrement dit, seules les confusions entre une feuille descendante et une feuille non-descendante du nœud de référence i ont un coût. Selon le déséquilibre de la base d'image, on pourra faire la moyenne des erreurs sur l'ensemble des exemples, ou par nœuds.

4.3.4.3 Annotation hiérarchique

Selon un principe analogue, il est également possible de contrôler les annotations en fixant une complexité maximale $\mathcal{C}$. On sélectionnera alors, pour une image donnée, le multilabel de probabilité maximale parmi l'ensemble des multilabels respectant la contrainte de complexité : dans le schéma précédent, cela revient à remplacer $\mathcal{Y}^p$ par $\mathcal{Y}_\mathcal{C} = \{\mathbf{y} \in \mathcal{Y} | \mathcal{C}(\mathbf{y}) \leq \mathcal{C}\}$, en reprenant la frontière de la même manière.

L'inconvénient de cette méthode est qu'elle évacue la notion de fiabilité : il n'y a plus de contrainte sur les probabilités "acceptables". Son avantage réside principalement en ce qu'elle permet de se ramener à des problèmes de classification multilabel équivalent avec un vocabulaire "plat". Par exemple, on pourra imposer une réponse au niveau des feuilles, ce qui permettra de se ramener au schéma classique de catégorisation. Ceci correspond à la réponse avec un seuil de fiabilité nul.

4.3.4.4 Application à la recherche d'images

Une autre manière d'exploiter l'ensemble des probabilités associées à une image apparaît naturellement lorsque l'on se place dans le contexte de la recherche d'image : étant donné une requête sous forme de mots-clés, on souhaite récupérer les images pour lesquelles ce mot-clé est une annotation.

En supposant que les mots-clés, ou groupes de mots-clés autorisés correspondent à des multilabels de $\mathcal{H}$, il suffit alors d'ordonner les images par rapport à la probabilité qui leur est associée en ce nœud, et d'afficher les N premières. Il est encore possible de fixer un seuil de fiabilité, pour se limiter aux réponses les plus sûres. D'une manière générale, la probabilité d'un nœud peut remplacer toute autre méthode de *ranking* utilisée classiquement en recherche d'images (voir par exemple [95, 191, 103, 133, 168, 190]).

Les probabilités sont calculées hors-ligne sur toute la base d'images, et pour tous les multilabels. Pour une requête donnée, le nombre de calculs à faire en ligne est donc très réduit, et la recherche d'images est très rapide. Si la requête est un label, la recherche est effectuée sur le multilabel minimal correspondant.

Les résultats obtenus sont présentés section 4.5.6.

4.3.5 Autres méthodes

4.3.5.1 Catégorisation ascendante

Dans le but d'étudier l'influence de la hiérarchie aux différentes étapes de l'estimation des probabilités. Nous comparons les méthodes basées sur l'hypergraphe avec des algorithmes plus standards. La première idée (appelée F) est de catégoriser au niveau des feuilles, sans utiliser la hiérarchie, et correspond à la méthode d'annotation présentée section 4.3.4.2.

La deuxième méthode (appelée FH) que nous testons n'utilise toujours pas la hiérarchie pour la classification, mais pour le calcul des probabilités. La catégorisation se fait à partir des feuilles uniquement. Une probabilité multiclasse est calculée sur les feuilles (soit par un lissage avec l'équation (4.10) simplifiée (FH-REG), soit par un modèle de Bradley-Terry un-contre-tous (FH-BT)). Les probabilités des feuilles sont propagées par somme aux multilabels de tout le graphe.

La troisième méthode (MH) que nous avons testée ne gère pas les problèmes multilabels. Il s'agit de faire des catégorisations multiclasse entre des nœuds de niveaux comparables : d'abord sur les nœuds de complexité maximale, puis sur les nœuds maximaux de complexité inférieure ou égale à $\mathcal{C}$, $\mathcal{C}$ diminuant. Ceci nécessite une partition des catégories à chaque niveau – donc pas de multilabel, pas d'hypergraphe, mais un arbre. Cependant, étant donné la structure du graphe Is-A, il est possible d'appliquer des catégorisations multiclasse successive dans les arbres sous-jacents, c'est-à-dire séparément pour chaque facette. Nous proposons de combiner les résultats ainsi obtenus pour avoir une réponse multilabel $\mathbf{y} \in \mathcal{H}$ de la manière suivante. Soit N_f le nombre de nœuds facettes à considérer, et $\mathbf{y}_1, \ldots, \mathbf{y}_{N_f}$ les multilabels prédits par les classifieurs issus de chaque facette à une complexité donnée. Soit $\mathbf{y} = \mathbf{y}_1 \vee \cdots \vee \mathbf{y}_{N_f}$ le multilabel union. Si $\mathbf{y} \in \mathcal{H}$, il est choisi comme prédiction finale. Sinon, le multilabel prédit est choisi au hasard dans l'ensemble des multilabels de $\mathcal{H}$ inclus dans $\mathbf{y}$ et de complexité maximale : $\mathcal{Y}_{\mathcal{C}} = \{\mathbf{y}' \in \mathcal{H}, \mathbf{y}' \subset \mathbf{y} | \forall \mathbf{y}_j \in \mathcal{H}, \mathbf{y}_j \subset \mathbf{y}, \mathcal{C}(\mathbf{y}_j) \leq \mathcal{C}(\mathbf{y}')\}$.

4.3.5.2 Catégorisation descendante

Nous comparons encore notre approche, qui prend la hiérarchie globalement, avec une approche descendante inspirée de Marszałek et Schmid [126]. Un classifieur est entraîné pour chaque arête, i.e. pour chaque paire $(\mathbf{y}_i, \mathbf{y}_j)$ telle que $\mathbf{y}_j \in par(\mathbf{y}_i)$. Le label binaire correspondant à l'arête $(\mathbf{y}_i, \mathbf{y}_j)$ sera noté $z^{(ij)}$ et défini pour une image $\mathcal{I}$ de multilabel $\mathbf{t} \prec \mathbf{y}_j$ par :

$$z^{(ij)} = \begin{cases} 1 & \text{si } \mathbf{t} \prec \mathbf{y}_i, \\ -1 & \text{sinon.} \end{cases} \tag{4.23}$$

Les images telles que $\mathbf{t} \not\prec \mathbf{y}_j$ ne font pas partie de l'ensemble d'apprentissage.

Le classifieur hiérarchique effectue une combinaison des fonctions de décision p_{ij} des différents classifieurs binaires. Nous l'appliquons sur les probabilités, et obtenons ainsi pour chaque nœud k :

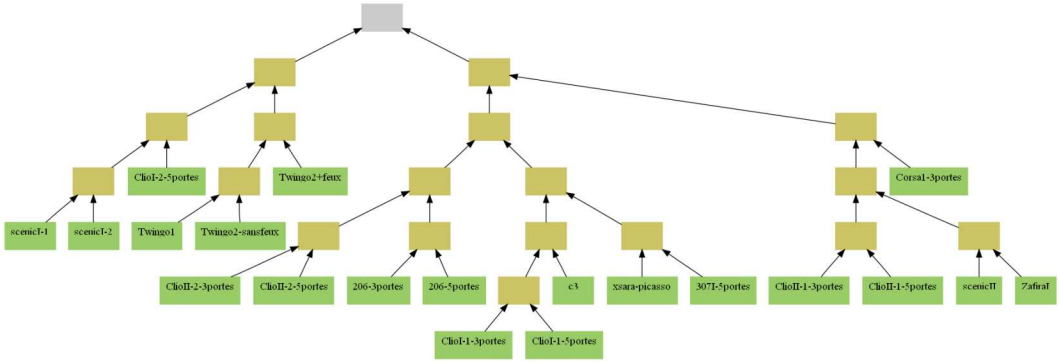
$$p_k = \max_{P \in \mathcal{P}(r,k)} \min_{(i,j) \in P} p_{ij}, \tag{4.24}$$

où $\mathcal{P}(r,k)$ est l'ensemble des chemins reliant la racine r au nœud k. Le graphe $\mathcal{H}$ ayant une seule racine, il est possible d'appliquer directement cette méthode.

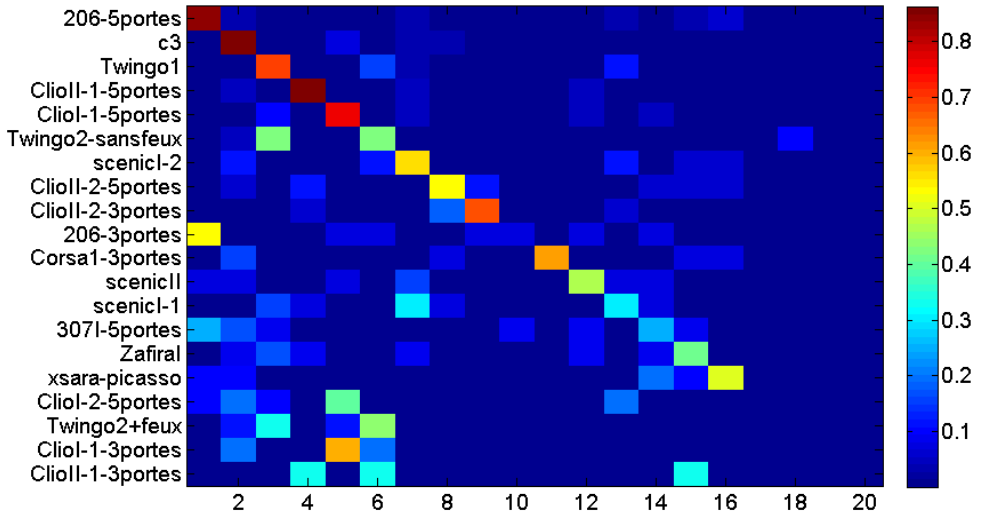
Un inconvénient notable des méthodes descendantes est qu'une erreur sur un nœud est transmise à ses fils, et est difficilement récupérable. En particulier, une erreur de branchement conduit typiquement à des catégorisations n'ayant plus de sens, pouvant donner des résultats imprévisibles.

4.3.6 Comparaison avec une taxonomie visuelle

Plusieurs travaux récents suggèrent de s'intéresser à la manière dont les données s'organisent en "taxonomie visuelle" : l'idée est de modéliser quels sont les liens visuels entre les catégories ([18, 82, 127, 167]). Nous les avons déjà passés en revue dans la section 2.2.8. Ce genre de taxonomie a l'avantage de pouvoir être



(a)



(b)

FIGURE 4.10 – *Taxonomie visuelle de la base de Voiture (a) construite à partir de la matrice de confusion donnée en (b). (a) :Les confusions au niveau des feuilles sont relativement proches de la réalité sémantique. Aux niveaux supérieurs, aucune correspondance n'apparaît. Dans la matrice de confusion (b), les classes sont ordonnées par rapport au nombre d'exemples disponibles : les confusions se font plutôt vers des classes proches sémantiquement et plus peuplées.*

construite automatiquement. Elle peut permettre une meilleure reconnaissance, ou une certaine compréhension de l'algorithme (confusions). En contrepartie, les nœuds créés par ce moyen n'ont a priori aucune signification, et ne permettent donc pas d'enrichir l'interprétation.

Notre objectif est ici d'étudier la différence de comportement entre une taxonomie visuelle et une représentation sémantique pour la classification multilabel hiérarchique. Pour cela, nous construisons une taxonomie visuelle selon l'un des protocoles proposé par Griffin et Perona [82] : après catégorisation (au niveau des feuilles), les catégories les plus sujettes à confusion sont groupées deux-à-deux. Nous obtenons ainsi une hiérarchie que nous utilisons pour estimer de nouvelles probabilités, selon l'une des méthode multilabel précédentes.

La figure 4.10 montre la taxonomie visuelle obtenue. On observe que les confusions plus importantes, au niveau des feuilles, correspondent à une proximité sémantique. Cependant, pour les niveaux supérieurs, il est plus difficile de trouver une correspondance avec le graphe sémantique. Les résultats de classification hiérarchique obtenus à partir de cette structure sont présentés en section 4.5.5.

4.3.7 Conclusion

Nous avons introduit différentes méthodes permettant d'exploiter la structure du vocabulaire, en l'intégrant à différents niveaux de l'algorithme. Une manière simple de les évaluer serait de faire la catégorisation au niveau des feuilles, et de comparer les taux de classification obtenus. Pour tenir compte de la hiérarchie, on peut utiliser une erreur adaptée. Cependant, notre objectif est de gérer un compromis entre fiabilité et précision, que ce genre d'évaluation simple évacue complètement. Nous introduisons donc maintenant une procédure d'évaluation mieux adaptée à l'exploitation d'une structure du vocabulaire.

4.4 Evaluation d'annotations multifacettes hiérarchiques

Les données à ce niveau sont les suivantes : un ensemble d'images $\mathcal{I}_i, i \in [n]$ où chacune est caractérisée par un multilabel $\mathbf{t}_i$ représentant la vérité terrain, et par un ensemble de probabilités $\{p^i_1, \ldots, p^i_{N_m}\}$ associées aux multilabels de $\mathcal{H}$.

L'objectif est de construire un critère d'évaluation qui permette de limiter l'importance d'une erreur à faible fiabilité d'une part, et à grande complexité d'autre part. En effet, l'erreur doit être réduite quand le taux de confiance est important et une confusion sur des classes génériques devrait être pénalisée plus fortement qu'une confusion entre des classes spécifiques. Par ailleurs, il est aussi intéressant de pondérer les taux d'erreur par rapport à l'écart au "vrai" label, en utilisant des fonctions de coût hiérarchiques.

4.4.1 Problématique et objectifs

Le problème de la classification hiérarchique est habituellement évalué de deux manières possibles :

- évaluation au niveau des feuilles, dans un problème de catégorisation,
- évaluation nœud par nœud, avec une mesure classique du taux de classification.

Ces évaluations sont parfois adaptées en utilisant des fonctions de coût hiérarchiques (Binder et al. [22]). Évaluer au niveau des feuilles présente l'avantage de pouvoir comparer les résultats aux approches "plates". Évaluer sur chaque nœud est a priori plus intéressant, dans le sens où cela permet de déterminer à quel niveau de complexité l'algorithme tire le meilleur profit de la hiérarchie.

Limites des représentations classiques de l'erreur

Cependant, lorsqu'il y a un grand nombre de nœuds, il n'est plus envisageable de présenter des résultats uniquement nœud par nœud. Dans notre cas par exemple, l'hypergraphe contient 92 multilabels, qui peuvent être réduits à 39 consets : pour rendre compte des résultats, il faudrait présenter une série de 39 taux de classification. Et nous n'avons que 20 catégories à la base. Très rapidement, ce genre de mesure devient illisible.

Nous présentons malgré tout quelques résultats de précision/rappel pour chaque nœud du graphe : cela permet une évaluation de l'algorithme dans le détail d'une part, et cela permet de se convaincre de la nécessité d'une mesure plus globale d'autre part.

Pour cela, nous adaptons la définition de la précision et du rappel par rapport à celle de la section 2.4, donnée dans le cas de la catégorisation ou de l'annotation avec un vocabulaire non structuré. On calcule la précision et le rappel par rapport à un multilabel donné $\mathbf{y}$ et pour un seuil de confiance p en utilisant la valeur $p(\mathbf{t}_i \prec \mathbf{y}|x_i)$ estimée. Soit T,FP et TP respectivement le nombre de positifs, faux

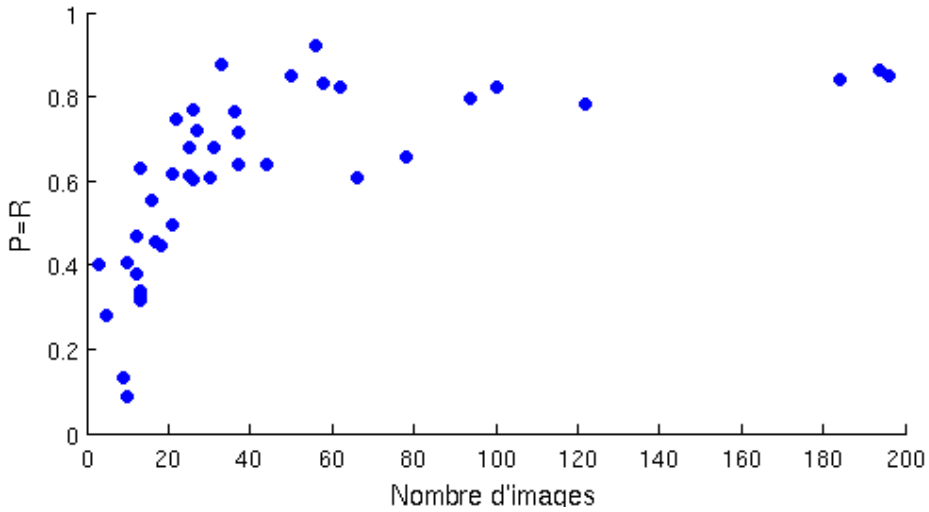


FIGURE 4.11 – *Influence du nombre d'images positives sur l'apprentissage : la courbe met en relation le nombre d'images à disposition et le taux de précision obtenu où P=R. On constate que les valeurs sont étroitement liées.*

positifs et vrais positifs, c'est-à-dire :

$$T(\mathbf{y}) = |\{i \in [n] | \, \mathbf{t}_i \prec \mathbf{y}\}|, \tag{4.25}$$

$$FP(\mathbf{y}) = |\{i \in [n] | \, (\mathbf{t}_i \not\prec \mathbf{y}) \wedge p(\mathbf{t}_i \prec \mathbf{y}|x_i) > p\}|, \tag{4.26}$$

$$TP(\mathbf{y}) = |\{i \in [n], \, (\mathbf{t}_i \prec \mathbf{y}) \wedge p(\mathbf{t}_i \prec \mathbf{y}|x_i) > p\}|. \tag{4.27}$$

La précision et le rappel sont définis par

$$R(\mathbf{y}) = \frac{TP(\mathbf{y})}{T(\mathbf{y})}, \tag{4.28}$$

$$P(\mathbf{y}) = \frac{TP(\mathbf{y})}{TP(\mathbf{y})+FP(\mathbf{y})}. \tag{4.29}$$

Nous reportons les résultats sous la forme de valeurs "P=R" : pour chaque nœud, on trace une courbe précision/rappel en faisant varier p entre 0 et 1, et on récupère la valeur de P à l'intersection avec $y = x$. La figures 4.12 et 4.13 présentent les performances obtenues en chaque nœud du graphe sous forme d'histogrammes, regroupés par complexité. Comme nous l'avons déjà remarqué, ce genre de représentation manque de lisibilité. On peut cependant déjà faire certaines remarques :

- le taux de classification d'un nœud est fortement lié au nombre d'exemples positifs disponibles pour l'apprentissage : à quelques exceptions près, nous pouvons dire qu'il faut au moins 20 exemples pour espérer avoir une valeur P=R supérieure à 0,5. La figure 4.11 montre cette tendance en mettant en relation la précision au point P=R avec le nombre d'images.
- nous n'observons pas les variations "inverses" de précision/rappel pointées par Zweig et Weinshall [203] : ceci est probablement dû au modèle de classification, qui est un-contre-tous pour nous, et un-contre-fond dans leur cas. Ainsi, dans notre modèle, une feuille est confrontée à ses voisins au moment du test, ce qui n'est pas le cas dans Zweig et Weinshall [203].

Nous présentons ces mêmes performances sous la forme d'un graphe, où la couleur du nœud correspond au taux de précision/rappel correspondant. Cette représentation présente l'avantage de permettre une visualisation des liens entre les taux de classification des différents nœuds. Cependant, il n'est pas aisé de comparer deux algorithmes par ce moyen.

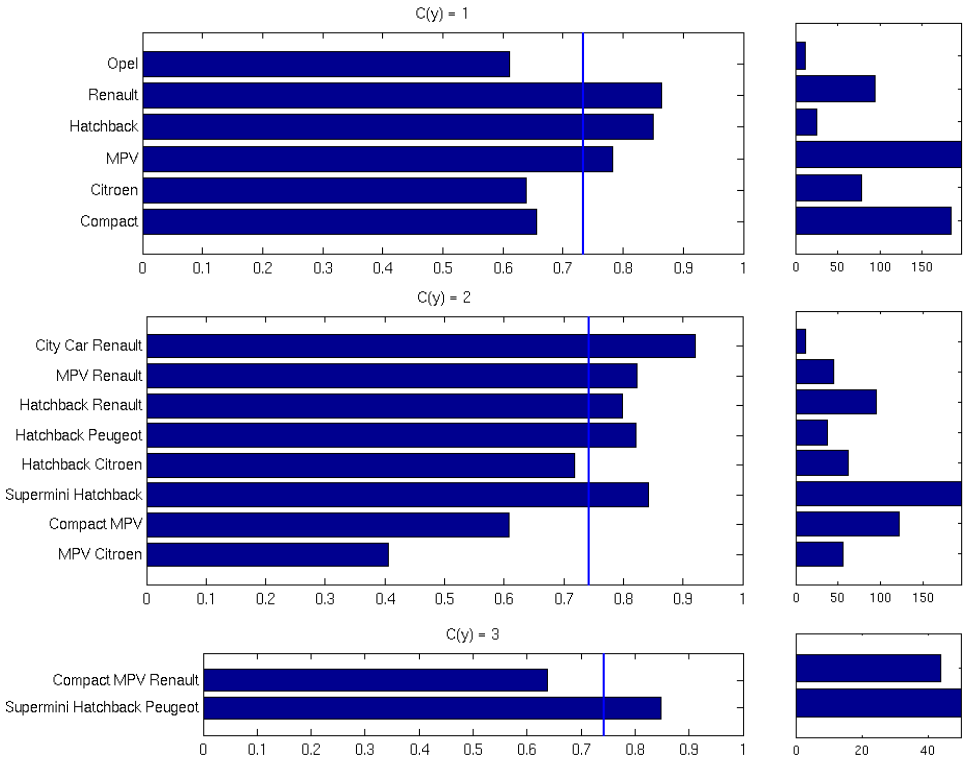


FIGURE 4.12 – *Performances obtenues en chaque nœud du graphe, évaluées par le point précision=rappel (première colonne), et mis en correspondance avec le nombre d'exemples positifs (première colonne). La ligne verticale représente la valeur P=R moyenne pour chaque complexité. Un seul nœud est donné par conset (celui de complexité maximale). Les nœuds sont regroupés selon leur complexité.*

Introduction de la courbe erreur/complexité

L'objectif de cette partie est d'élaborer des outils adaptés pour l'évaluation des performances en tenant compte du compromis précision-fiabilité. En effet, comme nous l'avons vu en introduction (en 2.4), les méthodes de mesure de performances utilisées actuellement en reconnaissance d'objet et en recherche d'images ne sont pas adaptées à un schéma hiérarchique tel que le nôtre. La contribution essentielle de cette partie de notre travail est une méthode de construction d'une courbe erreur/complexité permettant de combler cette lacune. Nous présentons d'autres critères d'évaluations permettant de compléter cette courbe.

Par construction, plus un multilabel est complexe, plus la fiabilité associée est faible. L'idée pour construire la courbe est d'utiliser cette propriété. Ainsi, en seuillant sur une fiabilité croissante, on obtient des complexités décroissantes. On observe l'évolution de l'erreur associée.

La construction de la courbe est expliquée plus en détail dans la section 4.4.2. La même courbe peut être calculée avec des erreurs différentes, présentées en section 4.4.3. Des éléments d'analyse de cette courbe sont donnés section 4.4.4. Les résultats obtenus dans le chapitre précédent sont présentés sous leur nouvelle forme (section 4.5), ce qui amène quelques remarques en section 4.6.

4.4.2 Construction de la courbe erreur/complexité

Nous proposons de construire la courbe en la paramétrant par le niveau de confiance $p \in [0,1]$. Pour une valeur de p donnée, une image est associée à un multilabel en utilisant la procédure d'annotation multifacette hiérarchique proposée au paragraphe 4.3.4.1. Un point $(c(p), \varepsilon(p))$ correspond à l'erreur et la complexité

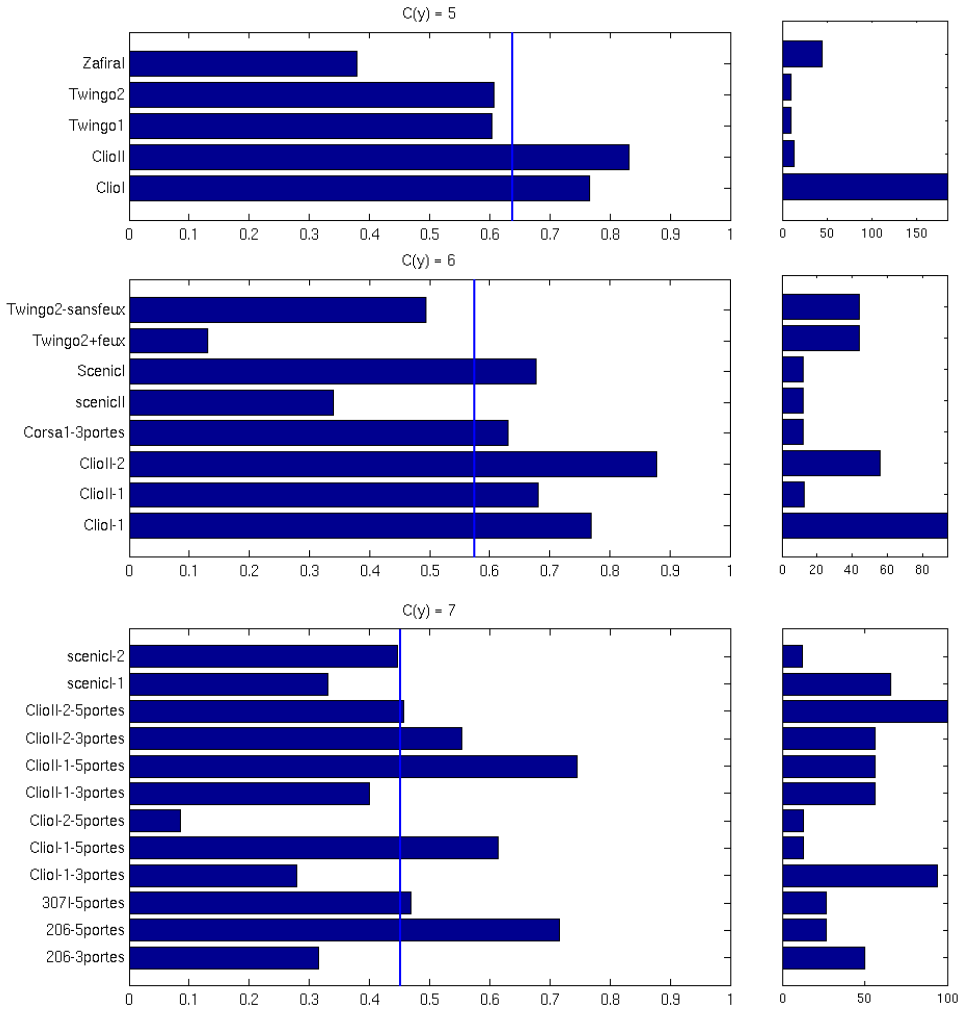
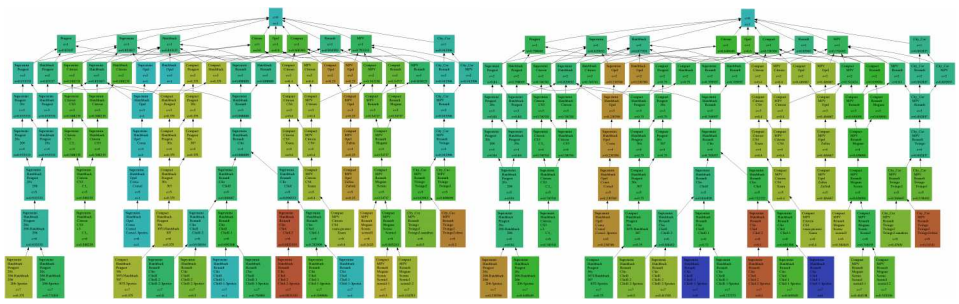


FIGURE 4.13 – *Performances obtenues en chaque nœud du graphe, évaluées par le point précision=rappel (première colonne), et mis en correspondance avec le nombre d'exemples positifs (première colonne). La ligne verticale représente la valeur P=R moyenne pour chaque complexité. Un seul nœud est donné par conset (celui de complexité maximale). Les nœuds sont regroupés selon leur complexité.*



(a) *Précision* (b) *Rappel*

FIGURE 4.14 – *Valeurs de précision/rappel en chaque nœud du graphe. Les couleurs varient du rouge (0) au bleu-turquoise (1) pour indiquer la valeur correspondante. Les calculs de précision/rappel sont faits pour* $p = 0,5$.

moyennes sur l'ensemble de test pour p :

$$c(p) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{C}(\mathbf{y}_i^p), \tag{4.30}$$
$$\varepsilon(p) = \frac{1}{N}\sum_{i=1}^{N} \ell(\mathbf{y}_i^p, \mathbf{t}_i), \tag{4.31}$$

où $\mathbf{t}_i$ est la vérité terrain de l'exemple i et ℓ est une fonction de coût, typiquement l'erreur o/1 :

$$\ell_{0/1}(\mathbf{y}_1, \mathbf{y}_2) = \begin{cases} 0 & \text{if } (\mathbf{y}_1 \prec \mathbf{y}_2) \vee (\mathbf{y}_2 \prec \mathbf{y}_1), \\ 1 & \text{sinon.} \end{cases} \tag{4.32}$$

Cette première version fait la moyenne des erreurs et des complexités sur tous les exemples donnés en entrée. Ce type de moyenne favorise les catégories les plus représentées. Dans le cas où la base est très déséquilibrée, cela peut changer considérablement les résultats. Selon l'application, il est peut être intéressant d'avoir un taux de réussite moyen par catégorie.

Lorsque l'on fait de la catégorisation à un seul niveau, il est assez évident de faire la moyenne par catégorie. Dans notre cas cependant, il faut adapter un minimum cette idée pour avoir des catégories à différents niveaux. Pour chaque nœud, on reprend les valeurs de rappel calculées pour le multilabel correspondant.

Pour un seuil donné p, c'est le multilabel $\mathbf{y}_r$ de complexité maximale et tel que $p_r > p$ qui sera sélectionné. Les résultats sont en fait ramenés aux feuilles de la manière suivante :

$$c(p) = \frac{1}{K}\sum_{k=1}^{K} c_k(p), \tag{4.33}$$
$$\varepsilon(p) = \frac{1}{K}\sum_{k=1}^{K} \varepsilon_k(p), \tag{4.34}$$
$$c_k(p) = \frac{1}{N_k}\sum_{\mathbf{y}(k)\prec\mathbf{y}\wedge\mathbf{t}_i\prec\mathbf{y}} \mathcal{C}(\mathbf{y}_i^p), \tag{4.35}$$
$$\varepsilon_k(p) = \frac{1}{N_k}\sum_{(\mathbf{y}(k)\prec\mathbf{y})\wedge(\mathbf{t}_i\prec\mathbf{y})} \ell(\mathbf{y}_i^p, \mathbf{t}_i), \tag{4.36}$$

où $N_k = |\{\mathbf{y}, \mathbf{y}(k) \prec \mathbf{y}\}|$.

4.4.3 Fonctions de coût hiérarchiques

Nous avons déjà vu comment mesurer l'erreur entre deux multilabels indépendamment de la connaissance de l'hypergraphe avec la fonction de coût $\ell_{0/1}$ (voir l'équation 4.32). Nous présentons ici quelques fonctions de coût intégrant une notion de distance entre les multilabels. La littérature propose un certain nombre de mesures de similarité entre concepts d'une taxonomie permettant de se rapprocher du jugement humain. Nous nous basons essentiellement sur les aperçus donnés par Resnik [154] et Cordì et al. [38].

L'idée est d'introduire dans la fonction de coût une mesure de la quantité d'information correspondant aux nœuds impliqués. Plus un nœud est complexe, i.e. précis, plus il contient d'information.

Nous proposons la fonction de coût suivante, que nous appelons E-loss :

$$\ell_E(\mathbf{y}_1, \mathbf{y}_2) = \frac{|\mathbf{y}_1 \oplus \mathbf{y}_2|}{|\mathbf{y}_1 \wedge \mathbf{y}_2| + \alpha \cdot |\mathbf{y}_1 \oplus \mathbf{y}_2|}, \tag{4.37}$$

en notant $\mathbf{y}_1 \oplus \mathbf{y}_2$ le nombre de labels différents entre les deux multilabels.

Introduit par Cesa-Bianchi et al. [34], le H-loss est défini sur une forêt contenant en tout N_l nœuds. Soit $\mathbf{y}_1$ et $\mathbf{y}_2$ deux multilabels, et $anc(j)$ l'ensemble des ancêtres d'un nœud j :

$$\ell_H(\mathbf{y}_1, \mathbf{y}_2) = \sum_{i=1}^{N_l} c_i [\![(y_{1,i} \neq y_{2,i}) \wedge (y_{1,j} = y_{2,j}), \forall j \in anc(i)]\!], \tag{4.38}$$

où les $c_1, \ldots, c_{N_l}$ sont des coûts à fixer. Autrement dit, en suivant un chemin depuis la racine, le long de chaque branche, dès qu'un nœud différent entre les deux multilabels est rencontré, le coût correspondant est ajouté, et ses descendants sont ignorés.

Nous avons adapté cette erreur à notre structure et à nos besoins. En effet, avec le H-loss tel qu'il est défini dans Cesa-Bianchi et al. [34], si un multilabel est inclus dans un autre, tout en étant différent, le coût ne sera pas nul. Or, nous considérons qu'il est juste de répondre un nœud parent, même si ce n'est pas optimal. Nous proposons donc une version simplifiée du H-loss, définie de la manière suivante :

$$\ell_H(\mathbf{y}_1, \mathbf{y}_2) = \begin{cases} 0 & \text{si } (\mathbf{y}_1 \prec \mathbf{y}_2) \vee (\mathbf{y}_2 \prec \mathbf{y}_1), \\ \gamma_{\mathbf{y}_1 \wedge \mathbf{y}_2} & \text{sinon.} \end{cases} \tag{4.39}$$

Par rapport au coût 0/1, cela consiste à pénaliser l'erreur en fonction du "plus grand multilabel commun" $\mathbf{y}_1 \wedge \mathbf{y}_2$. Nous souhaitons pénaliser une erreur d'autant plus qu'elle se situe sur un nœud de faible complexité, et nous avons donc choisi, pour un nœud $\mathbf{y}$, $\gamma_{\mathbf{y}} = \frac{1}{1+\kappa \cdot \mathcal{C}(\mathbf{y})}$.

On peut encore utiliser, comme Binder et al. [22], une fonction comptabilisant le nombre de labels différents entre les deux multilabels, que nous nommons X :

$$\ell_X(\mathbf{y}_1, \mathbf{y}_2) = \begin{cases} 0 & \text{si } (\mathbf{y}_1 \prec \mathbf{y}_2) \vee (\mathbf{y}_2 \prec \mathbf{y}_1), \\ \frac{1}{X_{max}} \cdot |\mathbf{y}_1 \oplus \mathbf{y}_2| & \text{sinon,} \end{cases} \tag{4.40}$$

où X_{max} correspond au coût maximal que l'on peut obtenir (dépendant de la profondeur de $\mathcal{H}$). Par rapport au E-loss, cette fonction sera plus sensible à la complexité d'un multilabel.

Le choix d'une fonction de coût adaptée dépend de l'utilisateur, et nous ne privilégieront donc pas une fonction par rapport à une autre.

4.4.4 Analyse de la courbe

La figure 4.15 montre une courbe représentative de ce que l'on obtient en pratique. Lorsque le seuil de confiance augmente, on obtient des réponses de moins en moins précises : la courbe se lit donc plutôt de droite à gauche. Le seuillage sur les probabilités en chaque nœud permet de contrôler un compromis entre fiabilité et complexité : plus on augmente le seuil, plus l'erreur diminue — mais aussi la complexité. On observe des erreurs moyennes et des complexités moyennes : pour une étude plus précise des algorithmes, il sera intéressant de considérer les histogrammes de répartition de ces valeurs.

La courbe permet également de retrouver le résultat que l'on obtiendrait en catégorisation au niveau de feuilles (i.e. , en autorisant ces seuls labels), en prenant le point de complexité maximale. On pourra donc ramener la courbe à une évaluation unique, non hiérarchique, permettant la comparaison avec les algorithmes classiques.

Il est assez délicat de résumer cette courbe par un seul chiffre : en effet, le but est de représenter un compromis, et la courbe est certainement le moyen le plus simple d'y parvenir. Cependant, il est parfois utile de pouvoir comparer deux approches directement à partir d'une seule valeur. Nous proposons alors d'utiliser l'aire sous la courbe (AUC) ou son complémentaire (CAUC), en ajoutant de plus une pondération selon la complexité présentant le plus d'intérêt pour l'utilisateur. Nous normalisons les bornes de la courbe, en la prolongeant par des droites, reliant les deux extrémités à l'origine et au point de complexité maximale. Nous pondérons les erreurs par une gaussienne centrée sur la complexité d'intérêt $\mathcal{C}_{pref}$,

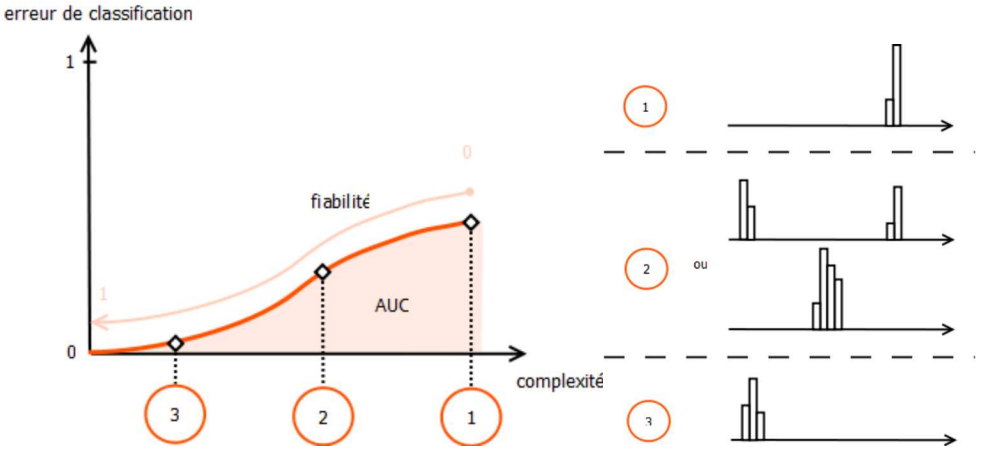


FIGURE 4.15 – *Analyse d'une courbe erreur/complexité, en (a), accompagnée d'histogrammes de complexité types en différents points de la courbe (b) ; (1) À seuil de confiance, on obtient les réponses de complexités maximales, et le taux d'erreur correspond au cas où l'on catégorise au niveau des feuilles, comparable aux taux d'erreur classiques. (2) Taux d'erreur à complexité intermédiaire. À ce niveau, il est intéressant de regarder aussi la répartition des complexités. (3) À seuil de confiance élevé, proche de 1, on observera généralement un taux d'erreur très proche de 0%, au prix d'une complexité réduite. En particulier, un certain nombre d'exemples ne seront plus classifiés.*

de largeur de bande moyenne ($\sigma = 0.3$). Dans nos expériences, nous avons utilisé une pondération par une gaussienne $\mathcal{N}(0,5;0,2)$ de manière à pénaliser surtout les complexités intermédiaires.

4.5 Expériences

Étant donné le peu de données ($|\mathcal{L}_b| = 318$), les résultats sont évalués par validation croisée :

- la base est partitionnée en N_f sous-ensembles $\mathcal{F}_1, \ldots, \mathcal{F}_{N_f}$,
- pour $f \in [N_f]$,
  - les paramètres du modèle sont sélectionnés par validation croisée sur l'ensemble $\mathcal{L}_b \backslash \mathcal{F}_f$,
  - les prédictions sont estimées sur les exemples de $\mathcal{F}_f$,
- les différentes erreurs sont estimées globalement sur $\mathcal{L}_b$.

La même partition des données est utilisée dans toutes les expériences, et dans le cas où tous les exemples sont impliqués (i.e. sauf dans la méthode *top-down*), les mêmes partitions des sous-ensembles sont utilisées pour la sélection par validation croisée. Nous estimons les courbes moyennes et leur variance en répétant les expériences 10 fois selon ce même principe.

4.5.1 Influence du classifieur binaire et des paramètres

Nous étudions ici la première partie de l'algorithme, dans un cadre un-contre-tous. Le but est double : (i) sélectionner un noyau adapté aux données, et (ii) établir l'estimation des probabilités en chaque nœud qui donne les meilleures performances, entre SVM+Platt et régression logistique.

La figure 4.16 donne l'évaluation des performances pour des classifieurs SVM avec différents noyaux. Le noyau linéaire donne les meilleures performances.

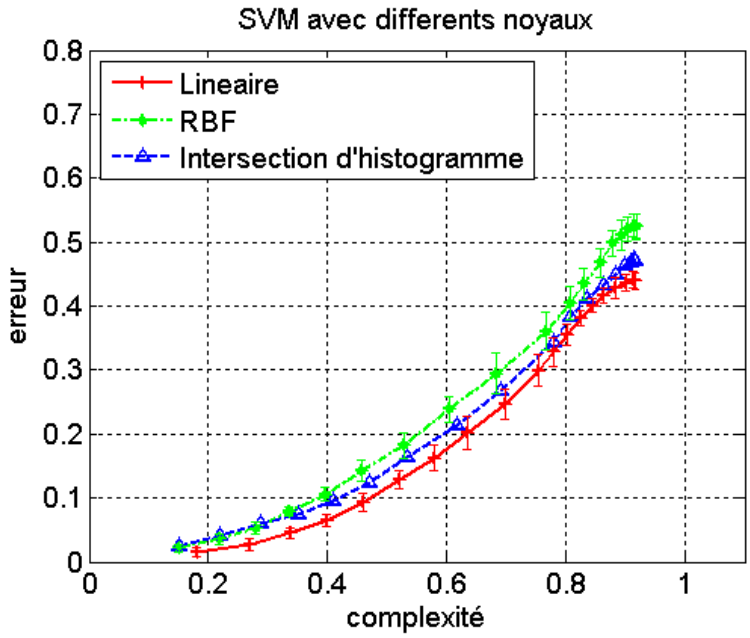


FIGURE 4.16 – *Courbes erreur/complexité obtenues pour différents noyaux, avec des classifieurs SVM.*

La figure 4.17 présente les courbes obtenues dans le cas linéaire pour le classifieur SVM et la régression logistique. Bien que la régression logistique permette naturellement l'estimation des probabilités, les SVMs suivi de l'estimation par Platt présentent un léger avantage. Dans la suite des expériences, c'est donc la méthode basée sur des classifieurs SVM linéaires qui nous servira de référence.

4.5.2 Différents lissages de probabilités

Nous étudions ici la deuxième partie de l'algorithme : une fois les probabilités estimées indépendamment en chaque nœud, quelle est la meilleure méthode pour avoir des probabilités cohérentes ? Nous comparons la méthode de Bradley-Terry (BT) à la nôtre, en utilisant différentes manières de calculer les poids $W = (w_1, \ldots, w_{N_m})$ dans la résolution du problème 4.10 :

1. QP-UNI : poids uniformes, $\forall j \in [N_m]$, $w_j = 1$,
2. QP-COMP : poids proportionnels à la complexité, pour augmenter l'influence des feuilles, $\forall j \in [N_m]$, $w_j = \mathcal{C}(\mathbf{y}_j) + 1$,
3. QP-ICOMP : poids inversement proportionnels à la complexité, pour augmenter l'influence des nœuds haut dans le graphe, $\forall j \in [N_m]$, $w_j = \frac{1}{\mathcal{C}(\mathbf{y}_j)+1}$.

Nous gardons comme référence les probabilités obtenues indépendamment sur chaque nœud : malgré les possibles incohérences, il est possible de les utiliser pour l'annotation (méthode IDP).

La figure 4.18 représente les courbes erreur/complexité obtenues pour les différents lissages. On constate que, si l'application d'un lissage est importante, le type de régularisation utilisée dans ce but a relativement peu d'importance. Le tableau 4.1 donne les valeurs de CAUC correspondant à ces mêmes courbes et confirme que les différences ne sont pas significatives.

La figure 4.19 donne quelques exemples d'annotation d'images obtenues avec et sans lissage.

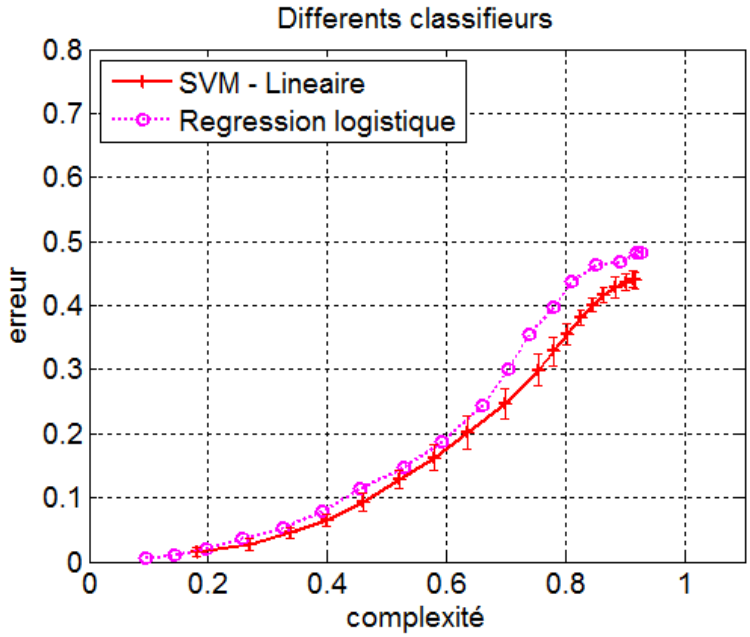
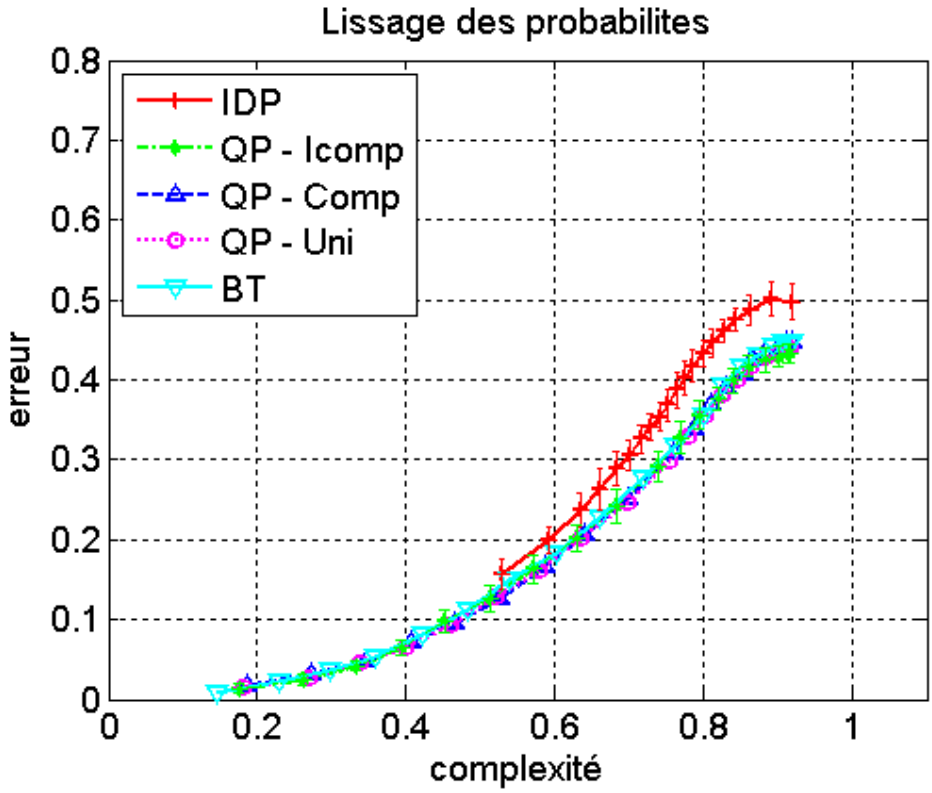


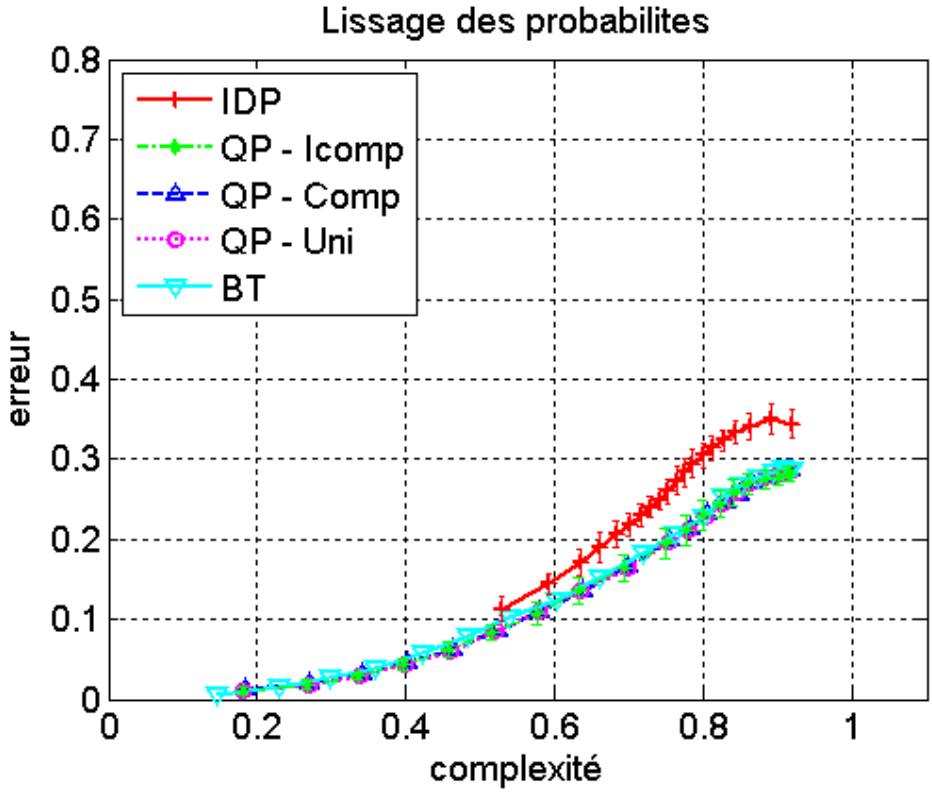
FIGURE 4.17 – *Courbes erreur/complexité obtenues différents estimateurs de probabilités.*

Méthode	**CAUC**
IDP	$85,86 \pm 1,02$
QP-UNI	$92,65 \pm 0,51$
QP-COMP	$92,73 \pm 0,57$
QP-ICOMP	$92,80 \pm 0,55$
BT	$92,53 \pm 0,54$

TABLE 4.1 – *Complémentaire de l'aire sous la courbe erreur/complexité pour différentes régularisations, pondérée par une gaussienne $\mathcal{N}(0,5;0,2)$, en %. Toutes les régularisations proposées donnent des résultats similaires.*



(a) *coût o/1*



(b) *E-loss*

FIGURE 4.18 – *Courbes erreur/complexité obtenues pour différentes régularisations des probabilités, pour différentes fonctions de coût hiérarchiques. Les différents coûts hiérarchiques donnent des courbes similaires, ce qui semble indiquer que des erreurs se retrouvent à tous les niveaux de la hiérarchie selon une distribution similaire pour toutes les méthodes.*

ClioII-1 5 portes

y	p
Berline, Citadine	0.99
Renault	0.78
Citadine, Renault	0.78
Clio	0.78
ClioII	0.85
ClioII-1	0.38
ClioII-1 5 portes	0.31

y	p
Berline, Citadine	1
Renault	0.69
Citadine, Renault	0.69
Clio	0.69
ClioII	0.65
ClioII-1	0.48
ClioII-1 5 portes	0.38

Zafira

y	p
Berline, Citadine	0.86
Citroën	0.44
Renault	0.77
Opel	0.64
Monospace	0.32
Clio	0.5
ClioII-2	0.38
C3	0.44
Zafira	0.64

y	p
Berline, Citadine	0.65
Citroën	0.29
Renault	0.41
Opel	0.30
Monospace	0.35
Clio	0.36
ClioII-2	0.25
C3	0.29
Zafira	0.30

Twingo2-sans feux

y	p
Renault	0.94
Monospace	0.66
Monospace, Renault	0.86
Mini	0.48
Berline, Citadine	0.32
ClioI-2	0.96
Twingo	0.48
Twingo1	0.24
Twingo2	0.25
Twingo2-sans feux	0.27

y	p
Renault	0.90
Monospace	0.60
Monospace, Renault	0.50
Mini	0.50
Berline, Citadine	0.40
ClioI-2	0.40
Twingo	0.50
Twingo1	0.24
Twingo2-sans feux	0.26

FIGURE 4.19 – *Exemples d'annotations sous forme de distributions de multilabels, avec et sans lissage des probabilités. Première ligne : images annotées. Deuxième ligne : annotation sans régularisation. Troisième ligne : annotation avec régularisation. Les multilabels (consets) de plus forts taux de confiance p sont indiqués (sous forme du label le plus précis lorsque c'est possible). La régularisation des probabilités permet d'avoir une répartition cohérente des niveaux de confiance p. Première colonne : exemple pour lequel les multilabels de forte probabilité sont cohérents. Deuxième colonne : exemple présentant une ambiguïté. Sans régularisation, le multilabel Zafira est plus probable que le multilabel Monospace, et les multilabels Clio, C3 ont une importance non négligeable. La régularisation permet de mettre en évidence la présence d'une ambiguité, et présente un cas difficile : l'annotation à seuillage nul est correcte (Zafira), et devient fausse quand le seuillage sur p augmente (Clio). Troisième colonne : cas où la régularisation ne modifie pas le comportement. L'annotation est fausse à seuillage nul (ClioI-2), mais correcte quand on augmente le seuil (à partir de 0.5, on obtient l'annotation Twingo).*

4.5.3 Comparaison des différentes méthodes hiérarchiques

La méthode un-contre-tous n'est pas forcément la meilleure a priori : en particulier, elle nécessite d'entraîner les classifieurs sur des données très déséquilibrées. Il est intéressant de voir si des méthodes plus équilibrées donnent de meilleures performances. Nous comparons les résultats obtenus par les méthodes de régularisation des probabilités (QP et BT) avec les différentes combinaisons de classifieurs binaires présentées dans la section 4.3.5 :

1. F : catégorisation sur les feuilles, utilisée comme référence pour tous les niveaux,
2. FH-REG : calcul des probabilités à partir des feuilles uniquement, revenant à résoudre 4.10 avec des poids uniformes sur les feuilles, et nuls partout ailleurs,
3. FH-BT : de même, en utilisant la régularisation par modèles de Bradley-Terry sur les feuilles,
4. MH : catégorisation multiclasse à différents niveaux,
5. TDH : calcul des probabilités de façon descendante, selon l'algorithme de Marszałek et Schmid [126].

Les résultats sont représentés par les courbes figure 4.20. Par rapport à la courbe précédente, où nous comparions les différentes méthodes de régularisation, ici les différentes méthodes présentent des comportements différents par rapport à la hiérarchie, ce qui s'observe sur les courbes d'erreurs avec des coûts hiérarchiques. Nous ne reportons pas les résultats avec le H-loss, qui sont très proches de ceux obtenus avec le E-loss. L'exploitation de la hiérarchie apporte clairement un gain sur la classification aux niveaux intermédiaires et supérieurs. L'utilisation d'un seuil de confiance permet d'éliminer les réponses les moins sûres en réduisant la complexité. En particulier, le système est capable de ne donner aucune réponse (multilabel vide).

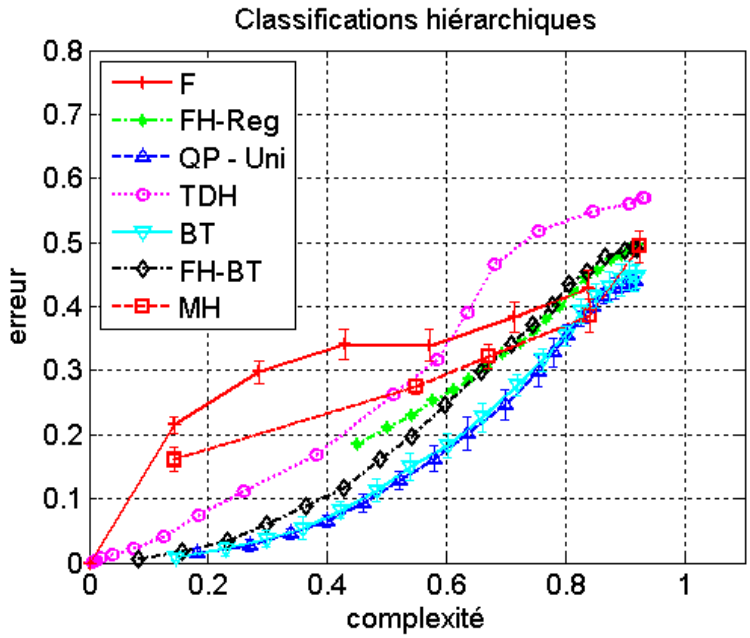
La figure 4.21 met en évidence ce phénomène. Nous avons représenté les histogrammes de répartition des complexités des multilabels estimés pour différentes valeurs du seuil de confiance. On peut constater (tout comme avec la courbe) que pour un même seuil de confiance la complexité moyenne des réponses n'est pas toujours la même. L'interprétation n'est donc pas aisée : il faut mettre en relation des points de complexités comparables.

4.5.4 Comparaison avec Caltech-101

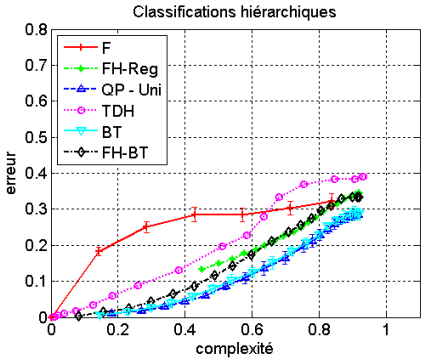
L'algorithme a été élaboré à l'origine pour travailler sur des bases de véhicules, et testé sur la base de voitures mentionnée. Nous avons aussi voulu tester ses capacités sur une base d'image généraliste telle que Caltech-101 [64]. Pour cela, nous avons construit une nouvelle hiérarchie décrivant les liens sémantiques entre les 101 catégories de Caltech-101. Par ailleurs, les signatures ont été calculées à partir de descripteurs classiques (SPM) : le classifieur de base sera donc basé sur un noyau intersection d'histogrammes plutôt qu'un noyau linéaire. Avec notre implémentation, nous obtenons les mêmes performances en catégorisation au niveau des feuilles que l'état de l'art, soit environ 67% de réussite en moyenne par catégorie.

La base Caltech-101 a des caractéristiques bien différentes de la base de voitures utilisée jusqu'à présent :

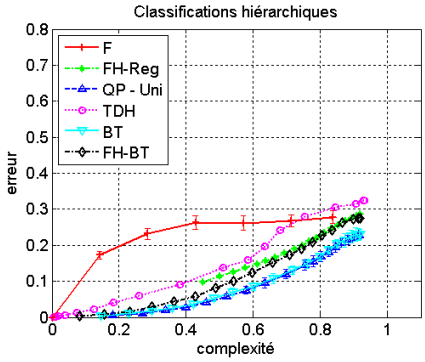
– les catégories d'objets qu'elle contient sont extrêmement variées, et la plupart ont des relations sémantiques très éloignées,



(a) *coût 0/1*

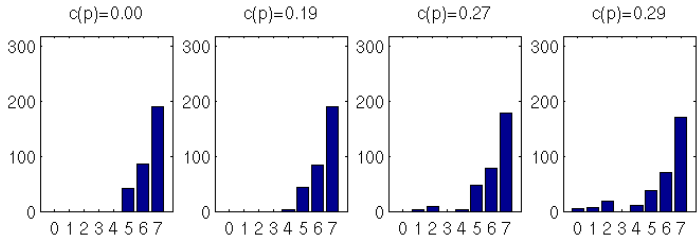


(b) *E-loss*

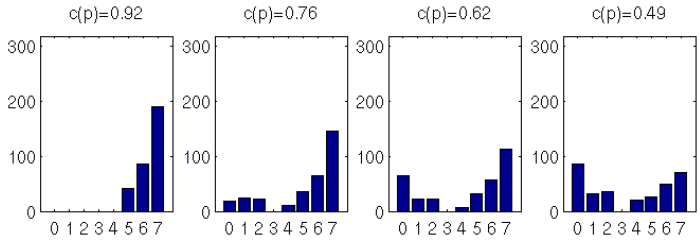


(c) *X-loss*

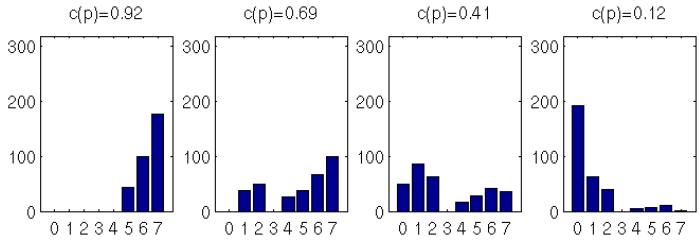
FIGURE 4.20 – *Courbes erreur/complexité obtenues pour les différentes méthodes hiérarchiques. L'exploitation de la hiérarchie apporte clairement un gain sur la classification aux niveaux intermédiaires et supérieurs.*



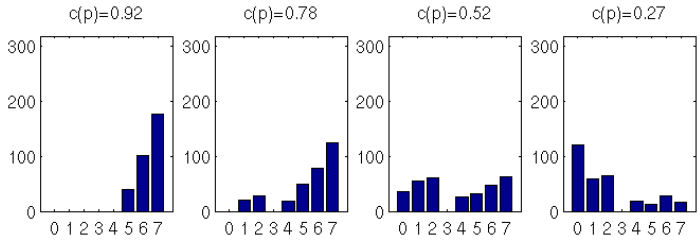
(a) F



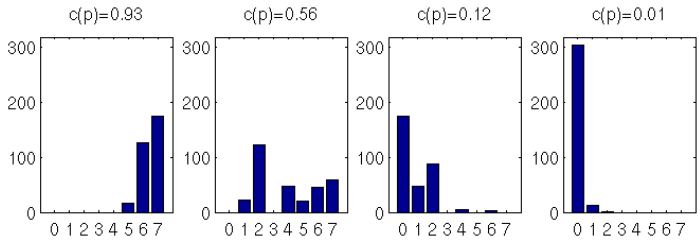
(b) FH-REG



(c) FH-BT



(d) QP-UNI



(e) TDH

FIGURE 4.21 – *Répartition des complexités pour différents points des courbes erreur/complexité présentées 4.20. En ligne, différentes méthodes (i.e. courbes). Sur chaque ligne, de gauche à droite, le seuil de confiance augmente, et les complexités moyennes associées $c(p)$ diminuent. Bien que les complexités ne puissent pas toujours être mises en correspondance, on peut observer différents types de comportement : certaines méthodes répartissent les réponses sur plus de complexités, certaines s'abstiennent plus souvent de répondre (complexité nulle).*

- les annotations se situent au niveau fondamental, i.e. les descriptions sont génériques,
- c'est une base assez grande : par rapport à la base de voitures, il y a beaucoup plus de catégories, et plus d'images par catégorie (minimum 31),
- la nature de la base fait que l'on utilise des signatures n'ayant pas de signification sémantique a priori.

La hiérarchie a été construite en s'inspirant de celle proposée pour Caltech-256 par Griffin et al. [81] et de WordNet, en veillant à respecter la contrainte d'exclusion (selon laquelle deux nœuds ayant au moins un parent commun sont mutuellement exclusifs). Elle est représentée figure 4.22. De nouveaux éléments d'annotations ont été introduits lorsque c'était possible, reprenant en particulier le problème de la catégorisation de scènes Intérieur/Extérieur : les catégories ont ainsi été réparties en trois groupes, selon que les objets de cette catégorie se trouvaient en intérieur (ex : piano), en extérieur (ex : nénuphar), ou les deux (ex : chien).

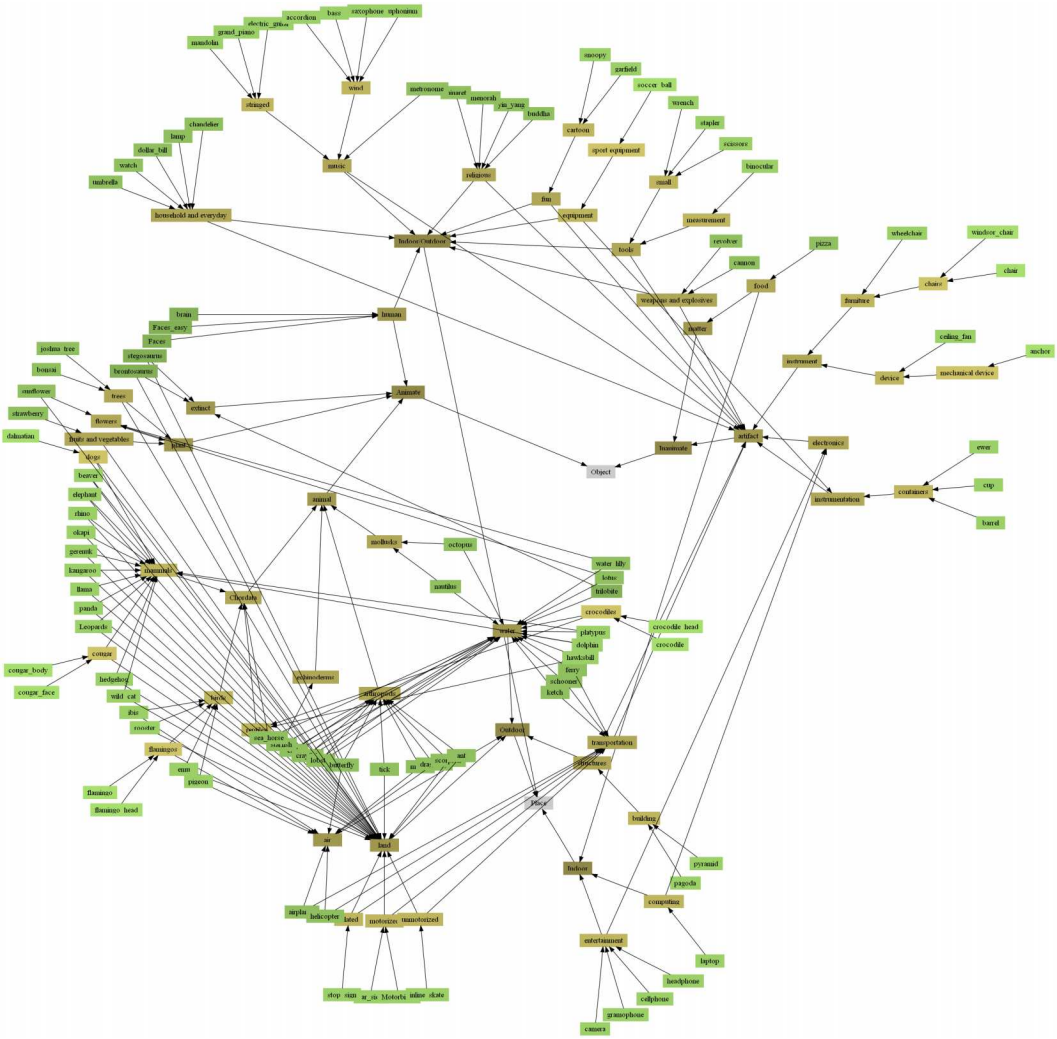
Etant donné les faibles liens sémantiques existant entre les catégories, on ne s'attend pas à améliorer grandement les performances en les introduisant. Cependant, l'annotation hiérarchique garde tout son sens : si les confusions sont entre des catégories proches ou éloignées sémantiquement, cela n'aura pas la même incidence sur les performances à niveau de complexité intermédiaire.

Nous effectuons les mêmes tests que précédemment, rapportés figure 4.23. Au niveau des feuilles, i.e. à complexité maximale, on retrouve les résultats de catégorisation de l'état de l'art (33% d'erreur). L'utilisation d'une régularisation sur tous les nœuds du graphe n'est pas avantageuse, dans le sens ou il y a plus d'erreur ; cependant les erreurs sont en générales moins graves que dans la méthode non hiérarchique, comme l'indique les courbes avec le X-loss. Les courbes ne sont pas directement comparables à celles obtenues pour la base de voitures, étant donné qu'elles dépendent des graphes : les niveaux de complexité ne varient pas de la même manière entre les deux bases (Voitures ou Caltech-101).

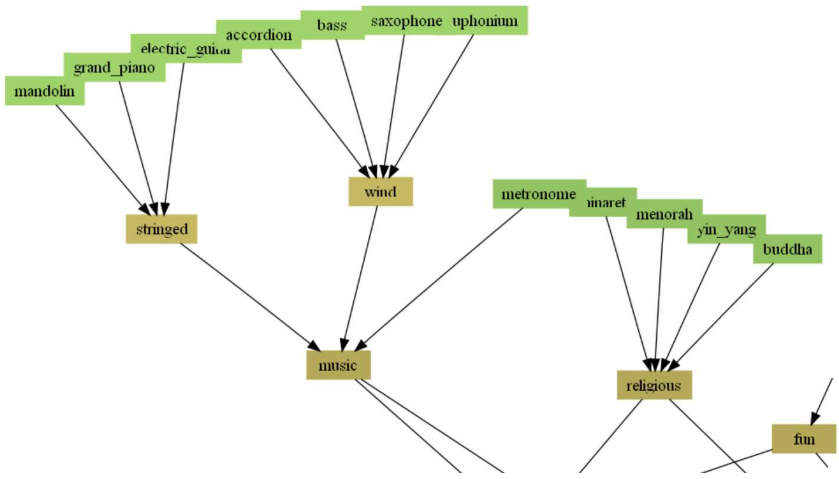
On remarque cependant que les mêmes méthodes ne présentent pas les mêmes améliorations de performances les unes par rapport aux autres. Ainsi, pour la base Caltech, il n'est pas intéressant d'utiliser les résultats de classification des nœuds internes au graphe, contrairement à la base de voitures. Une explication possible est liée à position des feuilles par rapport au niveau fondamental. Dans le cas de la base de voitures, les nœuds sont tous situés en dessous du niveau fondamental : on peut donc trouver une certaine homogénéité entre les catégories. Dans le cas de Caltech, les feuilles elle-mêmes sont situées au niveau fondamental : les catégories liées aux nœuds internes sont donc très génériques. Intuitivement, elles semblent plus difficiles à caractériser que les feuilles, et ainsi n'aident pas à la catégorisation. Selon les cas, on pourra donc utiliser la hiérarchie dès l'étape de catégorisation, comme pour la base de voitures, ou seulement au niveau de la décision, comme pour Caltech.

4.5.5 Comparaison de l'hypergraphe sémantique avec une taxonomie visuelle

La notion de complexité utilisée dans le tracé de la courbe dépend du graphe utilisé. Nous comparons ici l'utilisation de hiérarchies sémantiques avec des hiérarchies visuelles, construites automatiquement à partir des données et présentées au paragraphe 4.3.6. Dans le cas de la taxonomie visuelle, contrairement à l'hypergraphe, la complexité définie comme le nombre de nœuds ancêtres ne traduit pas la notion de précision sémantique. Il est possible de tracer les courbes ensemble en normalisant la complexité par rapport à la complexité maximale. La figure 4.24

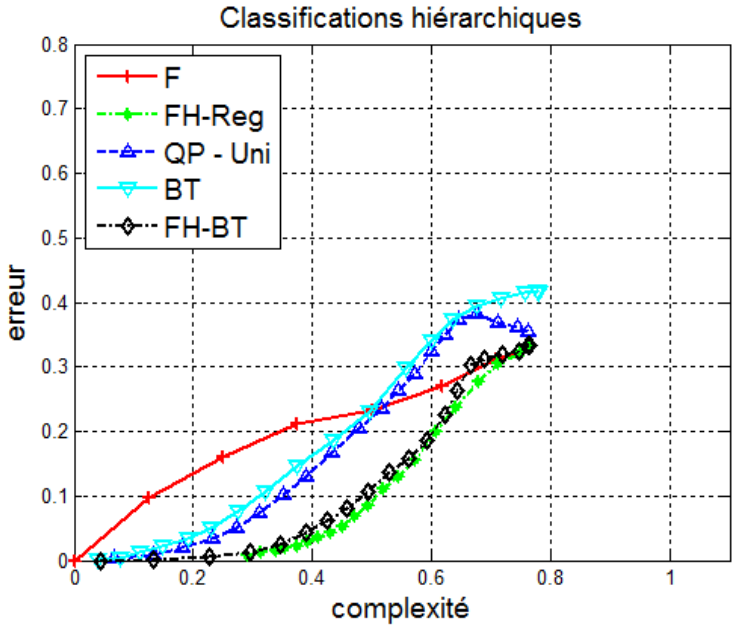


(a) *Graphe $\mathcal{G}$ complet*

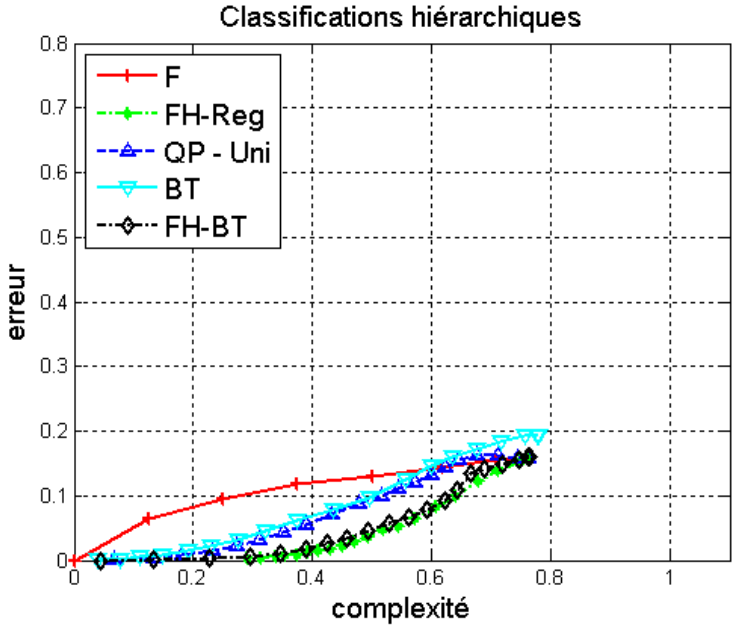


(b) *Zoom sur une partie du graphe $\mathcal{G}$*

FIGURE 4.22 – *Hiérarchie associée à Caltech-101.*



(a) *coût 0/1*



(b) *X-loss*

FIGURE 4.23 – *Courbes erreur/complexité obtenues Caltech-101. L'utilisation d'une régularisation sur tous les nœuds du graphe n'est pas avantageuse, mais la différence est moindre lorsque l'on évalue l'erreur avec une fonction de coût hiérarchique. Néanmoins l'utilisation de la hiérarchie pour la catégorisation améliore les performances (courbes* FH *meilleures que courbe F).*

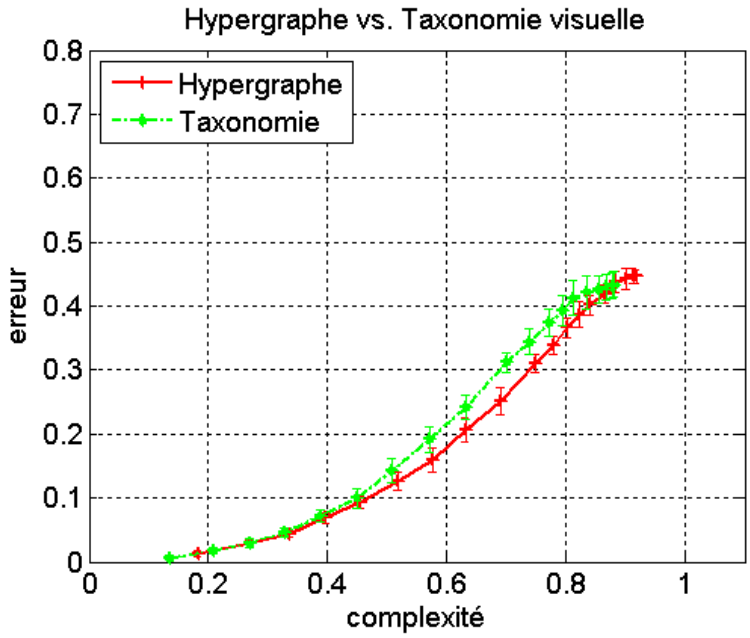


FIGURE 4.24 – *Courbes erreur/complexité obtenues pour l'hypergraphe et pour la taxonomie visuelle.*

donne les courbes obtenues sur la base de voitures. Les deux hiérarchies produisent des courbes similaires, ce qui indique que l'utilisation d'une taxonomie visuelle au lieu d'une hiérarchie sémantique ne produit pas une amélioration de performances dans le cas étudié. Par ailleurs, d'un point de vue interprétatif, les multilabels produits par la taxonomie visuelle présentent peu d'intérêt. Au mieux, ils informent qu'il y a ambiguïté.

4.5.6 Résultats en recherche d'images

Comme nous l'avons déjà remarqué, notre modèle d'annotation peut également être utilisé dans un contexte de recherche d'images. Nous présentons ici les résultats obtenus pour cette application, ainsi qu'ils ont été présentés dans Tousch et al. [181].

La figure 4.25 présente les premières images retournées par l'algorithme pour des requêtes à différents niveaux de complexité.

Les courbes précision/rappel correspondant à ces multilabels sont présentées figure 4.27, avec les courbes correspondant à des requêtes similaires. La courbe verte, et plus épaisse, représente la courbe moyenne obtenue sur toutes les requêtes. Un point est obtenu faisant varier le seuil sur la confiance, i.e. sur les probabilités, et en calculant les taux de précision/rappel sur les images retournées à chaque fois.

On peut remarquer des variations importantes des résultats par rapport aux différentes requêtes. Sur les exemples de la figure 4.27, les points correspondant à $P = R$ varient de $P = 50\%$ à $P = 90\%$, avec une moyenne de 68%. Globalement, les performances diminuent lorsque la complexité augmente, à quelques exceptions près. De même que dans les expériences précédentes, on peut surtout lier les performances au nombre d'éléments par catégorie, une propriété qui avait déjà été relevée par Huijsmans [95], et qui est évidemment liée à l'apprentissage. Quelques catégories avec peu d'exemples donnent malgré tout de bons résultats.

Les résultats obtenus avec la base Caltech-101 sont présentés figure 4.28. On retrouve les mêmes propriétés : le taux de réussite ne varie pas de façon monotone

(a) *Renault*

(b) *Clio I*

(c) *Clio II-2*

FIGURE 4.25 – *Les 12 premières images retournées par l'algorithme (de gauche à droite et de haut en bas) pour les labels (a) Renault, (b) ClioI et (c) ClioII-2 respectivement. Le système est capable de retourner des images de catégories présentant d'assez grandes variations intra-catégorie, comme dans (a) tout en étant capable de discriminer entre des petites variations. L'image entourée en rouge représente la seule erreur sur ces trois exemples.*

(a) *MPV*

(b) *Opel*

(c) *Compact,MPV*

FIGURE 4.26 – *Images retournées par l'algorithme pour des nœuds de faible complexité, présentant par conséquent de plus grandes variations intra-catégories : on constate que certaines catégories sont sur-représentées (par exemple, les Twingos parmi les monospaces, (a)), d'autres se répartissent plus équitablement (par exemple, les Corsa et les Zafira chez Opel (b)).*

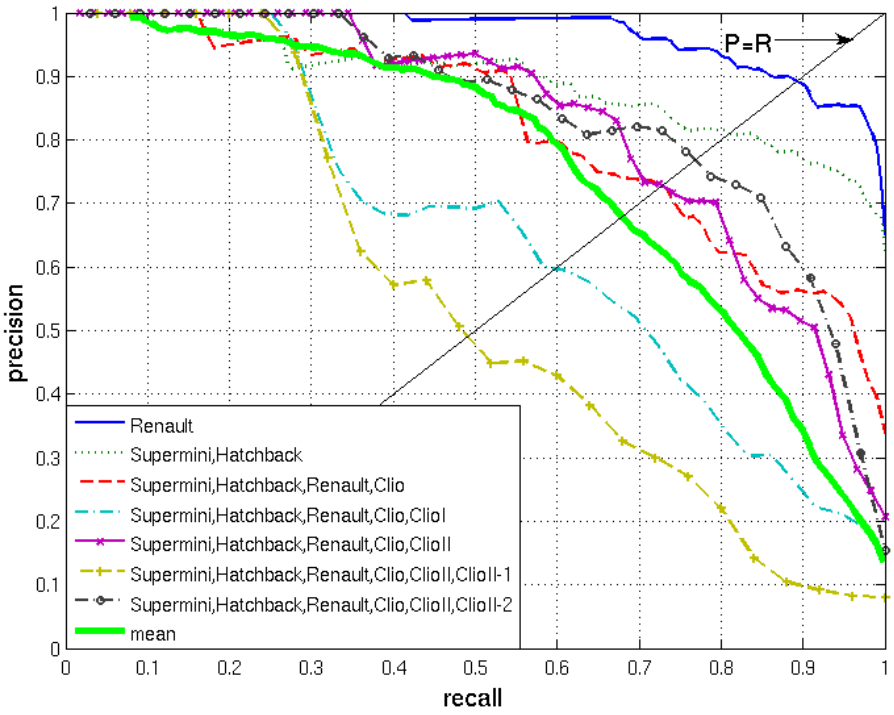


FIGURE 4.27 – *Courbes précision/rappel obtenues pour des labels à différents niveaux de précision sémantique. La recherche est effectuée sur l'ensemble de la base $\mathcal{L}_b$. La courbe verte est la courbe moyenne obtenue sur toute la base.*

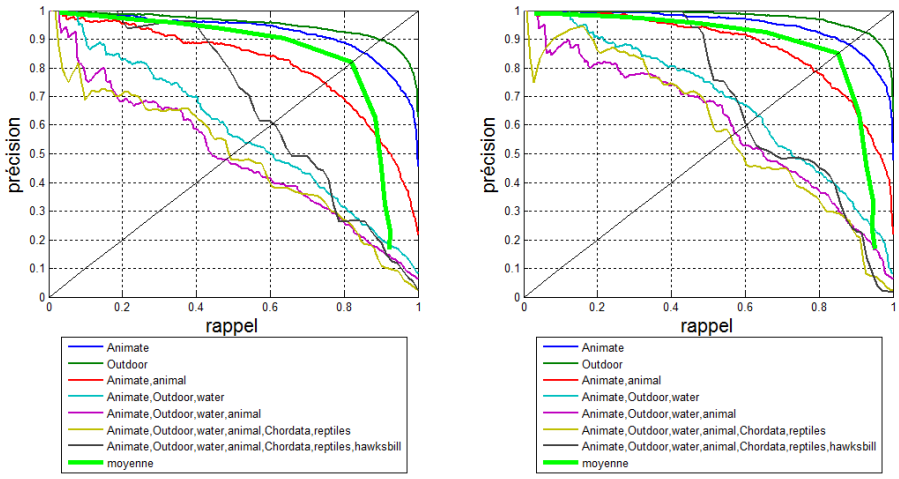
avec la complexité. Par exemple, il est a priori plus facile de retrouver des "animaux" que des "animaux marins". Ceci peut s'expliquer par le compromis entre la variété intra-classe, qui grandit forcément lorsque la complexité diminue, et la quantité d'images disponibles pour l'apprentissage. Ceci peut aussi s'expliquer, plus simplement, par le fait qu'il suffit qu'un nœud plus "facile" soit intégré au multilabel pour que la réussite de ce multilabel augmente.

Nous observons l'apport d'un lissage sur les probabilités, en utilisant la méthode par régularisation quadratique avec des poids inversement proportionnels à la complexité des nœuds. Les résultats montrent clairement qu'un traitement global des multilabels (i.e. avec régularisation) améliore les performances : la valeur $P = R$ passe de $82,1\%$ à $85,2\%$. La figure 4.29 montre les premières images retournées pour les deux méthodes, et permet de constater qu'il y a des différences de comportement importantes entre les deux.

4.6 Discussion et perspectives

Dans ce chapitre, nous avons élaboré une méthode, avec de nombreuses variations, permettant de construire une annotation multifacette hiérarchique associée à une image. Cette méthode s'appuie essentiellement sur les points suivants :

1. la modélisation des *multilabels consistants*, obtenus à partir d'un graphe représentant les liens entre les *labels* (sections 4.3.1.3 et 4.3.1.4),
2. le calcul de probabilités associées à chacun de ces multilabels (sections 4.3.2 et 4.3.3),
3. un processus de sélection des multilabels autorisant un contrôle du compromis précision/fiabilité (section 4.3.4.1).



(a) *Sans régularisation* (b) *régularisation* QP-ICOMP

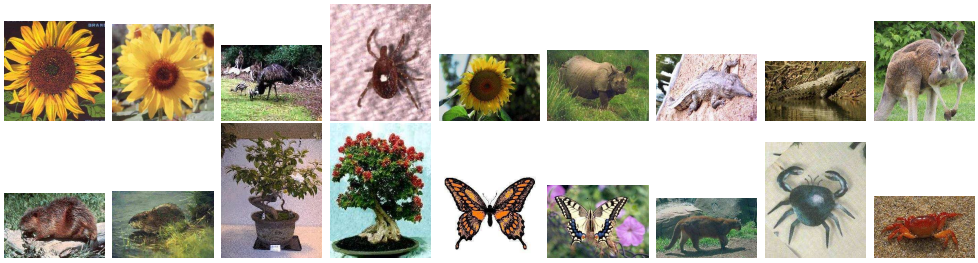
FIGURE 4.28 – *Courbes précision/rappel pour des labels de complexité croissante sur la base Caltech-101. En vert, la courbe courbe moyenne calculée sur toutes les catégories pour lesquelles $N_{test} \geq 5$. Les autres courbes sont obtenues pour le multilabel correspondant au caret (une tortue de mer) et tous ses ancêtres. La régularisation des probabilités améliore les performances globales.*

Nous avons également conçu des outils spécialisés dans l'analyse des performances de méthodes destinées à cette tâche.

Nous avons testé cette méthode sur deux bases d'images : une base de voitures, et une base classique, Caltech-101. Les deux bases possèdent des propriétés différentes, que nous avons soulignées, et que les tests effectués confirment. La partie de l'étude concernant la base de voitures a fait l'objet d'une publication à la conférence "Multimedia Information Retrieval", Tousch et al. [181].

Dans les deux cas, l'extension à une annotation multifacette hiérarchique peut se faire sans dégradation de performances. Mieux, plusieurs méthodes permettent d'améliorer la méthode "plate", en particulier dans le cas de la base de voitures, c'est-à-dire dans le cas où il existe des relations sémantiques fortes. Par rapport aux études précédentes sur la catégorisation hiérarchique, ou sur l'annotation multilabel, notre travail présente l'avantage d'intégrer les deux aspects, multilabel et hiérarchique, dans un cadre unifié, et fournit une analyse plus poussée de l'aspect hiérarchique par rapport aux seuls aspects de catégorisation. Tandis que les études de l'état de l'art présentaient des résultats sous forme locale, par multilabel, nous proposons une représentation globale, permettant une meilleure analyse des performances à tous les niveaux de la hiérarchie. Ce genre d'analyse répond mieux aux besoins qui apparaissent aujourd'hui dans les communautés traitant de l'interprétation automatique des images : toujours plus de catégories, toujours plus de degrés de liberté. Bien que la complexité d'apprentissage soit plus grande qu'avec un vocabulaire simple, l'introduction d'un vocabulaire structuré est essentielle pour offrir un contrôle sur la fiabilité d'un système.

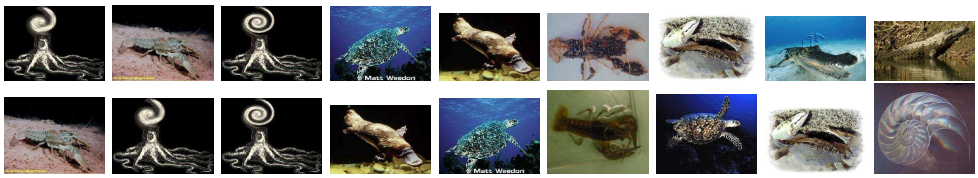
Le système présenté ici peut faire l'objet de multiples améliorations techniques : parallélisation des calculs, partage de calculs communs. Au niveau algorithmique, l'approche descendante est a priori plus efficace, cependant nous avons vu que cela se paie sur les performances. Il serait intéressant de rechercher de nouvelles stratégies d'annotations basées sur l'hypergraphe $\mathcal{H}$, plus efficaces en temps de calcul, sans dégrader fortement les performances globales. Dans cette



(a) *Vivant*

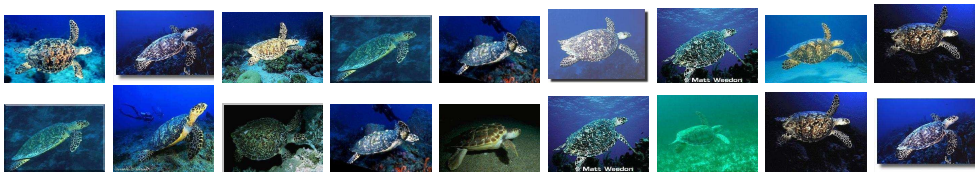
(b) *Extérieur*

(c) *Vivant,Animal,Chordé*



(d) *Vivant,Extérieur,Animaux,Eau*

(e) *Vivant,Extérieur,Animal,Chordé,Reptile*



(f) *Vivant,Extérieur,Animaux,Eau,Caret*

FIGURE 4.29 – *Les 9 premières images retournées pour les requêtes "Hawksbill" (caret) et quelques uns de ses ancêtres. Pour chaque multilabel, la première ligne correspond aux résultats obtenus pour les probabilités non régularisées, la deuxième à ceux obtenus après régularisation. Certaines confusions apparaissent pour les reptiles, pouvant s'expliquer par la présence d'eau (castor), ou de ciel, confondu avec de l'eau (avion), ou par l'absence de pattes (lama assis).*

perspective, la définition de mesures de la fiabilité de prédiction d'un multilabel, caractérisant la confiance que l'on a sur la valeur d'une probabilité, sera une étape fondamentale.

Enfin, l'élaboration de systèmes plus performants passe par une meilleure compréhension des liens entre les catégories, pour en faire une meilleure exploitation. En particulier, il est important de remarquer qu'un lien sémantique ne représente pas nécessairement une communauté de traits visuels entre deux catégories. Cette propriété est d'autant plus marquée que l'on remonte vers des catégories génériques. Par exemple, dans la hiérarchie de Caltech-101, les scorpions et les papillons sont voisins, alors que leurs traits visuels communs sont loins d'être évidents. La définition de la structure du vocabulaire étant par ailleurs sujette à des variations d'une application à l'autre, plusieurs questions mériteraient d'être étudiées :

- Quelle est l'influence d'un changement dans la hiérarchie sur l'annotation à différents niveaux ?
- Quel est la similarité entre la structure visuelle et la structure sémantique du vocabulaire ?
- Fan et al. [61] construisent une ontologie établissant une combinaison des deux. Cette ontologie permet-elle effectivement de faire le lien entre la représentation de l'utilisateur (sémantique) et l'observation (visuelle) ?
- La hiérarchie visuelle n'apportant pas de contenu sémantique, mais étant plus adaptée pour la reconnaissance, comment allier performances et lisibilité pour l'utilisateur ?

En particulier, il serait intéressant d'analyser plus précisément quelle est la fonction de coût la plus intéressante (a priori, celles que nous avons vues présentent toutes plus ou moins les mêmes propriétés), voire d'en définir une spécifique au problème. Ensuite, il faudrait intégrer cette fonction dans le processus d'apprentissage (comme par exemple Binder et al. [22]). Une meilleure exploitation de la fonction de coût nous paraît un aspect fondamental dans les futurs développement de systèmes d'annotation multifacette hiérarchique.

Dans le but de mieux comprendre comment se construit une décision, nous nous sommes intéressés au rôle joué par les caractéristiques aux différents niveaux de la hiérarchie. Dans un système de classification binaire, il est déjà souvent difficile d'interpréter les erreurs de classification. Lorsque la classification devient multiclasse, et de plus hiérarchique, la même caractéristique joue différents rôles, et il devient d'autant plus nécessaire de définir des outils spécifiques. C'est l'objet du chapitre suivant, dont l'objectif est de mener une première étude de cette problématique nouvelle.

5 Sélection hiérarchique de caractéristiques

Pour un utilisateur d'un système d'annotation, il est souvent important de savoir ce qui a amené le système à sélectionner certains mots plus que d'autres. Autrement dit, il est utile d'avoir des éléments permettant de comprendre les choix faits par l'algorithme, en particulier pourquoi il se trompe. Plus concrètement, on aimerait savoir quelles sont les caractéristiques qui influencent la décision. L'objectif de ce chapitre est d'appliquer les méthodes de sélection de caractéristiques pour produire une analyse de l'algorithme d'annotation multifacette hiérarchique. Pour cela, nous avons opéré en deux temps. Nous appuyant sur les méthodes classiques de sélection de caractéristiques, nous les avons d'abord appliquées de manière locale, pour chaque multilabel indépendamment. Nous avons ensuite introduit les contraintes liées à la structure du graphe $\mathcal{H}$ pour faire une sélection de caractéristiques hiérarchique.

Notre principale contribution à ce niveau consiste en l'élaboration d'algorithmes de sélection de caractéristiques pour des classifieurs présentant des dépendances hiérarchiques.

5.1 Introduction

Dans ce chapitre, nous nous intéressons à des méthodes de sélection de caractéristiques dans un contexte d'annotation multifacette hiérarchique. Ces méthodes s'appliquent généralement dans un contexte de classification binaire ou multiclasse, dans le cas où l'espace d'entrée est de très grande dimension. La sélection de caractéristiques comporte alors plusieurs avantages potentiels, pouvant permettre entre autres :

- d'avoir un algorithme plus rapide,
- d'augmenter les performances de prédiction,
- d'analyser les données, d'en avoir une meilleure compréhension.

Bien que les deux premiers nous intéressent, c'est surtout le dernier aspect qui suscite notre intérêt pour la sélection de caractéristiques. En particulier dans le cas où les caractéristiques ont une signification sémantique (comme dans le chapitre 3), l'analyse issue de la sélection de caractéristiques peut fournir des informations importantes susceptibles de guider l'étape de conception des signatures.

La problématique comporte donc plusieurs enjeux : d'une part, comprendre l'influence des différents détecteurs est utile pour expliquer les décisions prises par l'algorithme ; d'autre part, cela permettrait d'adapter l'algorithme en amont, par exemple en limitant les calculs à l'information utile, ou en optimisant l'extraction à ce niveau. L'influence peut être simplement évaluée en effectuant un classement des caractéristiques ; la réduction des calculs suppose d'en sélectionner un

sous-ensemble, ce qui peut amener à une dégradation des performances. Trouver une sélection de variables ne dégradant pas trop les performances permettra de confirmer la faible influence des variables non-sélectionnées.

Dans un premier temps, nous proposons un court état de l'art sur la sélection de caractéristiques. Nous précisons ensuite la problématique, et la façon dont nous l'abordons. Nous présentons les différentes méthodes utilisées sections 5.2 et 5.3, puis les résultats obtenus, dans la section 5.4. Les résultats sont suivis d'une discussion plus générale des enjeux de ce chapitre, section 5.5.

5.1.1 Remarques bibliographiques

On trouve dans la littérature un grand nombre de méthodes permettant de faire de la sélection de caractéristiques dans divers contextes. Notre but ici n'est pas de les recenser exhaustivement, ni d'en concevoir une meilleure, mais de voir comment ces méthodes peuvent être adaptées à un système multifacette hiérarchique. En effet, à notre connaissance, il n'existe pas d'étude de la sélection de caractéristiques dans ce contexte. Nous verrons quelques méthodes de référence, nous intéressant surtout à la manière dont un système de sélection de caractéristiques est conçu. Nous rappelons que l'objectif prioritaire, pour nous, est de donner une explication de la décision, i.e. il ne s'agit pas uniquement de sélectionner les meilleures caractéristiques pour la reconnaissance.

Guyon et Elisseeff [84] proposent une introduction à la sélection de caractéristiques, décrivant les différents objectifs qu'elle permet d'atteindre, et les particularités des différents systèmes.

Soit un vecteur de caractéristiques $x = \{x_1, \ldots, x_i, \ldots, x_d\} \in \mathbb{R}^d$. Blum et Langley [24] définissent deux notions permettant de guider la sélection de caractéristiques. La première est la *pertinence* d'une caractéristique x_i vis-à-vis d'un concept : une caractéristique est pertinente lorsqu'elle est la seule à pouvoir départager deux éléments de deux classes distinctes.

La seconde notion est l'*utilité* de x_i pour un algorithme d'apprentissage (également étudiée par Kohavi et John [108]) : une caractéristique est utile si elle permet un meilleur apprentissage. Contrairement à la pertinence, cette notion est liée à un algorithme d'apprentissage, et non aux propriétés intrinsèques des données.

On groupe généralement les algorithmes de sélection de caractéristiques en trois familles :

1. les filtres (*filters*), consistant à sélectionner les caractéristiques avant la classification, de manière indépendante du classifieur,
2. les wrappers, qui optimisent conjointement la sélection de caractéristiques et la classification,
3. les méthodes inclusives (*embedded*), lorsque le classifieur permet naturellement de sélectionner des caractéristiques.

La première famille est plutôt liée à la notion de pertinence des caractéristiques, les deux suivantes à la notion d'utilité. Pour optimiser la définition des signatures, c'est la notion de pertinence qui nous intéressera. Nous étudierons donc en priorité les méthodes par filtres.

5.1.2 Problématique

Dans ce chapitre, on considère un système d'annotation d'images dans lequel une image $\mathcal{I}$ est représentée par une signature $x \in \mathbb{R}^d$, et annotée par un multi-label $\mathbf{t}$. Le système utilise x pour calculer une probabilité $p(\mathbf{t} \prec \mathbf{y}|x)$ pour chaque

multilabel $\mathbf{y} \in \mathcal{H}$. Le but de l'étude qui suit est de déterminer comment les différentes composantes de x influencent dans le choix des multilabels, et comment cette influence est liée aux liens entre multilabels. Pour un multilabel donné $\mathbf{y}$, on cherche en fait à expliquer la valeur $p = p(\mathbf{t} \prec \mathbf{y}|x)$ par rapport aux composantes de x. Par exemple, soit la caractéristique i d'influence maximale, $x_i \gg 0$ pourra expliquer une forte probabilité p, et $x_i \ll 0$ pourra expliquer une faible probabilité.

Formellement, on considère qu'une signature $x = (x_1, \ldots, x_d)$ est issue d'un ensemble de d *détecteurs* notés δ_i, $i \in [d]$, correspondant ou non aux détecteurs du chapitre 3 et appliqués à l'image $\mathcal{I}$ de telle sorte que $\forall i \in [d], x_i = \delta_i(\mathcal{I})$. On appellera *caractéristique* la i-ième composante de x, i.e. la variable correspondant au détecteur δ_i. Le problème consiste à déterminer une fonction $\sigma_\mathbf{y} : \delta_i, i \in [d] \mapsto \sigma_\mathbf{y}(\delta_i) \in \mathbb{R}$ décrivant l'influence d'un détecteur δ_i sur la détection d'un multilabel $\mathbf{y}$. Dans l'idée, la valeur absolue $|\sigma_\mathbf{y}(\delta_i)|$ décrirait la force de l'influence, et son signe indiquerait le type d'influence exercée (en faveur ou au détriment de la détection du multilabel).

Le but de cette étude est double : il s'agit non seulement de calculer $\sigma_\mathbf{y}$ pour tout $\mathbf{y}$, mais de donner des outils pour l'interpréter.

5.1.3 Démarche adoptée

Dans le cas de la base de voiture, que nous étudierons en priorité, les signatures ont par définition un caractère sémantique, qui peut être directement exploité pour sélectionner des caractéristiques de manière ad hoc. En pratique, cette sélection pourra servir de référence, en particulier pour interpréter les autres résultats de sélection de caractéristiques. Par ailleurs, les caractéristiques ne sont clairement pas indépendantes entre elles : de par leur définition, certaines sont liées par co-occurence (même modèle) ou par exclusion (même type de détail). De plus, les détails sont définis pour les modèles, i.e. pour les feuilles : pour un multilabel donné, il n'y a donc pas toujours un seul type de détail associé. Il pourra être intéressant d'étendre la signature, par exemple par le produit des caractéristiques, pour découvrir des combinaisons de détails appropriées. Enfin, les méthodes permettant un classement de caractéristiques s'offrent comme une solution naturelle, en ce qu'elles permettent une évaluation des caractéristiques de manière individuelle et les unes par rapport aux autres.

Nous adopterons une démarche en deux temps. Pour commencer, nous étudierons l'algorithme de manière locale, c'est-à-dire que nous observerons l'influence des différentes caractéristiques sur chaque nœud séparément (section 5.2). Nous introduirons ensuite des méthodes de sélection hiérarchiques, de manière à étudier plus particulièrement les liens entre les différents nœuds, i.e. comment des variations d'une signature peuvent permettre d'ajouter un label à un multilabel (section 5.3). Les caractéristiques sélectionnées dans chaque cas pourront être comparées aux caractéristiques sémantiques.

5.2 Sélection de caractéristiques – étude locale

5.2.1 Algorithmes de sélection de caractéristiques

Notre étude sera basée sur plusieurs algorithmes classiques de sélection de caractéristiques, permettant d'établir un classement des caractéristiques nœud par nœud. Ainsi, pour un multilabel $\mathbf{y}$ et pour une caractéristique δ_i, $\sigma_\mathbf{y}(\delta_i)$ pourra représenter :

1. la valeur absolue du coefficient de corrélation de Pearson, ainsi qu'une version itérative de celui-ci [85],
2. l'information mutuelle conditionnelle, en binarisant les caractéristiques par seuillage par rapport à zéro [72],
3. le coefficient associé à x_i dans la fonction de classification d'un SVM linéaire,
4. un indicateur sémantique, déterminant si la caractéristique peut exister pour ce multilabel.

Soient les détecteurs $\delta_1, \dots, \delta_d$, et la base d'apprentissage $(x_i, \mathbf{t}_i) \in \mathbb{R}^d \times \{0,1\}^{N_m}$, où $\forall i \in [d]$, $x_i = (\delta_1(\mathcal{I}_i), \dots, \delta_d(\mathcal{I}_i)) = (x_{i1}, \dots, x_{id})$. Pour un multilabel donné $\mathbf{y}_j \in \mathcal{H}$, on utilisera, comme au chapitre précédent, la notation $z_i^{(j)} = +1$ si $\mathbf{t}_i \in \mathbf{y}_j$, et -1 sinon. Soit $X_l = (x_{1l}, \dots, x_{nl})^\mathsf{T}$ et $Z_j = (z_1^{(j)}, \dots, z_n^{(j)})$.

Nous rappelons rapidement les définitions des différentes méthodes de sélection de caractéristiques. Le coefficient de corrélation de Pearson est défini par :

$$\sigma_{\mathbf{y}_j}^{\text{CORR}}(\delta_l) = \frac{\langle X_l - \bar{X}_l, Z_j - \bar{Z}_j \rangle}{\sqrt{(X_l - \bar{X}_l)^2 (Z_j - \bar{Z}_j)^2}}, \tag{5.1}$$

en notant $\bar{X}_l = \frac{1}{n}\sum_{i=1}^n x_{il}$ et $\bar{Z}_j = \frac{1}{n}\sum_{i=1}^n z_i^{(j)}$.

Nous proposons également de calculer ce coefficient de manière itérative. Dans cette méthode, notée SCORR, la corrélation est calculée par rapport à $Z_j^t \in \mathbb{R}$ dépendant des caractéristiques déjà sélectionnées à l'étape t. Dans ce qui suit, on suppose que les données X_l et Z_j sont centrées.

À chaque étape t, on met à jour Z_j^t selon l'équation :

$$Z_j^t = Z_j^{t-1} - \alpha_{t-1} X_{l_{t-1}}, \tag{5.2}$$

où

$$\alpha_t = \frac{\langle X_{l_t}, Z_j^t \rangle}{\langle X_{l_t}, X_{l_t} \rangle}. \tag{5.3}$$

On calcule les coefficients de corrélation par rapport à Z_j^t, et on sélectionne la caractéristique δ_{l_t} pour laquelle le coefficient est maximal. On en déduit :

$$\sigma_{\mathbf{y}_j}^{\text{SCORR}}(\delta_{l_t}) = \frac{\langle X_{l_t}, Z_j^t \rangle}{\sqrt{(X_{l_t})^2 (Z_j^t)^2}}, \tag{5.4}$$

L'information mutuelle conditionnelle est une notion issue de la théorie de l'information, dont on trouvera une bonne introduction au chapitre 2 de Cover et Thomas [39]. Dans la méthode de Fleuret [72], les variables sont sélectionnées de manière itérative, et l'information mutuelle est donc calculée conditionnellement à toutes les variables déjà sélectionnées. Soit $\mathcal{X}_l^{\mathbf{y}_j}$ l'ensemble des variables déjà sélectionnées, le coefficient associé à δ_l est :

$$\sigma_{\mathbf{y}_j}^{\text{IMC}}(\delta_l) = \min_{X_k \in \mathcal{X}_l^{\mathbf{y}_j}} I(Z_j; X_l | X_k). \tag{5.5}$$

La fonction de classification du classifieur SVM linéaire au nœud $\mathbf{y}_j$ vaut $\langle w, x \rangle$, où $w \in \mathbb{R}^d$, et la définition de la fonction de classement associée est donc directe : $\forall l \in [d], \sigma_{\mathbf{y}_j}^{\text{SVM}}(\delta_l) = w_l$.

Enfin, l'indicateur sémantique est défini (pour la base de voitures uniquement) par :

$$\sigma^{\text{SEM}}_{\mathbf{y}_j}(\delta_l) = \begin{cases} 1 & \text{si } model(\delta_l) \prec \mathbf{y}_j \\ 0 & \text{sinon,} \end{cases} \tag{5.6}$$

en notant $model(\delta_l)$ le multilabel minimal correspondant à l'annotation du détecteur δ_l. Dans le cas des SPM, pour Caltech-101 par exemple, nous n'avons pas d'information sémantique sur les caractéristiques.

5.2.2 Interprétation du classement

Dans un premier temps, nous proposons de faire une étude locale de la sélection de caractéristiques. Chaque algorithme est donc appliqué de manière indépendante pour chaque classifieur, selon le principe classique, et donne une fonction de classement $\sigma_{\mathbf{y}}$ pour chaque nœud. Le classement donné par $\sigma^{\text{SEM}}_{\mathbf{y}}$ sert de référence. Notre souci étant d'examiner la valeur explicative de $\sigma_{\mathbf{y}}$ pour la valeur $p(\mathbf{t} \prec \mathbf{y}|x)$, nous observons la corrélation de ces deux valeurs dans chaque cas, ainsi que l'écart à la moyenne. Les résultats sont rapportés dans la section 5.4.

5.2.3 Interprétation avec extensions des signatures

Dans le cas de la base de voitures, l'inconvénient de la méthode proposée précédemment est qu'elle se base sur la signature définie à partir de détecteurs δ_l tels que $model(\delta_l)$ est toujours une feuille de $\mathcal{H}$. Pour permettre une meilleure estimation aux différents niveaux de l'hypergraphe, nous proposons d'étendre cette signature en combinant des détecteurs, selon les combinaisons de labels consistants définies par $\mathcal{H}$. On note $type(\delta_l)$ le type de détail détecté par le détecteur δ_l (par exemple, *logo* ou *rétroviseur gauche*). Soit un multilabel $\mathbf{y}$. Soit $feuilles(\mathbf{y})$ l'ensemble des nœuds feuilles descendant de $\mathbf{y}$ dans l'hypergraphe $\mathcal{H}$. Soit $\Delta(type, \mathbf{y}) = \{\delta_l | type(\delta_l) = type \text{ et } model(\delta_l) \in feuilles(\mathbf{y})\}$. On construit un nouveau détecteur $\delta(type, \mathbf{y}) = comb(\Delta(type, \mathbf{y}))$ où $comb$ peut être défini de plusieurs manières :

- $comb(\Delta) = \max_{\delta \in \Delta} \delta$,
- $comb(\Delta) = \min_{\delta \in \Delta} \delta$,

La combinaison correspondant à la réalité, sémantiquement, est la disjonction logique, modélisée par le maximum. En pratique, cependant, il se trouve que certaines caractéristiques sont identiques entre modèles, et que la conjonction logique est la combinaison adpatée. Par exemple, les différentes phases d'un même modèle n'ont que peu de différences (par exemple, ClioII-1 et ClioII-2 diffèrent essentiellement par la forme des phares).

5.3 Sélection hiérarchique de caractéristiques

Dans la section précédente, nous avons appliqué des algorithmes de sélection de caractéristiques de manière indépendante sur les différents multilabels. Lorsque l'on annote une image, et qu'en jouant sur le seuil de fiabilité, on passe d'un nœud à un autre, il serait intéressant de s'assurer qu'il y a une cohérence entre ces deux nœuds. En effet, entre deux nœuds voisins, on s'attend à ce qu'il y ait un certain nombre de caractéristiques communes – que les objets se ressemblent. A l'opposé, pour deux nœuds différents (ou plus exactement, deux

consets différents), il faut éviter d'avoir exactement les mêmes ensembles de caractéristiques : pour pouvoir distinguer entre les catégories, il faut au moins une caractéristique pertinente.

Tandis que nous nous intéressions tout-à-l'heure à classer les caractéristiques, le problème que nous étudions dans cette partie correspond plus exactement à de la sélection de caractéristiques : il s'agit de sélectionner $K < d$ caractéristiques représentatives de l'ensemble, pour chaque nœud, en intégrant des conditions liées à leur voisinage.

La hiérarchie est donc introduite sous la forme des contraintes suivantes :

1. deux nœuds voisins (parents et fils) doivent partager des caractéristiques,
2. deux nœuds voisins doivent avoir des caractéristiques différentes,

qui seront implémentées sous la forme d'objectifs.

Nous utiliserons les notations suivantes : $\mathcal{M}_K(\mathbf{y}) \subset [d]$ désignera l'ensemble des K caractéristiques sélectionnées avec la méthode locale $\mathcal{M} \in \{\text{CORR,IMC,SVM...}\}$ au nœud $\mathbf{y}$, représentées par leurs indices ; par exemple, $\mathcal{M}_K(\mathbf{y}) = \{3, 6, 9\}$ si les caractéristiques sélectionnées sont δ_3, δ_6, δ_9. $\mathcal{M}^h_K(\mathbf{y})$ désignera l'ensemble des caractéristiques sélectionnés par la méthode hiérarchique.

5.3.1 Méthodes ascendantes

Le principe est le suivant : la sélection de caractéristiques est d'abord effectuée sur les feuilles, et conditionne la sélection sur les parents, en remontant récursivement.

Soit $\mathcal{M}_K(\mathbf{y})$ l'ensemble des K caractéristiques sélectionnées pour un nœud $\mathbf{y}$ par la méthode $\mathcal{M}$. Si $\mathbf{y}$ est une feuille, i.e. $\mathbf{y} \in \hat{\mathcal{H}}$, $\mathcal{M}^h_K(\mathbf{y}) = \mathcal{M}_K(\mathbf{y})$. Sinon, soit $\mu(\mathbf{y}) = \bigcup_{\mathbf{y}_j \in fils(\mathbf{y})} \mathcal{M}^h_K(\mathbf{y}_j)$. La sélection de caractéristiques est alors effectuée sur $\mathbf{y}$, avec la même méthode $\mathcal{M}$, en se restreignant aux caractéristiques dans $\mu(\mathbf{y})$, donnant directement $\mathcal{M}^h_K(\mathbf{y})$.

Cet algorithme permet de s'assurer que les caractéristiques d'un nœud sont proches de celles de ses voisins (parents, fils), mais de manière globale : elle n'empêche nullement que toutes les caractéristiques proviennent d'un même fils. Pareillement, la contrainte 2 n'est pas forcément respectée. Cependant, en pratique, cet algorithme tend à se rapprocher des contraintes.

Une adaptation toute simple permet de remplir à coup sûr le deuxième objectif : il suffit de récupérer un sous-ensemble de $\mathcal{M}^h_K(\mathbf{y})$, obtenu comme précédemment, i.e. $\mathcal{M}_{K'}(\mathbf{y}) \subset \mu(\mathbf{y})$, avec $K' < K$, puis d'appliquer de nouveau la sélection de caractéristiques sur les éléments restants : $\mathcal{M}_K^{(\mathcal{M}_{K-K'})}(\mathbf{y}) \subset [d] \backslash \mu(\mathbf{y})$. Alors $\mathcal{M}^h_K(\mathbf{y}) = \mathcal{M}_{K'}(\mathbf{y}) \cup \mathcal{M}_K^{(\mathcal{M}_{K-K'})}(\mathbf{y})$. Ainsi, au minimum $K - K'$ caractéristiques sont différentes entre $\mathbf{y}$ et ses fils. Cependant, de la même manière que précédemment, le nombre minimum de caractéristiques partagées K_α peut être nul, si l'un des fils ne donne aucune caractéristique au parent : $\exists \mathbf{y}_j \in fils(\mathbf{y}), \mathcal{M}_{K'}(\mathbf{y}) \cap \mathcal{M}^h_K(\mathbf{y}_j) = \emptyset$. Nous nommons ces deux méthodes H1 et H2 dans les expériences.

5.3.2 Méthode globale

5.3.2.1 Principe

L'idée est de minimiser un critère global de manière à s'approcher des contraintes. Ce critère est exprimé sous la forme d'une fonction de coût. Une solution globale est notée $\tau = (\mathcal{M}(\mathbf{y}_1), \ldots, \mathcal{M}(\mathbf{y}_{N_m}))$. Chaque nœud contribue au

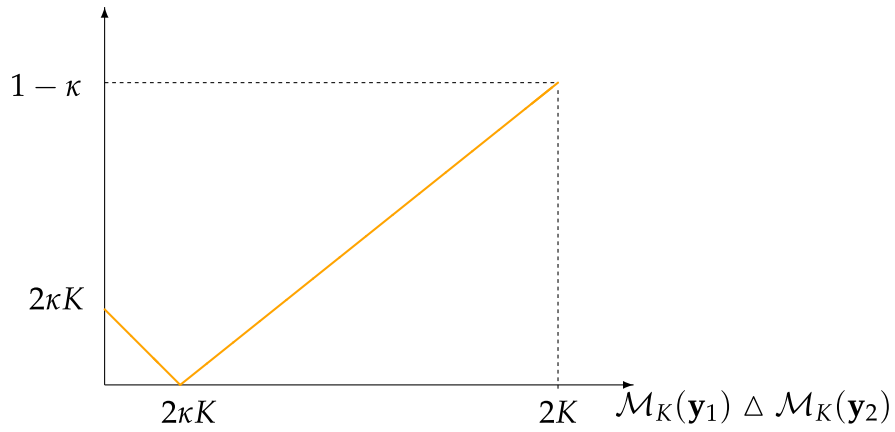


FIGURE 5.1 – *Forme de la fonction mesurant le coût dû aux variations entre les ensembles de caractéristiques* $f_h(\tau)$.

coût selon une fonction dépendant de son voisinage : les caractéristiques sélectionnées au nœud $\mathbf{y}$ doivent être en partie différentes, et en partie partagées, par les nœuds qui lui sont liés, c'est-à-dire les nœuds de $par(\mathbf{y})$ et de $fils(\mathbf{y})$. Le coût total dû aux variations de $\mathcal{M}_K$ peut donc être représenté par une somme de coûts locaux, définie par

$$f_1(\mathcal{M}_K(\mathbf{y})) = \frac{1}{\xi_{\mathbf{y}}} \sum_{\mathbf{y} \in fils(\mathbf{y}_j) \vee \mathbf{y} \in par(\mathbf{y}_j)} f_2(\mathcal{M}_K(\mathbf{y}_j), \mathcal{M}_K(\mathbf{y})), \tag{5.7}$$

où $\xi_{\mathbf{y}}$ est un facteur de normalisation correspondant à une erreur maximale, et $f_2(\mathcal{M}_K(\mathbf{y}_1), \mathcal{M}_K(\mathbf{y}_2))$ représente le coût d'une différence entre les caractéristiques sélectionnées pour les nœuds voisins $\mathbf{y}_1$ et $\mathbf{y}_2$. Cette fonction est définie de manière à pénaliser à la fois les différences trop grandes et trop petites, exprimées à partir de la différence symétrique $A \triangle B$ entre les ensembles de caractéristiques A et B. L'importance des "petites" variations (i.e. $\mathcal{M}_K(\mathbf{y}_1) \triangle \mathcal{M}_K(\mathbf{y}_2)$ proche de zéro) est contrôlée par un facteur $\kappa \in [0, 1[$. Typiquement, on veut surtout éviter d'avoir des ensembles complètement différents, et on pénalise plus une trop grande différence qu'une trop grande similarité. La figure 5.1 présente la forme que prend cette fonction. Elle est normalisée de manière à valoir 1 lorsque la différence $\mathcal{M}_K(\mathbf{y}_1) \triangle \mathcal{M}_K(\mathbf{y}_2)$ est maximale ($= 2K$).

$$f_2(\mathcal{M}_K(\mathbf{y}_1), \mathcal{M}_K(\mathbf{y}_2)) = |\mathcal{M}_K(\mathbf{y}_1) \triangle \mathcal{M}_K(\mathbf{y}_2) - 2\kappa K|. \tag{5.8}$$

La fonction de coût globale f vaut alors :

$$f(\tau) = \frac{1}{N_m} \sum_{j=1}^{N_m} f_1(\mathcal{M}_K(\mathbf{y}_j)). \tag{5.9}$$

Minimiser uniquement ce coût comporte le risque de s'écarter de manière incontrôlée de la sélection initiale. Nous proposons de conserver le lien avec les données en introduisant une fonction de coût dépendant de l'erreur de généralisation des classifieurs, calculée par validation croisée :

$$f_d(\tau) = \sum_{j=1}^{N_m} \varepsilon(\mathbf{y}_j, \mathcal{M}_K(\mathbf{y}_j)), \tag{5.10}$$

où $\varepsilon(\mathbf{y}_j, \mathcal{M}_K(\mathbf{y}_j))$ est l'erreur de validation croisée obtenue pour le classifieur au nœud $\mathbf{y}_j$, en utilisant seulement les caractéristiques $\mathcal{M}_K(\mathbf{y}_j)$.

Renommant f_h la fonction de coût définie par l'équation (5.9), décrivant l'attache à la hiérarchie, nous utiliserons finalement une fonction de coût définie par :

$$f(\tau) = \alpha \cdot f_h(\tau) + (1 - \alpha) \cdot f_d(\tau), \tag{5.11}$$

où $\alpha \in [0, 1]$ est un facteur de contrôle permettant de définir l'importance relative de la hiérarchie ou de l'erreur de classification.

f n'est pas une fonction convexe et ne peut donc pas être optimisée de manière analytique. Nous avons donc opté pour une optimisation par recuit simulé.

5.3.2.2 Recuit simulé : présentation

Introduit indépendamment dans les années 80 par Kirkpatrick et al. [107] et Černỳ [33], le recuit simulé (en anglais, *simulated annealing*) a été utilisé pour résoudre une grande variété de problèmes d'optimisation non-convexes. Mathématiquement, il peut être modélisé en s'appuyant sur la théorie des chaînes de Markov finies. La méthode étant classique, nous décrirons son principe de manière fonctionnelle. Pour plus de détails, et notamment pour la preuve de convergence et la discussion des paramètres, nous renvoyons le lecteur à Laarhoven et Aarts [110] ou Aarts et al. [1].

Le recuit simulé vise à minimiser une fonction de coût f par un processus de Monte-Carlo. Le principe est décrit par l'algorithme 1. La fonction INITIALISE() consiste simplement à choisir la configuration de départ. La fonction NON_STOP() permet de tester si le critère de convergence est atteint ou non. A chaque étape de l'algorithme, une nouvelle configuration est sélectionnée parmi les configurations voisines (i.e. les transitions possibles) grâce à la fonction ALTERE(). La nouvelle configuration est acceptée avec une probabilité 1 si le coût diminue, et avec une probabilité non-nulle sinon. La probabilité d'acceptation dépend de la variation du coût $f(S_t) - f(s)$, ainsi que d'un paramètre de contrôle T, également appelé température. La fonction TEMPERE() permet de gérer l'évolution de la température au cours du temps : alors qu'elle diminue, la probabilité d'accepter des configurations augmentant le coût est de plus en plus faible. La vitesse de convergence de l'algorithme dépend de la vitesse de décroissance de T, mais plus T diminue lentement, plus il y a de chances de s'approcher du minimum global.

Algorithme 1: Recuit simulé

Entrées : fonction de coût f
Sorties : configuration optimale S
$t = 0$;
$S_0 =$ INITIALISE() ;
T_0 ;
tant que NON_STOP() **faire**
  $s =$ ALTERE(S_t) ;
  si random$(0, 1) < e^{\frac{f(S_t) - f(s)}{T_t}}$ **alors**
    $S_{t+1} = s$;
  sinon
    $S_{t+1} = S_t$;
  $T_{t+1} =$ TEMPERE();
  $t = t + 1$;

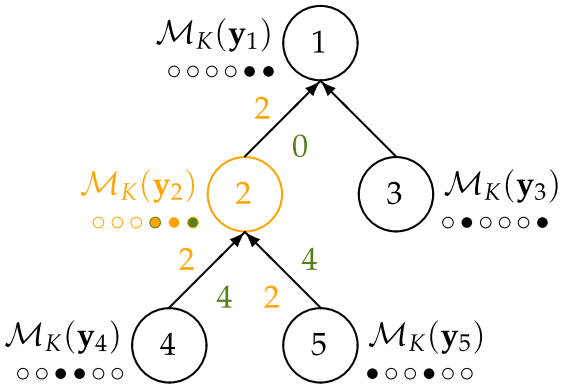


FIGURE 5.2 – *Exemple de transition sur un graphe composé de 5 nœuds, avec $d = 6$ et $K = 2$. La configuration est modifiée sur le nœud 2 : l'ancienne est en orange, la nouvelle en marron. Il se trouve que la nouvelle configuration est telle que $\mathcal{M}_K(\mathbf{y}_2) = \mathcal{M}_K(\mathbf{y}_1)$. Cette modification entraîne une augmentation du coût $f_h(\mathcal{M}_K(\mathbf{y}))$, que l'on observe facilement en considérant la contribution de chaque voisin avant et après, indiquées respectivement en orange et en marron.*

5.3.2.3 Recuit simulé : application à la sélection de caractéristiques hiérarchique

Nous avons vu dans la section 5.3.2.1 comment la fonction de coût a été définie de manière à sélectionner des caractéristiques cohérentes sans trop dégrader le taux de classification. Ainsi, nous utilisons le recuit simulé pour estimer l'ensemble de caractéristiques à sélectionner en chaque nœud pour minimiser la fonction de coût définie équation 5.11. Une configuration correspond à un ensemble de solutions τ. Pour l'initialisation, on applique la sélection de caractéristiques sur chaque nœud $\mathbf{y}$, donnant $\sigma_\mathbf{y}$ et $\mathcal{M}_K(\mathbf{y})$. Le coût associé est calculé pour chaque nœud. Une transition se fera en sélectionnant un nœud $\mathbf{y}$, et en remplaçant une caractéristique $\delta_i \in \mathcal{M}_K(\mathbf{y})$ par une autre $\delta_j \notin \mathcal{M}_K(\mathbf{y})$. La figure 5.2 donne de transition et la variation du coût associée. On peut également orienter la transition en introduisant une caractéristique déjà présente chez un voisin, tout en en retirant une qui est absente chez ce même voisin. Pour calculer le coût associé à la nouvelle configuration, il suffit de calculer le coût pour le nœud auquel la modification a été appliquée. Cela nécessite d'entraîner à nouveau le classifieur au nœud $\mathbf{y}$, et d'estimer l'erreur de généralisation par validation croisée. L'algorithme s'arrête lorsqu'un régime stationnaire est atteint, ou lorsque le nombre d'itérations maximal est dépassé. On peut également s'arrêter lorsque l'on parvient à un minimum fixé d'avance.

En pratique, pour pouvoir tester sur un bloc de données $\mathcal{F}_i$, $i \in [4]$, on applique la sélection de caractéristiques indépendamment pour chaque ensemble $\mathcal{L}_b \backslash \mathcal{F}_i$ de la partition de $\mathcal{L}_b$ (voir section 4.5). Or, l'erreur de généralisation est estimée pour l'optimisation des classifieurs par validation croisée sur ce même ensemble $\mathcal{L}_b \backslash \mathcal{F}_i$. Il y a donc un risque de sur-apprentissage. Ce risque peut être limité soit en réduisant l'importance de l'erreur de généralisation dans la fonction de coût (α proche de 1), soit en lui imposant de rester au-dessus d'un certain seuil, donc en pénalisant des erreurs trop petites dans la définition de $f_d(\tau)$. Dans l'équation (5.11), on remplacera donc f_d par f_e, définie de la manière suivante :

$$f_e(\tau) = \begin{cases} f_d(\tau) - f_d(\tau_0) & \text{si } f_d(\tau) - f_d(\tau_0) > 0, \\ \min(\alpha_e \cdot |f_d(\tau) - f_d(\tau_0)|, 1) & \text{sinon,} \end{cases} \tag{5.12}$$

en introduisant α_e un paramètre permettant de pénaliser plus ou moins fortement un écart au seuil. Après expérience, nous fixons $\alpha_e = 1,2$.

Enfin, nous testons différents modes de refroidissement de la température :

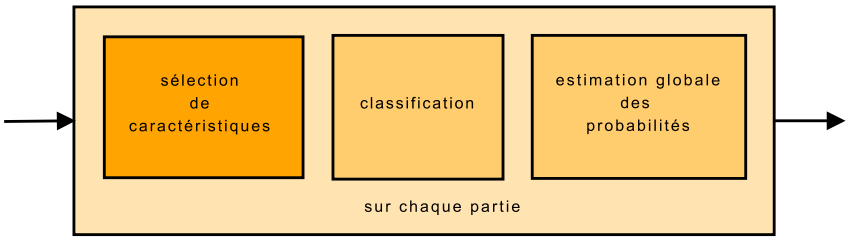


FIGURE 5.3 – *Schéma de principe : sur les données test, positionnement de la sélection de caractéristiques comme une étape préliminaire, et indépendante de la classification.*

$T_{t+1} = \beta T_t$ (avec $\beta = 0.99$ et $\beta = 0.999$), et $T_{t+1} = \frac{\log 1+t}{\log 2+t} \cdot T_t$. En pratique, un refroidissement rapide donne des résultats satisfaisants.

Un des inconvénients de la sélection de caractéristiques hiérarchique, ainsi que nous l'avons conçue, est qu'elle ne permet pas de calculer de fonctions de classements $\sigma_{\mathbf{y}}$ associées aux multilabels $\mathbf{y} \in \mathcal{H}$. Cela peut constituer un frein pour l'analyse de l'algorithme d'annotation. Cependant, il serait possible d'adapter la méthode de recuit simulé pour effectuer des mises à jour de $\sigma_{\mathbf{y}}$.

5.4 Expériences et évaluation

Nous suivons la même procédure expérimentale que dans le chapitre 4, en introduisant la sélection de caractéristiques avant le calcul des probabilités (voir le schéma de principe figure 5.3). Nous ne redécrirons donc pas les expériences. Dans un premier temps, nous observons les caractéristiques sélectionnées, localement et globalement, pour en faire une première évaluation de manière qualitative : le lien avec l'aspect sémantique des détecteurs est-il visible ? comment les relations hiérarchiques se traduisent-elles ? Nous étudions comment la sélection de caractéristiques permet d'interpréter les résultats d'annotation.

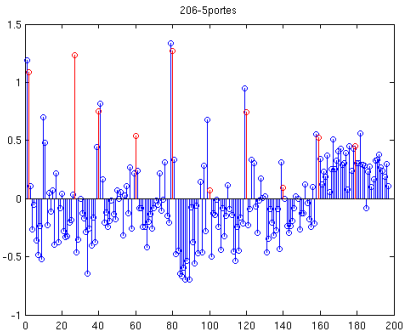
Dans un deuxième temps, nous proposons une analyse plus quantitative de ces résultats, c'est-à-dire en termes de performances, notamment avec les courbes erreur-complexité introduites au chapitre 4.

5.4.1 Analyse qualitative

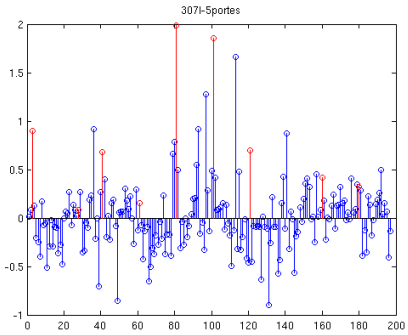
Avant même d'observer les résultats de la sélection de caractéristiques, nous observons simplement les signatures moyennes par catégories, comparées à la signature moyenne globale, i.e. la valeur $\mathbb{E}[X|\mathbf{t}] - \mathbb{E}[X]$. Les résultats sont présentés figure 5.4. Les caractéristiques sémantiques, sélectionnées selon l'équation 5.6, sont représentées en rouge. On constate ainsi que les caractéristiques sémantiques sont, dans la grande majorité des cas, bien représentatives de leur catégorie.

Une première manière d'utiliser la sélection de variables pour l'interprétation consiste à observer directement quelles sont les caractéristiques sélectionnées. Pour faciliter l'interprétation, nous représentons les caractéristiques sous la forme de vignettes représentatives de chacune. La figure 5.5 donne un extrait de la représentation : à chaque nœud de l'hypergraphe $\mathcal{H}$, nous associons les vignettes des caractéristiques sélectionnées, ce qui permet d'observer les similarités éventuelles entre groupes de caractéristiques, par rapport à leur position dans le graphe. En mettant en relation les ensembles sélectionnés pour des nœuds voisins, on peut repérer les caractéristiques les plus discriminatives à chaque niveau.

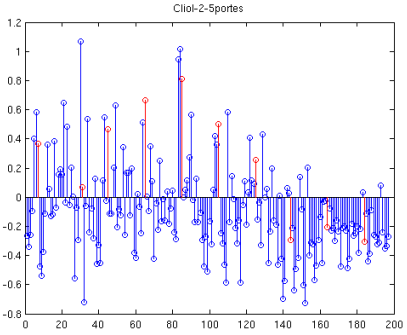
Un des inconvénients de cette représentation est qu'elle ne met pas en évidence le sens de l'influence d'une caractéristique (i.e. le signe de σ_i).



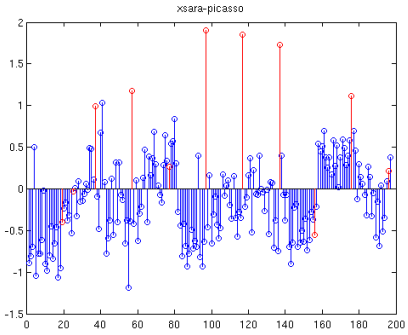
(a) *206, 5 portes*



(b) *307*



(c) *Clio I-2, 5 portes*



(d) *Xsara Picasso*

FIGURE 5.4 – *Ecart entre la signature moyenne de la classe et la signature moyenne globale. En rouge, les caractéristiques sémantiques sélectionnées, c'est-à-dire les caractéristiques dont le modèle correspond à la catégorie.*

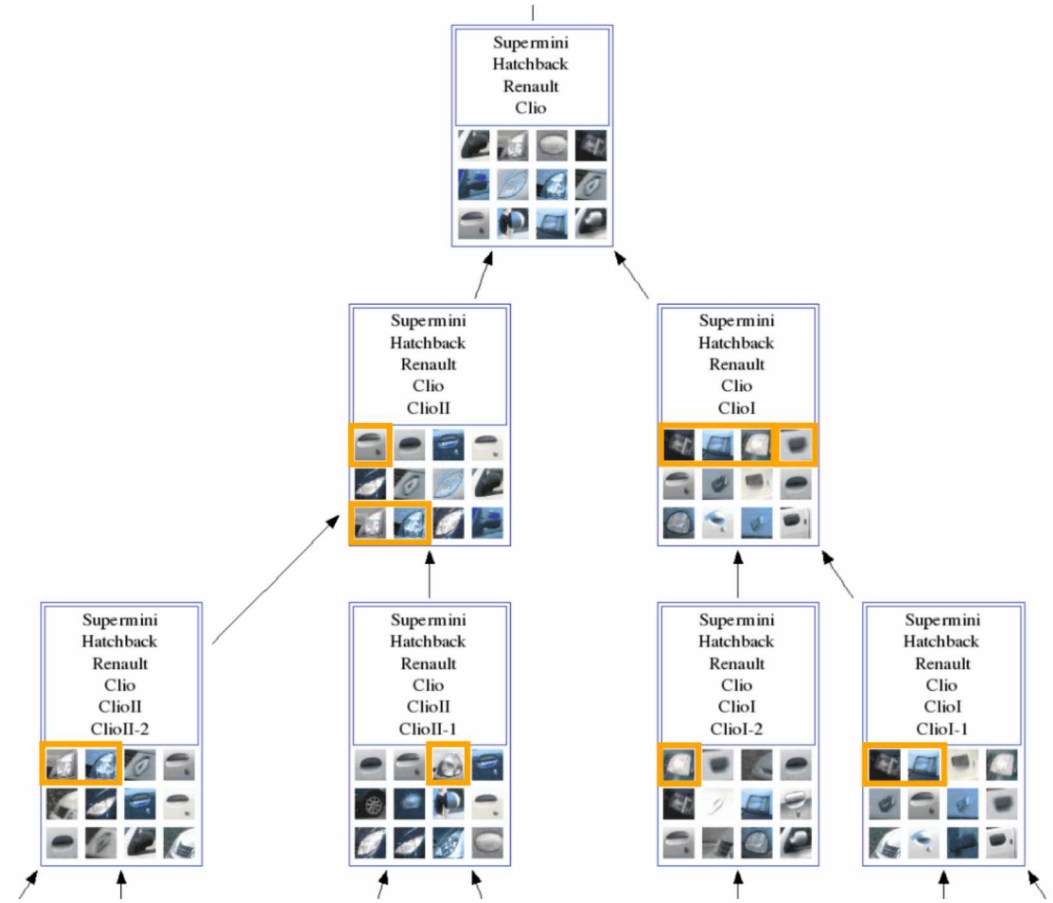


FIGURE 5.5 – *Sélection de caractéristiques ; l'exemple présente les 12 premières caractéristiques sélectionnées par la corrélation itérative pour quelques nœuds liés. Pour chaque détecteur, une imagette représentative est affichée, selon l'ordre donné par la sélection (de gauche à droite et de haut en bas). Nous avons marqué en orange quelques caractéristiques permettant effectivement une bonne discrimination entre modèles voisins selon un jugement cognitif humain : les poignées de portes et les phares avant gauches entre Clio I et Clio II, les phares avant gauches entre Clio I-1 et I-2, et entre Clio II-1 et II-2. Les différences entre différentes phases d'une même version d'un modèle étant en général cantonnées aux alentours de la calandre (phares avant, logo, capot...), il est intéressant de constater que notre algorithme sélectionne des détails situés dans cette zone.*

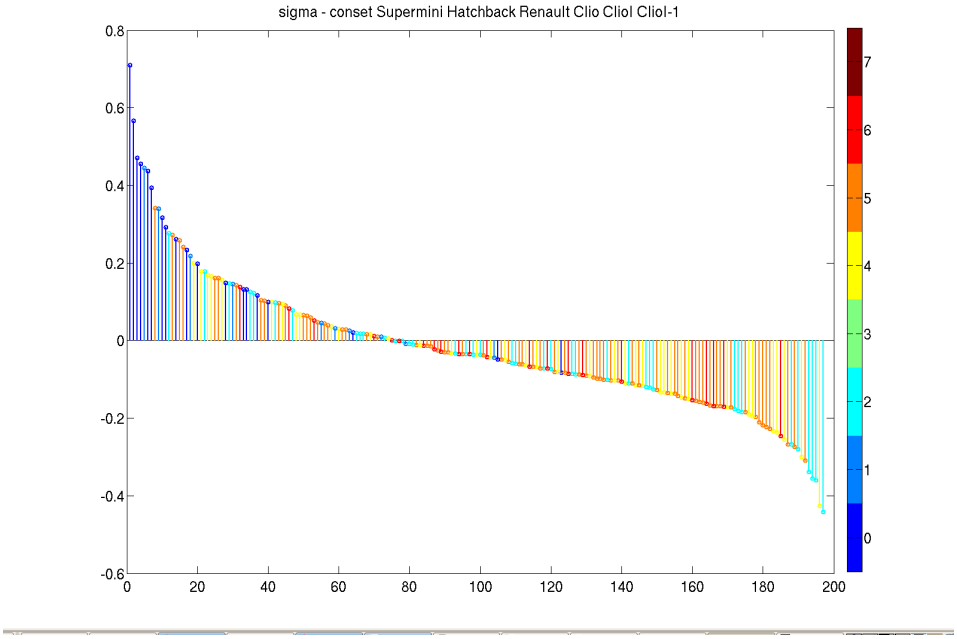


FIGURE 5.6 – *Représentation de $\sigma_{\mathbf{y}}$ pour $\mathbf{y} = \mathbf{y}(Clio1\text{-}1)$. Les couleurs indiquent la distance entre les multilabels, i.e. le nombre de labels différents entre le multilabel $\mathbf{y}$ et celui de chaque détecteur, ou zéro si le multilabel du détecteur est une feuille de $\mathbf{y}$. On constate que les multilabels de σ maximum correspondent à des détecteurs "descendants" de $\mathbf{y}$. On peut également remarquer que les détecteurs influant négativement correspondent plutôt à des voisins proches.*

Nous lui préférerons donc une représentation ordonnée des caractéristiques. La figure 5.6 représente les valeurs de $\sigma_{\mathbf{y}}$ pour chaque détecteur. Une coloration a été ajoutée en fonction de la distance du multilabel du détecteur à $\mathbf{y}$ (ce multilabel étant obtenu à partir de $model(\delta)$). Dans la plupart des cas, on constate que les multilabels de σ maximum correspondent à des détecteurs liés à $\mathbf{y}$. Dans l'exemple de la figure 5.6, comme dans plusieurs autres, on peut également remarquer que les détecteurs influant négativement correspondent plutôt à des voisins proches. Ainsi, l'absence de détecteurs peut aussi avoir une influence sur la décision.

Nous proposons également de visualiser la signature obtenue pour une image donnée, en mettant en évidence les caractéristiques "détectées" (i.e. tel que x_i dépasse un seuil donné) et les valeurs de σ associées. Nous donnons des exemples de détections en comparant l'annotation vérité terrain $\mathbf{t}$ avec une annotation pour un seuil de confiance nul $\mathbf{y}^0$. La figure 5.7 montre une telle représentation dans le cas où l'annotation estimée est correcte. Nous avons fixé un seuil de détection assez élevé (0.1) pour ne pas surcharger l'image. Les figures 5.8,5.9 et 5.10 montrent des exemples de confusions. L'analyse de la signature et des valeurs de $\sigma_{\mathbf{t}}^{\text{SCORR}}$ et $\sigma_{\mathbf{y}^0}^{\text{SCORR}}$ permet de comprendre la source de l'erreur. Dans la section 3.4.2, nous avons distingué entre les fausses alarmes de type I, correspondant à une confusion entre détails similaires, et celles de type II, dans le cas contraire. En utilisant cette distinction, on peut constater plusieurs sources de confusions entre les multilabels :

type 1 : des détecteurs influents correspondent à des fausses alarmes de type II. Dans ce cas-là, la confusion peut avoir un coût hiérarchique important. Par ailleurs, jouer sur le paramètre de confiance peut ne pas être suffisant pour trouver un multilabel acceptable (figure 5.8).

type 2 : certains objets, peuvent posséder quelques détails proches, produisant des

Erreur de type	1	2	3
FRÉQUENCE (%)	**54**	18	33

TABLE 5.1 – *Evaluation des types d'erreur commises par l'algorithme par observation des signatures et des valeurs de σ^{SCORR} (comme dans les figures 5.8,5.9,5.10). La majorité des erreurs est de type 1, c'est-à-dire que la confusion est liée à des fausses alarmes "non-explicables".*

fausses alarmes de type I sans que les objets soient similaires. Par exemple, les phares de 206 et de 307 se ressemblent. Si d'autres détails manquent, la confusion entre ces détails peut gêner la reconnaissance (figure 5.9).

type 3 : certains modèles sont très proches et diffèrent à un détail près (3/5 portes), ou sur des détails très similaires (par exemple, les phares avant de la ClioI-1, vs. ClioI-2). Dans ces cas-là, on trouve de nombreuses fausses alarmes de type I, qui peuvent difficilement être évitées, mais la confusion a un coût hiérarchique mineur, c'est-à-dire que le premier nœud parent constitue en général une annotation correcte (figure 5.10).

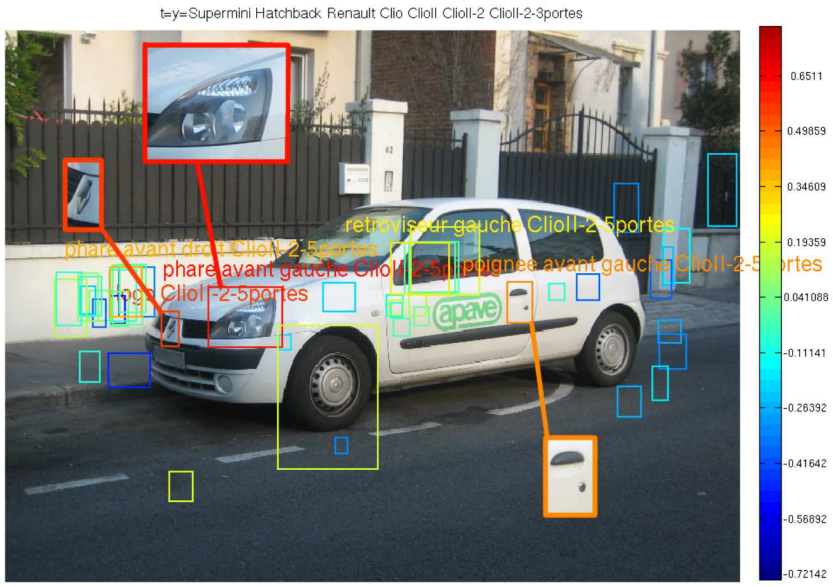
Cette représentation de la signature directement sur l'image a l'avantage d'être assez simple à interpréter (telle bonne détection favorise l'annotation, telle autre provoque une confusion...). Cependant, étant donné le nombre de détecteurs, elle peut être très vite surchargée. L'idéal dans ce cas serait d'avoir une représentation analogue sous forme interactive (permettant de faire varier aisément les seuils contrôlant l'affichage des boîtes englobantes et de leurs labels).

L'analyse de ces erreurs nous permet de définir plusieurs orientations en vue de l'amélioration de la description des images. Les confusions de type 1 peuvent être limitées en apportant des améliorations techniques, de manière à avoir des détecteurs plus fiables (par exemple en agrandissant la base d'apprentissage). Les confusions de type 2 sont liées aux fait que certaines catégories (mêmes éloignées sémantiquement) ont des détails similaires. Par exemple, les C3 et les 307 ont des poignées de portes similaires. Cette propriété peut être intégrée de deux manières : (a) en créant une seule caractéristique à partir des deux, (b) en créant un détecteur permettant de mieux discriminer entre les deux : par exemple, dans un premier temps, détection de la poignée, puis classification C3 ou 307. Ce genre de traitement peut également servir pour réduire les confusions de type 3, lorsque des détails entre deux phases d'un modèles ont subi de légères modifications (par exemple, le phare de la ClioI-1 vers la ClioI-2). Cependant, dans ce cas, il est plus intéressant d'avoir un détecteur (supplémentaire) correspondant à la réunion des deux. En effet, celui-ci est potentiellement plus robuste, et correspond à la catégorie parente, contrairement au cas de type 2, où la caractéristique créée ne correspond à aucun label.

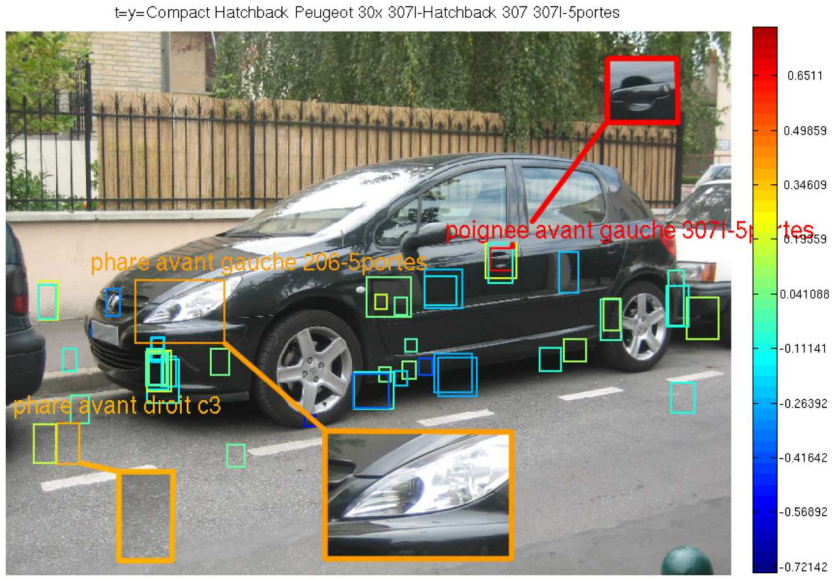
Nous avons évalué grossièrement la répartition des erreurs, sachant que certains cas ambigus peuvent compter pour plusieurs erreurs. Cette répartition, présentée tableau 5.1, montre que le plus gros des erreurs reste dû au manque de fiabilité d'un certain nombre de détecteurs (erreurs de type 1). Il est possible de déterminer les détecteurs à optimiser en priorité en considérant une mesure globale de l'influence, par exemple la somme des influences sur les nœuds :

$$\sigma(\delta) = \sum_{\mathbf{y} \in \mathcal{H}} \sigma_{\mathbf{y}}(\delta). \tag{5.13}$$

Pour éviter de surcharger cette section, les résultats sont rapportés dans l'annexe C.3.



(a)



(b)

FIGURE 5.7 – *Visualisation de la signature par rapport au classement pour le multilabel vérité terrain de chaque image $\sigma_{\mathbf{t}}$ pour des images correctement annotées. La couleur de chaque boîte englobante dépend de l'influence du détecteur δ_i correspondant $\sigma_{\mathbf{y}}(\delta_i)$. Seules les détections pour lesquelles $x_i > 0.1$ sont représentées pour éviter de surcharger l'image et faciliter l'interprétation. Nous indiquons les labels des détecteurs tels que $\sigma_{\mathbf{y}}(\delta_i) > 0.3$. (a) : Exemple où plusieurs détails influents sont correctement détectés (logo, phare avant gauche, poignée avant). (b) : Exemple bien annoté malgré plusieurs fausses alarmes (phare avant droit de C3, phare avant gauche de 206-5 portes), grâce à une bonne détection (poignée avant gauche).*

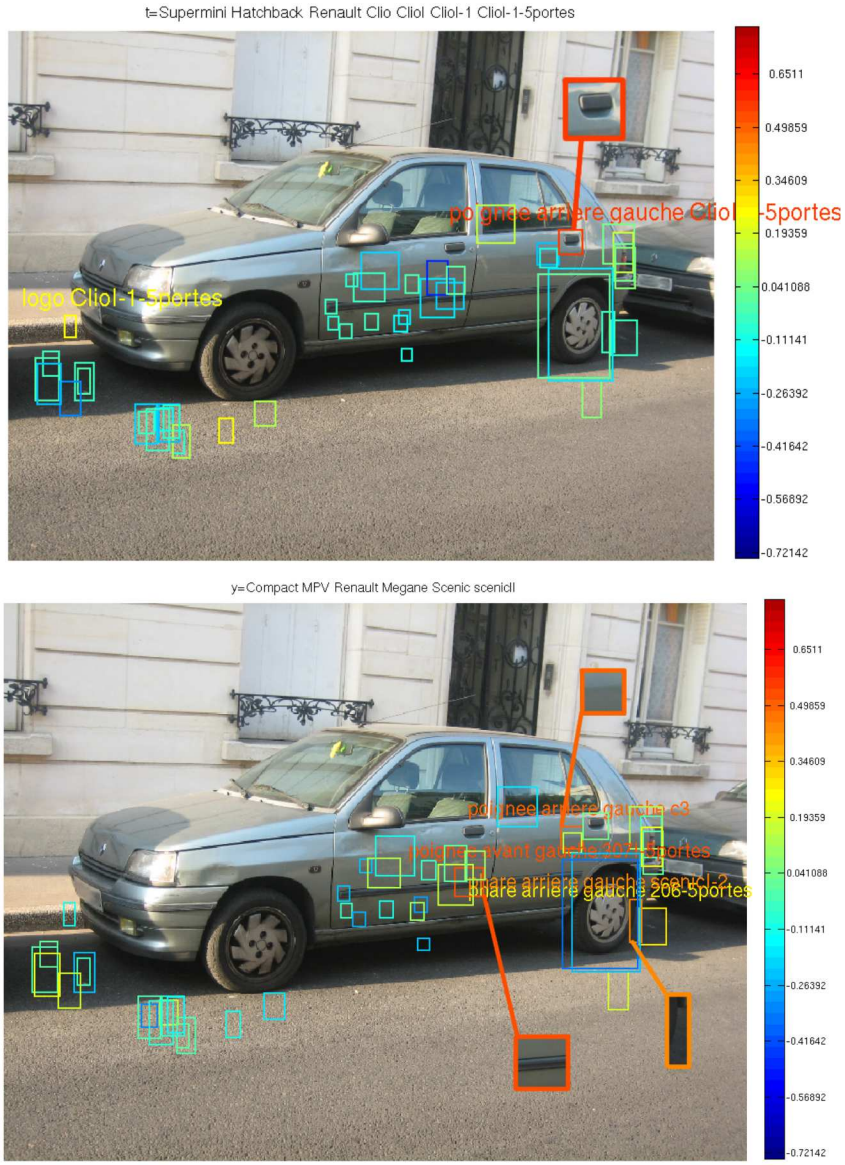


FIGURE 5.8 – *Exemple d'image pour laquelle un trop grand nombre de fausses alarmes empêche l'interprétation. Visualisation de la signature par rapport au classement pour le multilabel vérité terrain de chaque image $\sigma_{\mathbf{t}}$ (en haut) et pour le multilabel estimé $\mathbf{y}^0$ (en bas). Nous indiquons les labels des détecteurs tels que $\sigma_{\mathbf{y}}(\delta_i) > 0.3$. En haut, on remarque une bonne détection, qui devrait permettre une annotation correcte. En bas, différentes fausses alarmes responsables du mauvais label sont mises en valeur.*

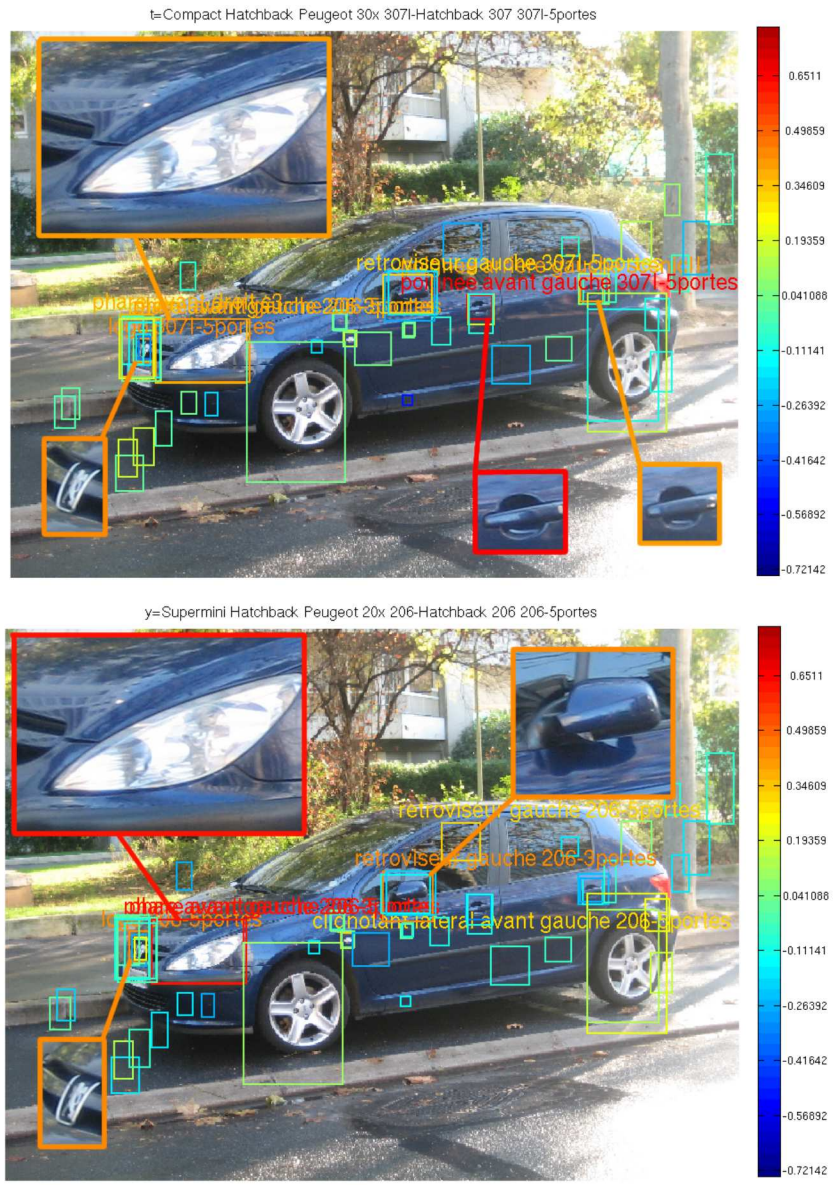


FIGURE 5.9 – *Exemple d'image pour laquelle des détails d'une autre catégorie, proches visuellement, dominent par rapport aux détails différents. Visualisation de la signature par rapport au classement pour le multilabel vérité terrain de chaque image $\sigma_{\mathbf{t}}$ (en haut) et pour le multilabel estimé $\mathbf{y}^0$ (en bas). Nous indiquons les labels des détecteurs tels que $\sigma_{\mathbf{y}}(\delta_i) > 0.3$. Sur la figure du haut, plusieurs bonnes détections sont visibles : logo, poignée avant gauche, rétroviseur de 307. La détection du phare de 206-5 portes a également une influence favorable à l'annotation 307. En bas, on constate que la détection du même phare a une influence plus forte pour le multilabel 206-5 portes. Les fausses détections logo 206-5 portes et rétroviseur 206-3 portes, de type I, jouent également en faveur de ce multilabel, et sont responsable de la confusion.*

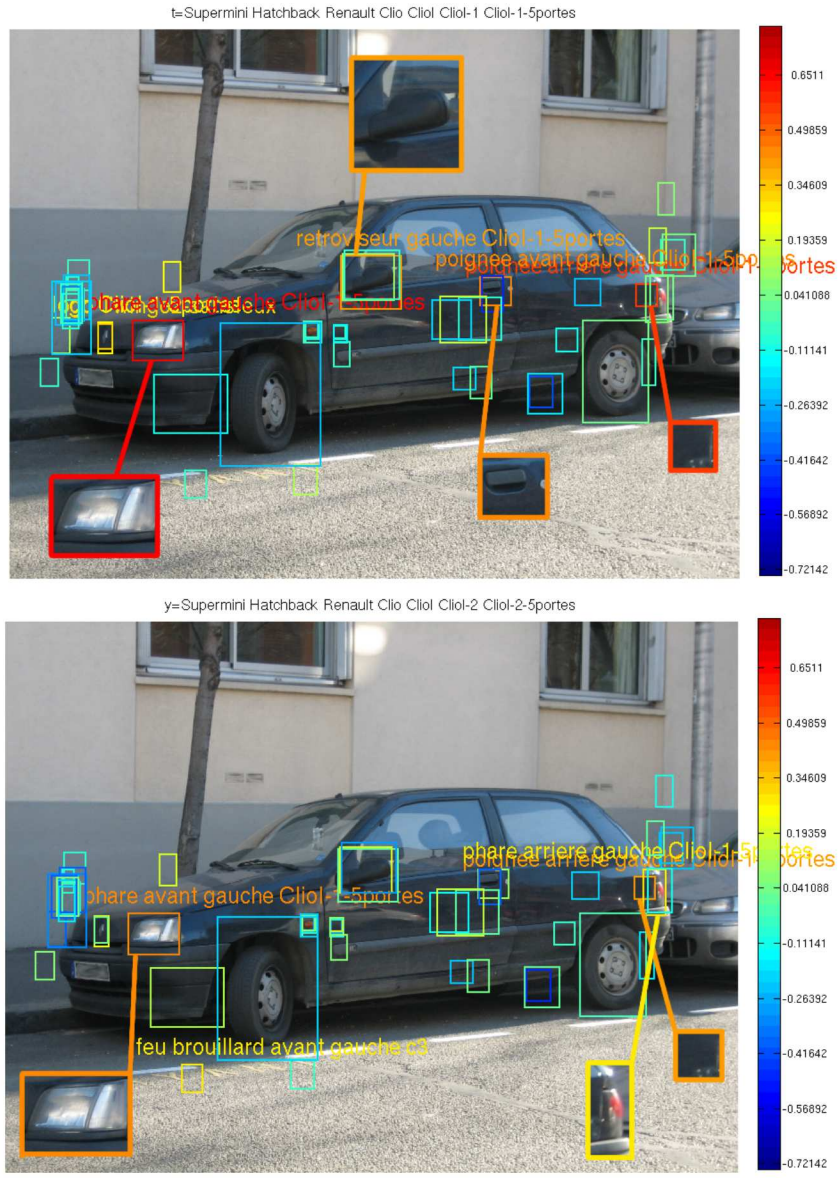


FIGURE 5.10 – *Exemple d'image pour laquelle la confusion est facilement compréhensible : les deux catégories ont un parent commun, et ne diffèrent que par un détail, en l'occurence le phare. Visualisation de la signature par rapport au classement pour le multilabel vérité terrain de chaque image $\sigma_{\mathbf{t}}$ (en haut) et pour le multilabel estimé $\mathbf{y}^0$ (en bas). Nous indiquons les labels des détecteurs tels que $\sigma_{\mathbf{y}}(\delta_i) > 0.3$. En haut, on peut constater que plusieurs détecteurs influents correspondent au bon multilabel, ClioI-1 5 portes (phare avant, poignée avant, rétroviseur), y compris une fausse alarme (poignée arrière). Cependant ces détecteurs sont égalements influents pour l'annotation de ClioI-2 5 portes, en particulier le phare avant. La détection (correcte) du phare arrière a une influence favorable sur le choix du multilabel ClioI-2 5 portes. Les modèles étant très proches, cela prête à confusion.*

Méthode	10	20	30	50	100	toutes
SCORR	92,44	93,32	93,11	**93,32**	93,14	92,52
CORR	**91,59**	91,45	89,22	86,56	90,22	92,52
IMC	90,39	91,36	92,07	**92,56**	92,28	92,52

TABLE 5.2 – *Complémentaire de l'aire sous la courbe erreur-complexité pour différentes valeurs de K et différentes méthodes de sélection, pondérée par une gaussienne $\mathcal{N}(0,5;0,2)$, en %. Pour chaque méthode, nous avons mis en évidence la valeur de K optimale. Seule la méthode par corrélation itérative permet une amélioration des performances.*

5.4.2 Analyse quantitative

Dans la section précédente, nous avons observé l'influence des caractéristiques, avec l'objectif de donner des outils favorisant l'interprétabilité des annotations, correspondant à la notion de *pertinence*. Nous nous intéressons ici à l'*utilité* des caractéristiques, c'est-à-dire à leur capacité à améliorer les performances.

Dans un premier temps, nous évaluons les performances des différentes méthodes de sélection au niveau local. Le tableau 5.2 donne les performances des méthodes CORR, SCORR et IMC pour différentes valeurs de K. Les performances sont évaluées par le score CAUC présenté au paragraphe 4.4.4. Seule la méthode par corrélation itérative (SCORR) et par information mutuelle conditionnelle (IMC) permettent d'améliorer les performances par rapport à la méthode de référence, i.e. sans sélection. Ces deux méthodes ont des performances similaires. Dans les tests de sélection de caractéristiques suivants, nous utiliserons la méthode SCORR avec $K = 20$ comme référence. La valeur $K = 20$ permet de réaliser un compromis entre le nombre de caractéristiques sélectionnées, que l'on souhaite garder assez faible pour une meilleure lisibilité des résultats, et les performances (voir tableau 5.2).

La figure 5.11 représente la courbe erreur/complexité pour les différentes méthodes de sélection de caractéristiques, avec $K = 20$. La méthode sémantique sélectionne un nombre de caractéristiques dépendant du modèle : il se trouve qu'elle n'est pas la meilleure. Cela confirme l'observation faite dans la section précédente selon laquelle l'algorithme d'annotation s'appuie sur certaines caractéristiques "négatives". La sélection par SVM linéaire donne des résultats particulièrement décevants. Ce phénomène est probablement lié à un sur-apprentissage : étant donné le peu de données, la base n'a pas été re-subdivisée pour l'apprentissage. Pour une évaluation correcte de cette méthode, il aurait fallu avoir plus de données à disposition.

La figure 5.12 présente les résultats obtenus pour les différentes méthodes globales. Le tableau 5.3 rapporte ces mêmes résultats sous forme numérique (CAUC). Etant donné la petite taille de la base d'image, le choix de la partition des données a un impact important. Nous avons donc réitéré les expériences plusieurs fois, avec différentes partitions, pour estimer la variance de ces résultats (de la même manière qu'en section 4.5). On constate que les différentes méthodes donnent des résultats similaires.

Jusqu'ici, nous avons travaillé avec des caractéristiques dont la description sémantique se situait au niveau des feuilles. Comme nous l'avons vu au paragraphe 5.2.3, pour avoir véritablement une interprétation au niveau des multilabels, il serait intéressant d'avoir des caractéristiques issues de combinaisons de détecteurs. Les caractéristiques sélectionnées parmi ce jeu étendu seraient plus facilement interprétables. Comme préalable à ces améliorations, nous étudions les performances apportées par les différentes extensions proposées section 5.2.3. Les résultats sont rapportés figure 5.13, toujours sous la forme de courbes erreur-

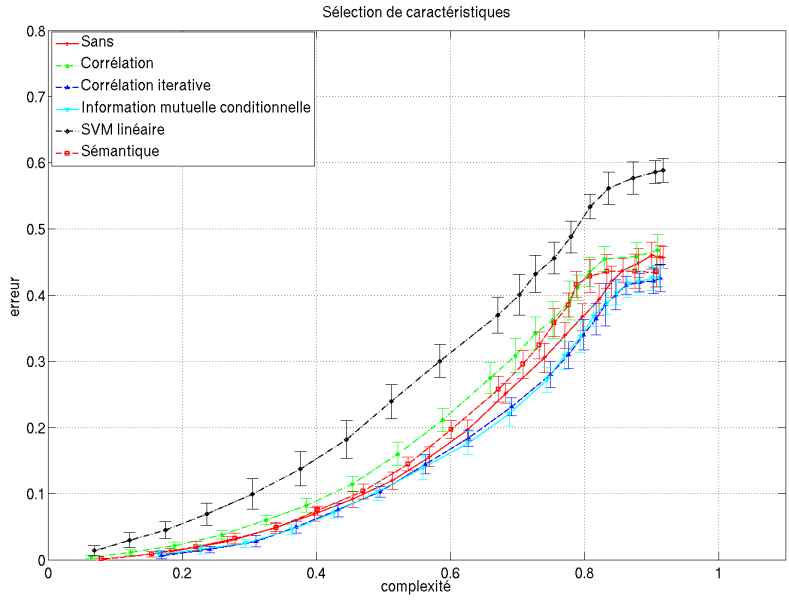


FIGURE 5.11 – *Test des différentes méthodes de sélection de caractéristiques de manière locale. Résultats moyens sur 10 partitions différentes de $\mathcal{L}_b$. Globalement,* SCORR *et* IMC *donnent les meilleurs résultats.*

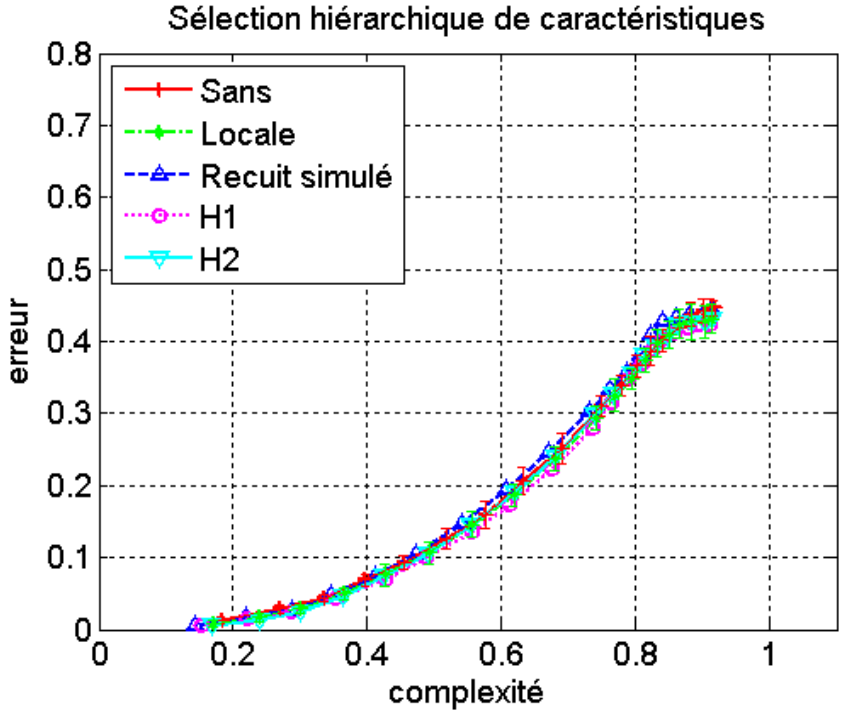


FIGURE 5.12 – *Test de différentes méthodes hiérarchiques basées sur la méthode* SCORR *avec $K = 20$.*

Méthode	CAUC
AUCUNE	$92,65 \pm 0,45$
LOCALE	$92,92 \pm 0,56$
RECUIT	$92,43 \pm 0,28$
H1	$\mathbf{93,17 \pm 0,45}$
H2	$92,92 \pm 0,50$

TABLE 5.3 – *Résultats obtenus par les différentes méthodes de sélection de caractéristiques hiérarchique, pour $K = 20$, comparés avec la sélection de caractéristiques locale et l'absence de sélection, sur la base de voitures.*

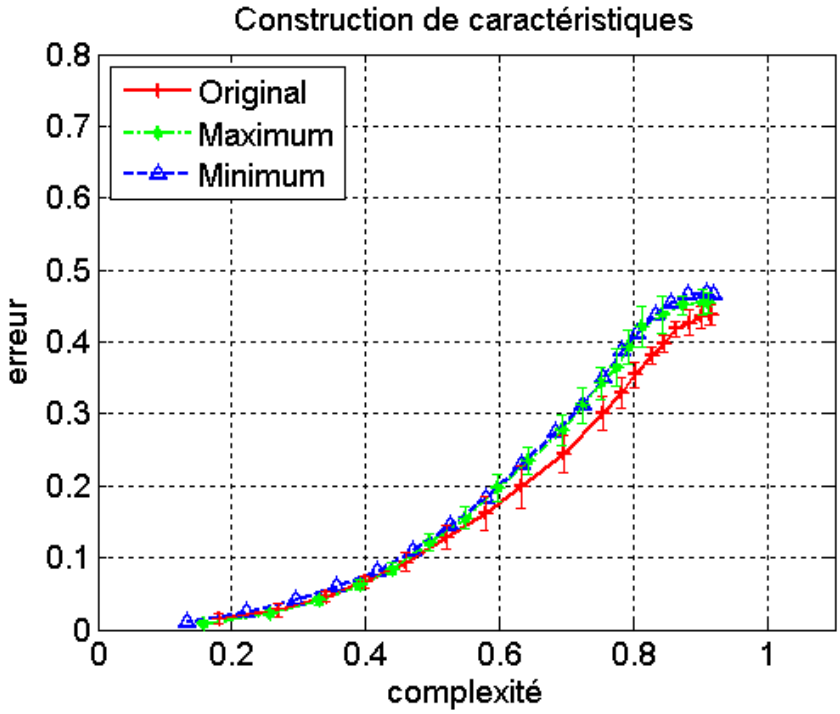


FIGURE 5.13 – *Courbes erreur-complexité pour différentes extensions de la signature s'appuyant sur la hiérarchie.*

complexité. Contrairement à ce que l'on aurait pu attendre, l'extension de la signature ne permet pas d'améliorer les performances. Une explication possible est que certaines combinaisons de détecteurs renforcent l'influence de fausses alarmes.

Nous appliquons ensuite la sélection de caractéristiques sur le jeu étendu de caractéristiques. À ce niveau d'expérience, nous ne pouvons pas proposer une interprétation précise. Nous donnons simplement les résultats en termes de performances, figure 5.14 : ce sont à peu près les mêmes résultats que pour la sélection de caractéristiques locale.

D'une manière générale, nos résultats montrent que la sélection de caractéristiques, hiérarchique ou non, n'apporte pas grand'chose en termes de performances, mais qu'elle est très utile à l'interprétation des résultats. En particulier, elle peut s'avérer un outil intéressant pour déterminer quelles sont les faiblesses de la description extraite de l'image et orienter les optimisations futures.

5.5 Discussion et perspectives

Dans ce chapitre, nous avons cherché à donner plus de sens aux caractéristiques extraites de l'image en étudiant la façon dont elles influencent la décision. Pour cela, nous nous sommes intéressés à la traditionnelle sélection de caractéristiques, plus particulièrement aux méthodes fournissant une fonction de classement. Notre but étant d'interpréter les caractéristiques sélectionnées, nous avons traité le problème en priorité dans le cas de la base de voitures, pour laquelle les signatures présentent une signification sémantique pouvant être directement mise en lien avec les annotations. Nous avons d'abord appliqué quelques méthodes classiques de sélection de caractéristiques de manière locale, c'est-à-dire de manière indépendante sur chaque classifieur (i.e. chaque $\mathbf{y} \in \mathcal{H}$). Nous avons ensuite proposé plusieurs méthodes pour avoir une sélection de caractéristiques hiérar-

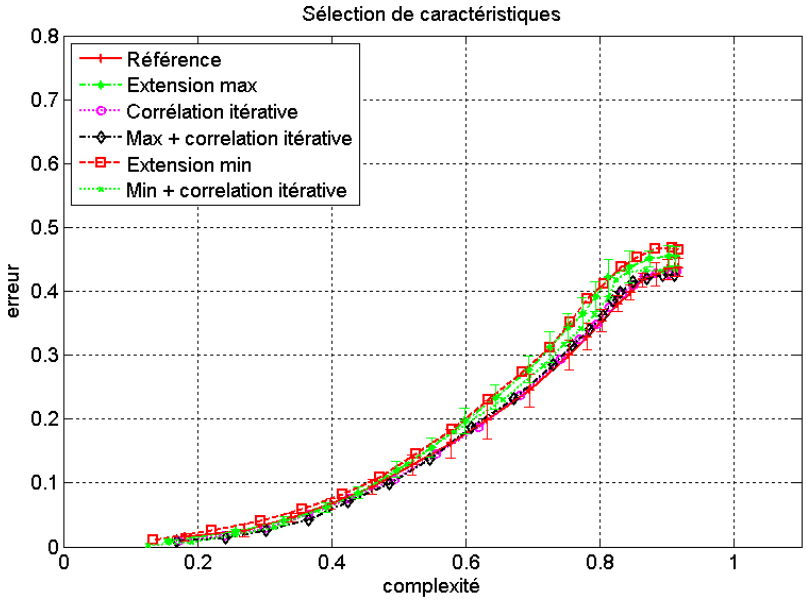


FIGURE 5.14 – *Courbes erreur-complexité pour la sélection de caractéristiques après extension de la signature.*

chique, c'est-à-dire que les caractéristiques sélectionnées soient cohérentes avec la structure de l'hypergraphe.

Nous avons présenté les résultats des expériences sous deux formes. Tout d'abord, nous avons étudié l'apport de la sélection de caractéristiques pour l'analyse des résultats d'annotation. Puis, nous avons vu leur intérêt pour l'annotation elle-même, c'est-à-dire les performances.

La signature étudiée est déjà de relativement faible dimension ($d = 197$, à comparer par exemple avec $d = 4200$ pour les SPM). Il se trouve que la sélection de caractéristiques ne permet pas d'augmenter les performances de manière significative. Cependant, elle permet une analyse des caractéristiques de laquelle nous pouvons tirer plusieurs conclusions, ouvrant de nombreuses perspectives d'améliorations des signatures.

En effet, les expériences nous ont permis de confirmer l'intuition naturelle : dans la majorité des cas, c'est bien la présence de quelques indices liés à la classe qui ont permis la reconnaissance. Ceci renforce notre thèse, déjà défendue au chapitre 3, selon laquelle il est utile de s'appuyer sur quelques détails significatifs. Nous avions alors démontré leur efficacité en termes de performances. Nous avons maintenant démontré leur intérêt en termes d'interprétabilité des résultats. L'analyse effectuée nous permet de proposer plusieurs orientations de recherche :

- En interprétant l'origine des erreurs, nous pouvons mieux cibler les optimisations à apporter sur le calcul des signatures. En particulier, nous avons montré que ce sont les fausses alarmes de type II qui gêne le plus la reconnaissance. Des recherches complémentaires au chapitre 3 sont donc nécessaires pour améliorer les performances générales des détecteurs de détails.
- Il est possible de réduire le nombre de caractéristiques, en ne prenant que celles sélectionnées pour au moins 1 multilabel. Il est possible de l'augmenter, en construisant des caractéristiques communes à plusieurs modèles. Plus généralement, il serait intéressant d'étudier les différentes formes de signatures possible, en combinant et supprimant des détails. Par exemple, en construisant une signature plus souple, plus modulable, il sera plus facile d'envisager l'ajout de nouvelles catégories.

– Nous avons montré quelques résultats exploitant la sélection de caractéristiques hiérarchique, mais cette partie mérite encore une attention spéciale. En effet, c'est le meilleur moyen de produire une interprétation plus approfondie de l'algorithme et de comprendre (a) comment adapter les signatures au contexte de la hiérarchie, et (b) comment adapter l'algorithme d'annotation pour qu'il puisse exploiter au mieux la hiérarchie.

Le développement d'outil d'analyse plus souples et plus performants pourrait lui-même constituer un sujet de recherche. En effet, la quantité de données en jeu (nombre d'images, nombre de caractéristiques, nombre de multilabels) fait qu'il est difficile de visualiser l'information de manière claire. Nous en avons proposé ici quelques unes, et nous avons pu en tirer une première analyse. L'introduction d'une interface interactive pourrait être une solution.

Conclusion

6

Cette thèse avait pour objet le problème de l'annotation sémantique d'images : le but était de fournir une information hiérarchique et multifacette, c'est-à-dire gérant différents points de vue descriptifs. Dans un premier temps, nous avons justifié cette approche, en montrant que pour s'adapter aux différents contextes d'emploi, il est nécessaire de créer des métadonnées assez riches, et organisées d'une manière naturelle, c'est-à-dire selon une structure sémantique faisant consensus. Nous avons vu que si un vocabulaire structuré était assez souvent introduit dans les systèmes d'annotation, il était plus rare qu'il soit utilisé dans la phase d'apprentissage. Enfin, il est apparu qu'aucun des systèmes actuels ne permettait de gérer nativement des aspects multifacette et hiérarchique, tout en intégrant un contrôle par rapport à un compromis précision/fiabilité.

6.1 Contributions

L'étude que nous avons présentée avait pour objectif de combler ce manque, en développant un système d'annotation ayant ces propriétés. Pour cela, nous avons tout d'abord introduit une base d'images de voitures, dont les modèles ont été sélectionnés de manière à permettre différents points de vue descriptifs, et différents niveaux de précision sémantique.

Introduction d'une nouvelle base de données

Nous avons ensuite évalué les algorithmes classiques de classification d'objets, qui ne s'appuient que sur un vocabulaire non-structuré. Nous avons montré que ces méthodes de description classiques donnaient des performances faibles sur notre base d'images de référence. Nous avons donc développé une signature à base de descripteurs de détails sémantiques, ayant un meilleur pouvoir informatif, et permettant d'améliorer de manière conséquente les performances de référence : les SPM [113] permettaient d'avoir un taux de classification de 31%, notre signature permet d'atteindre les 51%.

Développement d'une signature spécifique, performante

Nous sommes ensuite passés au vif du sujet, à savoir le développement d'une méthode d'annnotation multifacette hiérarchique. Pour cela, nous avons tout d'abord proposé une formalisation du problème permettant d'intégrer des contraintes sur la notion de multilabel, notamment en munissant l'espace des multilabels d'une structure d'hypergraphe. Nous avons ensuite développé une méthode d'annotation utilisant les multilabels définis par cet hypergraphe, en nous appuyant sur des techniques classiques d'apprentissage statistique.

Formalisation du problème

Développement d'une méthode d'annotation répondant aux objectifs

Nous avons également proposé une méthode d'évaluation adaptée à l'annotation multilabel hiérarchique : la courbe erreur/complexité. Nous avons présenté les résultats d'expériences effectuées sur la base d'images de voitures introduite, ainsi que sur la base Caltech-101. Ces résultats montrent que l'exploitation de la hiérarchie est surtout utile pour la reconnaissance aux niveaux intermédiaires et

Méthode d'évaluation

supérieurs de la hiérarchie. Les évaluations ont également montré de grandes disparités de performances selon les bases d'images, suggérant d'utiliser la hiérarchie à différentes étapes selon la nature des données.

Développement d'une méthode de sélection hiérarchique de caractéristique

Enfin, dans un dernier chapitre, nous avons examiné le problème de la sélection de caractéristiques hiérarchique, avec deux objectifs : améliorer les performances, et donner des éléments d'interprétation des décisions. Nous avons proposé une méthode de sélection de caractéristiques pour des données structurées basée sur la définition d'une fonction de coût et d'un recuit simulé.

6.2 Perspectives

L'étude que nous avons présentée dans ce manuscrit peut être poursuivie selon plusieurs orientations. Nous les partagerons en deux groupes :

1. des orientations *techniques*, visant purement à améliorer les performances,
2. des orientations *théoriques*, visant à fournir une analyse de l'influence de la hiérarchie pour l'annotation, et à se rapprocher d'un processus d'interprétation humain.

D'un point de vue technique, nous pouvons imaginer des améliorations à tous les niveaux de la chaîne de traitement proposée :

- Par rapport à la *description des images*, décrite au chapitre 3. Les détecteurs de détails proposés pourraient être améliorés de manière simple et efficace en introduisant des contraintes géométriques. L'algorithme de détection lui-même pourrait être rendu plus performant en introduisant de l'apprentissage actif. L'étude du chapitre 5 nous a permis de mettre en évidence la nécessité d'améliorer les performances des détecteurs d'une manière générale pour augmenter le taux d'annotations correctes.
- Au niveau de l'*annotation* multifacette hiérarchique, certains calculs pourraient être factorisés, d'autres parallélisés, pour réduire le temps de calcul. D'autres stratégies d'exploitation de la structure d'hypergraphe pourraient être imaginées, notamment en exploitant des fonctions de coût hiérarchiques, et en étudiant le rôle des taux de confiance.
- Enfin, la *sélection de caractéristiques* peut être utilisée pour réduire la quantité de calcul en estimant des signatures simplifiées. Il serait aussi intéressant d'étudier la possibilité de calculer ces signatures de manière progressive pour affiner l'annotation (donc avec une démarche descendante).

Sur un plan plus théorique, l'exploitation de la hiérarchie suscite également des questions à différents niveaux. Par exemple, il serait intéressant d'étudier de manière plus poussée quelles sont les signatures "sémantiques" les plus efficaces, selon la précision souhaitée. L'étude commencée en ce sens au chapitre 5 mériterait d'être approfondie. Cette question est étroitement liée au problème de l'ajout de nouvelles catégories, soulevé au chapitre 3. En effet, la signature que nous avons introduite est apprise pour un nombre déterminé de catégories. Comme nous l'avons vu au chapitre 2, la création de vocabulaires visuels structurés a déjà été étudiée dans la littérature avec des catégories non structurées [174, 56, 71, 4]. En exploitant la structure d'hypergraphe, il serait intéressant de proposer une signature "adaptative", dont les éléments seraient eux-mêmes structurés.

Notre étude montre qu'il est non seulement possible, mais utile d'exploiter un vocabulaire structuré pour avoir une annotation plus souple et plus riche. Il serait maintenant intéressant d'essayer de comprendre plus exactement quelle est l'influence de la structure. Comme nous l'avons relevé au chapitre 4, il faudrait observer comment des changements de la structure modifient les annotations :

par exemple, si une structure touffue est plus intéressante qu'une structure clairsemée. Par ailleurs, la comparaison entre la structure d'hypergraphe "sémantique" et les possibles taxonomies visuelles, dépendantes d'un algorithme, mérite une attention particulière. Il s'agit en fait d'étudier la cohérence entre similarités sémantiques et similarités visuelles. Il est clair que ces deux aspects diffèrent largement : c'est le fossé sémantique. Cependant, la définition de signatures sémantiques devrait permettre, logiquement, la création de taxonomies visuelles plus proches des hiérarchies sémantiques.

Nous avons été confrontés au problème des données : la base Caltech est mal adaptée, car trop diverse, et la base de voitures contient peu de données. Pour les futures recherches, il faudra s'intéresser à de nouvelles bases d'images. ImageNet semble pouvoir être adaptée sans trop de difficultés, bien que l'aspect multifacette ne soit pas explicite. Quoiqu'il en soit, la communauté vision montre un intérêt croissant pour les vocabulaires structurés, qui représentent une manière peu coûteuse d'ajouter de la supervision.

Enfin, nous avons présenté l'annotation essentiellement dans un contexte de recherche d'images. Le format d'annotation proposé, sous la forme d'une distribution de multilabels, peut être exploité dans d'autres contextes. Dans cette optique, il serait intéressant d'explorer le rôle des taux de confiance, par exemple en les intégrant dans des mesures de similarité entre distributions de multilabels. Se pose également le problème de la fusion de distributions de multilabels, pour s'adapter aux besoins de l'utilisateur en fonction du contexte.

Extraction de caractéristiques

A

A.1 Introduction

Dans le chapitre 2, nous avons présenté les différentes manières dont était abordé le problème de l'analyse sémantique d'images dans l'état de l'art. Nous avons vu que les algorithmes utilisés pour cela comportaient en général deux parties : (1) l'extraction des caractéristiques "bas-niveau", et (2) l'interprétation de ces caractéristiques. Dans ce chapitre, nous nous intéressons plus particulièrement à la première phase du traitement : partant de l'image $\mathcal{I}$ brute, le but est d'avoir en sortie un vecteur caractéristique de dimension fixe d, parfois appelé signature de l'image. Dans ce chapitre bibliographique, nous cherchons à présenter (a) quelles sont les méthodes les plus performantes, (b) quelles sont les représentations pouvant se rapprocher le plus de représentations "sémantiques".

A.2 Descripteurs locaux

Les descripteurs locaux, au lieu de décrire l'image dans sa globalité, s'appuient sur des points d'intérêt, et extraient un voisinage pour la description. En pratique, un point d'intérêt sera défini par sa position (x, y) dans l'image, et par une échelle σ, qui sert à définir la taille du voisinage. Il y a plusieurs méthodes pour choisir les points d'intérêts, nous ne les verrons pas toutes, mais nous nous arrêteront sur celles qui sont utiles dans la suite du manuscrit. En général le choix des points d'intérêts est indépendant des descripteurs. Il existe un grand nombre de descripteurs locaux possibles, nous en verrons quelques uns qui nous paraissent significatifs ou qui seront utilisés plus loin dans notre étude. Une étude approfondie des descripteurs locaux est proposée par Mikolajczyk et Schmid [128].

A.2.1 Sélection de zones d'intérêts

Il existe de nombreuses méthodes permettant de détecter automatiquement les zones d'intérêt de l'image. Mikolajczyk et al. [129] en font une revue très détaillée et nous ne les reprendrons pas ici. Bien que ces méthodes soient très utilisées pour la reconnaissance d'objets, elles ont été développées à l'origine pour la mise en correspondance d'image : les zones qui se répètent sur deux images similaires ne sont pas forcément les mêmes que les zones similaires sur des objets différents. Les zones recherchées dans le premier cas sont "réellement" invariantes par transformation géométrique, et peuvent être approchées par une transformation affine. Par contre, pour deux objets différents d'une même catégorie, il ne s'agit plus de transformation géométrique.

Si la détection de points d'intérêts est intéressante pour reconnaître des instances d'objets ou pour faire de la mise en correspondance, il n'est pas évident

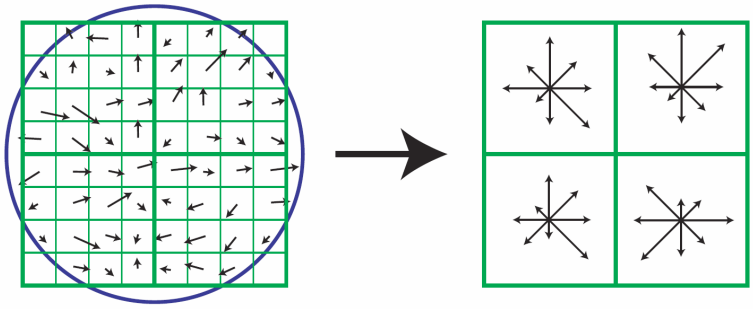


FIGURE A.1 – *Principe de calcul des SIFT : l'amplitude et l'orientation du gradient est calculée en chaque pixel, l'amplitude étant pondérée par une gaussienne (cercle) ; ces valeurs sont ensuite accumulées dans un histogramme sur 4x4 régions (2x2 sont représentées) et 8 orientations. Figure extraite de [121].*

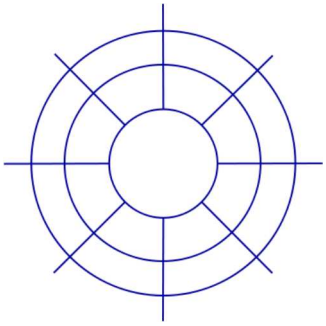


FIGURE A.2 – *Division de l'espace pour le descripteur GLOH*

que ce soit la méthode la plus performante pour la reconnaissance d'objets plus génériques. Nowak et al. [137] ont montré qu'il est utile d'avoir une représentation dense de l'image. Or, en détectant des points d'intérêts, il est difficile de contrôler le nombre de points extraits. Nowak et al. [137] montrent qu'une sélection dense de points aléatoires est plus performante. Fei-Fei et Perona [65] confirment ce résultat, et montrent de plus qu'utiliser une grille dense est encore plus efficace (pour la reconnaissance de scènes).

A.2.2 Descripteurs SIFT

L'étude proposée par Mikolajczyk et Schmid [128] conclut que le descripteur le plus efficace est le descripteur SIFT, introduit par Lowe [121], et mieux encore, sa variation GLOH. Un descripteur SIFT (*Scale Invariant Feature Transform*) décrit le voisinage d'un point en construisant un histogramme des orientations du gradient. Le vecteur gradient (orientation et amplitude) est calculé en chaque pixel à l'échelle correspondant au voisinage. Les orientations sont discrétisées pour n'avoir plus que 8 orientations. Le voisinage est divisé en une grille 4×4. Dans chaque case, et pour chaque orientation, les amplitudes des gradients sont accumulées, pour donner un vecteur caractéristique de dimension $d = 128$. La figure A.1 illustre ce principe.

Le descripteur GLOH (*Gradient Location-Orientation Histograms*), proposé par Mikolajczyk et Schmid [128], part du même principe que les SIFT, mais divise l'espace par rapport coordonnées log-polaires (voir figure A.2). Avec 3 rayons et 8 orientations, mais 1 seule orientation pour le premier rayon, ils obtiennent 17 cases spatiales. Ils quantifient les orientations en 16 cases, ce qui donne un descripteur de dimension 272. Ils utilisent une ACP pour réduire cette dimension à 128.

A.2.3 Histogrammes d'orientations de gradients

Le principe des histogrammes de gradients orientés (HOG), proposé par Dalal et Triggs [41], est assez proche des SIFT. Cependant ils sont prévus pour être calculés sur une grille dense uniforme, avec une normalisation locale du contraste permettant d'améliorer les performances. L'image est divisée en petites régions, ou cellules, sur lesquelles on calcule un histogramme accumulant les directions du gradient. Chaque vecteur calculé sur une cellule est normalisé par rapport à une "énergie" calculée sur une région plus grande de l'image, appelée bloc. Le descripteur HOG correspond au descripteur normalisé sur un bloc. Dalal et Triggs évaluent précisément l'influence des différents paramètres pour la reconnaissance de personnes : une normalisation préliminaire en gamma ou en couleur a peu d'influence ; le gradient le plus simple, sans lissage est le meilleur ; une division assez fine des orientations est intéressante ; le signe des gradients n'aide pas pour la reconnaissance de personnes, mais peut aider pour d'autres objets ; diminuer le pas entre blocs (augmenter les chevauchements) améliore les résultats ; normaliser des blocs séparément vaut mieux que normaliser par cellule.

Plus généralement, la représentation des gradients locaux (contours) et une bonne normalisation semblent donner des informations importantes pour la reconnaissance.

A.2.4 Descripteurs de segments

Ferrari et al. [70] proposent une méthode d'extraction de caractéristiques basées sur les contours : les kAS. Soit une liste de segments $P = (s_1, \ldots, s_k)$, $\mathbf{r}_i = (r_i^x, r_i^y)$ le vecteur reliant le milieu de s_1 au milieu de s_i, θ_i, $l_i = ||s_i||$ l'orientation et la longueur de s_i. Le descripteur de P, de longueur $d = 4k - 2$ est alors :

$$x = \left(\frac{r_2^x}{N_d}, \frac{r_2^y}{N_d}, \ldots, \frac{r_k^x}{N_d}, \frac{r_k^y}{N_d}, \theta_1, \ldots, \theta_k, \frac{l_1}{N_d}, \ldots, \frac{l_k}{N_d} \right), \tag{A.1}$$

où N_d est la distance entre les deux points milieux les plus éloignés, qui définit en même temps l'échelle du descripteur. La position du kAS est définie comme le centre de tous les points-milieux. La figure A.3 montre des exemples de 2AS.

Pour comparer deux descripteurs, les auteurs proposent d'utiliser la distance suivante :

$$D(a,b) = w_r \sum_{i=2}^{k} ||\mathbf{r}_i^a - \mathbf{r}_i^b|| + w_\theta \sum_{i=1}^{k} D_\theta(\theta_i^a, \theta_i^b) + \sum_{i=1}^{k} |\log(\frac{l_i^a}{l_i^b})|, \tag{A.2}$$

avec $D_\theta \in [0,1]$ mesurant la différence entre 2 orientations normalisée par π. Le dernier terme, qui compare les différences entre longueurs, est peu fiable : un poids plus faible lui est associé.

A.2.5 Image intégrale

Introduite par Viola et Jones [189] pour accélérer le calcul des ondelettes de Haar, l'image intégrale d'une image naturelle $\mathcal{I}$ est telle qu'un pixel en x, y contient la somme des pixels au-dessus et à gauche de ce point dans $\mathcal{I}$. Formellement,

$$ii(x,y) = \sum_{x' \leq x, y' \leq y} i(x', y'), \tag{A.3}$$

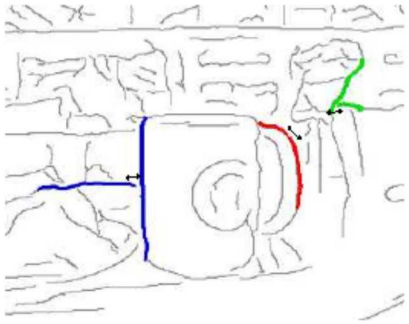


FIGURE A.3 – *Image de contours, avec trois exemples de 2AS. Figure extraite de [70].*

où ii est l'image intégrale et i est l'image d'origine. Cette image peut être construite en une seule passe sur les pixels de l'image en utilisant les formules récursives suivantes :

$$s(x,y) = s(x,y-1) + i(x,y), \tag{A.4}$$
$$ii(x,y) = ii(x-1,y) + s(x,y), \tag{A.5}$$

où $s(x,y)$ est la somme des pixels de coordonnées (x_i, y_i) telles que $x_i \leq x$ et $y_i \leq y$, et $s(x,-1) = ii(-1,y) = 0$. Grâce à l'image intégrale, une somme de pixel sur n'importe quel rectangle de l'image peut être calculée en seulement 4 références.

A.3 Combinaisons de descripteurs locaux

A.3.1 sacs-de-mots

Les sacs-de-mots permettent de décrire l'image de manière globale en s'appuyant sur des descripteurs locaux. Ils ont été introduits par Dance et al. [42], et sont devenus une référence. Le principe des sacs-de-mots est le suivant :

- extraire un certain nombre de descripteurs des images de la base d'apprentissage ;
- les regrouper en d *clusters* ; chaque cluster correspond à un *mot visuel* ;
- décrire une nouvelle image en construisant l'histogramme décomptant le nombre de descripteurs associés à chacun des mots visuels.

La figure A.4 illustre le principe de construction du vocabulaire. La figure A.5 illustre la construction d'une signature sac-de-mots pour une nouvelle image. L'idée de "sacs" dénote le désordre géométrique. En effet, en prenant les histogrammes d'occurrences, on perd toute information géométrique. Étonnamment cela n'empêche pas d'avoir de bonnes performances.

La méthode par sacs-de-mots est à l'origine une méthode d'analyse de documents textuels. Elle a été adaptée pour l'analyse d'image par l'intermédiaire de l'analogie entre les "mots" et les "mots visuels", qui sont en fait des vecteurs décrivant des caractéristiques visuelles de bas-niveau (couleur, texture...). Dans sa formulation originale, les sacs-de-mots s'appuyaient sur des descripteurs SIFT extraits autour des points d'intérêts de Harris. Les principales améliorations utilisent d'autres points d'intérêts, ou des combinaisons de points d'intérêt, ou mieux encore, une grille dense (cf. A.2.1). Les résultats de Nowak et al. [137] et de Fei-Fei et Perona [65] semblent montrer que c'est le nombre de points extraits qui fait la fiabilité du descripteur, plutôt que leur position. Le vocabulaire est construit par l'algorithme des K-moyennes, avec $K = 1000$ mots (fixé empiriquement). Des

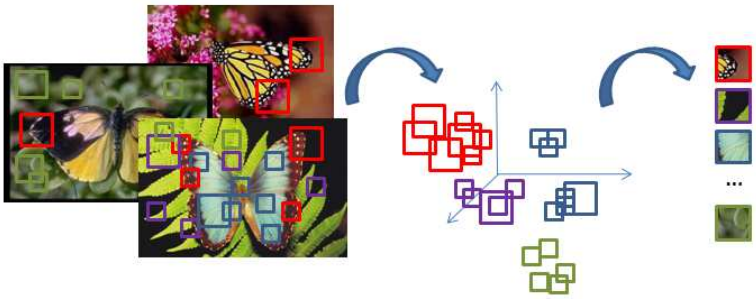


FIGURE A.4 – *Construction du vocabulaire par sacs-de-mots : extraction de descripteurs ; regroupement ; création du vocabulaire, dans lequel un mot visuel correspond à un cluster (ou plus précisément, à son centre).*

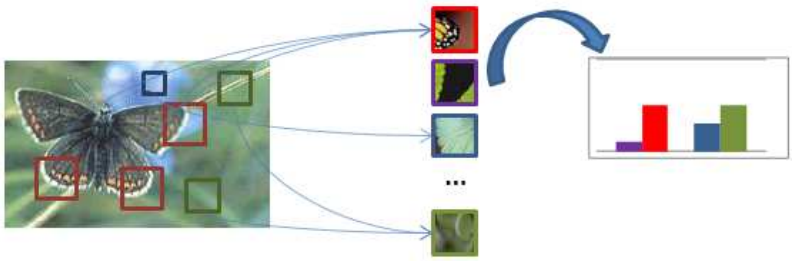


FIGURE A.5 – *Construction d'un descripteur sac-de-mots : extraction de descripteurs ; assignation au mot visuel le plus proche ; construction de l'histogramme des occurrences de chaque mot.*

améliorations sont aussi proposées à ce niveau. Par exemple Jurie et Triggs [105] utilisent l'algorithme *mean-shift*, qui s'adapte mieux aux variations de densité dans l'espace de description. Ils montrent qu'une sélection de caractéristiques sur un vocabulaire dense permet d'améliorer les résultats. Moosmann et al. [132] utilisent des arbres aléatoires pour construire un vocabulaire plus discriminatif, plus robuste, et pour lequel apprentissage et test sont plus rapides. La construction de vocabulaires discriminatifs pour la reconnaissance est un problème complexe et fait l'objet de recherches spécifiques (voir Larlus [112]). Zhang et al. [201] proposent une analyse de l'influence de différents paramètres dans le schéma sacs-de-mots sur la reconnaissance de textures et de catégories d'objets. Ils évaluent les détecteurs de points d'intérêt, les descripteurs locaux, ainsi que différents noyaux. Ils montrent par exemple qu'il faut éviter d'avoir des descripteurs trop invariants. Sachant que les méthodes sacs-de-mots utilisent des caractéristiques extraites de toute l'image, et décrivent donc le fond avec l'objet, ils étudient également l'influence de l'arrière-plan : ils montrent que le classifieur généralise mieux lorsque les images d'apprentissage ont des arrière-plans variés et complexes.

A.3.2 Pyramides de descripteurs

Bosch et al. [28] s'inspirent des Spatial Pyramid Matching et des HOG pour construire des PHOG (Pyramides d'histogrammes d'orientations de gradients, voir figure A.6). L'image est divisée de la même manière que pour les Spatial Pyramid Matching, mais les histogrammes sont des HOG et non des sacs-de-mots. Les histogrammes, plus précisément, accumulent les orientations des contours, où les contours sont calculés par le détecteur de Canny, et les orientations des gradients par un filtre de Sobel. Les histogrammes sont concaténés, avec les poids sur les niveaux appris globalement ou pour chaque classe (contrairement aux Spatial

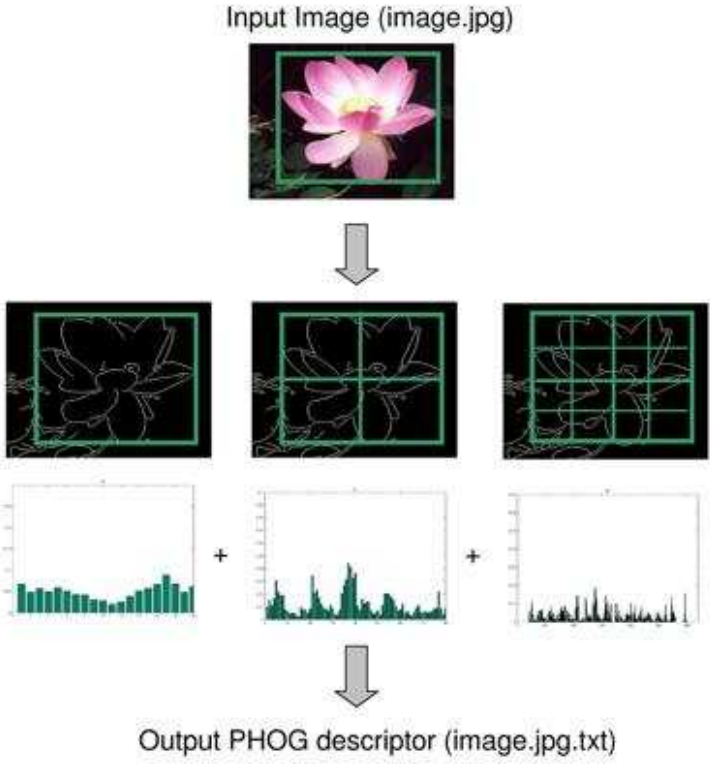


FIGURE A.6 – *Principe du descripteur PHOG, appliqué ici sur une partie de l'image. Source : http://www.robots.ox.ac.uk/~vgg/research/caltech/phog.html.*

Pyramid Matching où les poids sont fixés). Ils montrent que la combinaison de descripteurs PHOG (forme) et de SPM (apparence) surpasse chacun des deux pris séparément de manière significative.

A.3.3 Hiérarchies de caractéristiques

Ullman et al. [186] montrent que les caractéristiques de complexité intermédiaires sont les plus adaptées à la reconnaissance d'objets au niveau fondamental. Les caractéristiques utilisées sont des fragments d'images de tailles et de résolutions variées. La quantité d'information apportée par un fragment est définie en utilisant l'information mutuelle. En faisant varier la taille des fragments, ils trouvent que l'information mutuelle est maximale lorsque les fragments ont une taille correspondant à 11-16% de la taille de l'objet. Vidal-Naquet et Ullman [188] approfondissent ces résultats et montre que des caractéristiques basées sur la présence/l'absence de tels fragments permettent une classification plus efficace que des caractéristiques simples comme des ondelettes type Haar. Dans Epshtein et Ullman [56], les fragments sont associés à des parties d'objets, et organisés hiérarchiquement en parties de plus en plus petites. De plus, les noeuds de la hiérarchie sont associés à des parties "sémantiquement équivalentes", en utilisant les relations de géométries.

Fidler et Leonardis [71] adoptent une démarche incrémentale et non supervisée dans la construction des caractéristiques. Les caractéristiques de bas-niveau sont extraites sur toutes les catégories de manière indifférente et correspondent aux niveaux les plus bas de la hiérarchie. Les niveaux supérieurs sont appris séparément pour chaque catégorie. La hiérarchie s'incrémente en ajoutant de nouvelles catégories, sans avoir besoin de recalculer la hiérarchie entière. Les parties du niveau

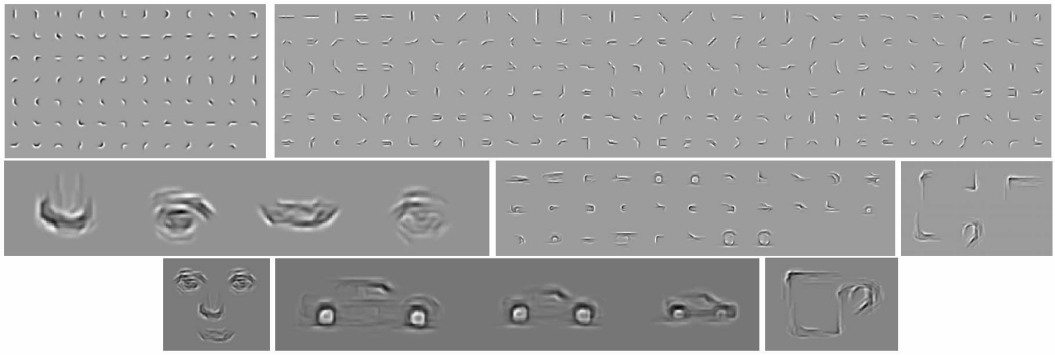


FIGURE A.7 – *Représentation des reconstructions moyennes des parties apprises à différents niveaux. 1^{ière} ligne : $\mathcal{L}_2$, $\mathcal{L}_3$, parties universelles ; 2ième ligne : $\mathcal{L}_4$ pour des visages, des voitures, et des tasses ; 3ième ligne : $\mathcal{L}_5$ pour des visages, des voitures à trois échelles, et des tasses.*

le plus bas ($\mathcal{L}_1$) sont les maximas locaux issus de filtres de Gabor à différentes orientations et échelles. Chaque niveau est ensuite construit par combinaison des parties du niveau inférieur. la figure A.7 montre différents niveaux de la hiérarchie. Au niveau $\mathcal{L}_4$, on arrive à reconnaître certaines parties d'objets. Au niveau $\mathcal{L}_5$, les objets eux-mêmes sont reconnaissables.

Techniques d'apprentissage statistique

B

B.1 Introduction

Nous présentons quelques algorithmes d'apprentissage statistique. Quand il y a lieu, nous mentionnons leur usage dans la littérature sur l'analyse sémantique d'image. Pour une connaissance plus approfondies des nombreuses méthodes d'apprentissage, nous renvoyons le lecteur à des ouvrages de référence dédiés à celles-ci, tels que Hastie et al. [89], Bishop [23] ou Bakir et al. [11].

B.1.1 Formalisation du problème

On considère ici un ensemble d'apprentissage $\mathcal{A}_n$:

$$(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathcal{L}, \tag{B.1}$$

où :

- $n \in \mathbb{N}$ est le nombre d'observations,
- les x_i sont les entrées, aussi appelées caractéristiques, ou observations,
- $\mathcal{X}$ est l'espace de représentation des observations, en général $\mathcal{X} \subset \mathbb{R}^d$,
- les y_i sont les sorties, ou labels, représentant une *vérité terrain*,
- et $\mathcal{L}$ est un ensemble de labels.

Les couples $(x_1, y_1), \ldots, (x_n, y_n)$ sont supposés être des réalisations indépendantes d'une loi P inconnue.

Le but de l'apprentissage statistique est d'apprendre une fonction de prédiction $g : \mathcal{X} \to \mathcal{L}$ à partir d'un ensemble d'apprentissage.

On note $\ell : \mathcal{L} \times \mathcal{L} \to \mathbb{R}$ la fonction de perte (ou fonction de coût), mesurant le coût $\ell(y, y')$ d'une confusion entre le label réel y et le label estimé $y' = g(x)$. La fonction de prédiction optimale minimise le coût moyen, appelé aussi risque, ou erreur de généralisation :

$$R(g) = \mathbb{E}[\ell(y, g(x))]. \tag{B.2}$$

P étant inconnue, le risque sera estimé par sa version empirique R_{emp} calculée sur $\mathcal{A}_n$:

$$R_{emp}(g) = \sum_{i=1}^{n} \ell(y_i, g(x_i)). \tag{B.3}$$

B.1.2 La sélection de modèle

L'algorithme d'apprentissage dépend d'un certain nombre de paramètres. On souhaite les optimiser pour avoir la plus petite erreur de prédiction possible. Pour

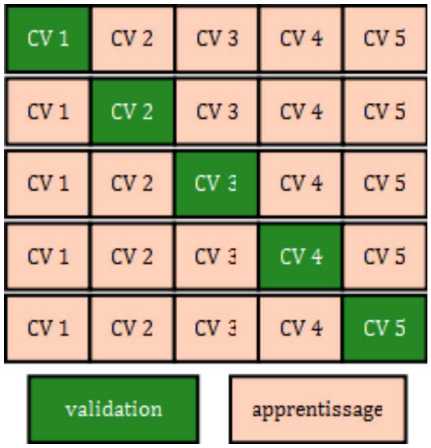


FIGURE B.1 – *Principe de la validation croisée avec $K = 5$. La base est divisée en K partitions qui servent successivement d'ensemble de validation.*

cela, on utilise une partie de l'ensemble d'apprentissage pour apprendre la fonction de prédiction, et le reste, appelé ensemble de *validation*, pour estimer l'erreur de généralisation. Lorsqu'il y a peu de données d'apprentissage, l'erreur de généralisation devient très bruitée, et l'optimisation des paramètres devient d'autant plus difficile.

Dans ce cas, il est intéressant d'utiliser la validation croisée. Cette méthode consiste à diviser l'ensemble d'apprentissage en K ensembles, à répéter K fois la procédure de calcul de l'erreur et avoir ainsi une estimation plus robuste. Ce principe est illustré figure B.1.

Soit g une fonction d'interprétation sur un ensemble d'apprentissage $\mathcal{A}_n$ de taille n. On note L la fonction de perte associée. L'erreur est estimée par le risque empirique R_{emp} associé à un ensemble test.

Soit $\kappa : \{1,\dots,n\} \mapsto \{1,\dots,K\}$ une fonction d'indexation définissant une partition de $\mathcal{A}_n$ en K ensembles. On note $\hat{g}^{-k}$ la fonction d'interprétation apprise sur $\mathcal{A}_n \backslash \mathcal{A}_{\kappa^{-1}(k)}$, c'est-à-dire $\mathcal{A}_n$ privé du k-ième ensemble.

L'erreur estimée par validation croisée est :

$$CV = \frac{1}{n}\sum_{i=1}^{n} L(u_i, \hat{g}^{-\kappa i}(x_i)). \tag{B.4}$$

Si on a un ensemble de modèles paramétrés par α, on note $g^{-k}(x,\alpha)$ la fonction d'interprétation apprise avec la k-ième partie de l'ensemble supprimée et le paramètre α. L'erreur de validation croisée est alors une fonction de α.

$$CV(\alpha) = \frac{1}{n}\sum_{i=1}^{n} L(u_i, \hat{g}^{-\kappa i}(x_i, \alpha)). \tag{B.5}$$

Il suffit ensuite de choisir le modèle correspondant au paramètre $\hat{\alpha}$ minimisant la courbe obtenue.

B.2 Le classifieur Bayésien Naïf

B.2.1 Principe

Dans ce modèle, on fait l'hypothèse que les caractéristiques x_k sont indépendantes les unes des autres. La densité de probabilité d'un vecteur caractéristique

$x = \{x_1, \ldots, x_d\}$ pour une classe k s'écrit donc

$$f_k(x) = \prod_{j=1}^{d} f_{kj}(x_j). \tag{B.6}$$

Cette hypothèse est rarement vérifiée en pratique, mais elle permet de simplifier grandement l'estimation. En effet, on peut alors estimer séparément les densités de chacune des caractéristiques.

D'après le théorème de Bayes, on peut écrire :

$$P(\mathcal{C}_k|x) = \frac{1}{\xi} P(\mathcal{C}_k) \prod_{j=1}^{d} P(x_j|\mathcal{C}_k). \tag{B.7}$$

Les probabilités $P(\mathcal{C}_k)$ et $P(x_j|\mathcal{C}_k)$ sont estimées sur la base d'apprentissage tout simplement en comptant les occurrences des x_j pour chaque classe. Si les x_j sont binaires, on a $P(x_j = 1|\mathcal{C}_k) = 1 - P(x_j = 0|\mathcal{C}_k)$. ξ est un facteur de normalisation qui ne dépend que de x.

$$\xi = \sum_{k=1}^{K} P(\mathcal{C}_k) \prod_{j=1}^{d} P(x_j|\mathcal{C}_k), \tag{B.8}$$

en supposant que les classes sont indépendantes.

B.2.2 Remarques bibliographiques

Malgré les hypothèses fortes, ce classifieur donne souvent de très bons résultats. Il reste très populaire, du fait de sa simplicité. Ainsi, Dance et al. [42] comparent la classifieur bayésien naïf à un SVM linéaire pour classifier des caractéristiques issues des sacs-de-mots. Bien que le SVM linéaire soit globalement meilleur, le classifieur bayésien naïf fait mieux sur au moins une classe (*phone*). Aksoy et al. [5] utilisent ce type de classifieur pour segmenter des images satellites, en faisant une classification au niveau pixellique.

B.3 Les K-plus-proches-voisins

B.3.1 Principe

L'algorithme des K-plus-proches-voisins est un algorithme de classification par moyennage local. Pour classifier un point x, l'idée est de trouver ses K plus proches voisins dans l'ensemble d'apprentissage $\mathcal{A}_n$. Notons $\{\mathcal{C}_k\}$ les catégories représentées dans $\mathcal{A}_n$. La probabilité a posteriori que x appartienne à la classe $\mathcal{C}_k$ vaut :

$$p(\mathcal{C}_k|x) = \frac{K_k}{K}, \tag{B.9}$$

en notant K_k le nombre de points parmi les K voisins de x qui appartiennent à la catégorie $\mathcal{C}_k$.

Pour classifier x, il suffit de choisir la classe qui minimise la probabilité d'erreur de classification, et donc qui maximise la probabilité a posteriori. Ceci peut se faire en comptant le nombre de représentant de chaque catégorie parmi les K plus proches voisins de x :

$$g(x) = \operatorname*{argmax}_k p(\mathcal{C}_k|x) = \operatorname*{argmax}_k K_k. \tag{B.10}$$

En cas d'égalité entre catégories, la solution la plus courante est d'en choisir une au hasard.

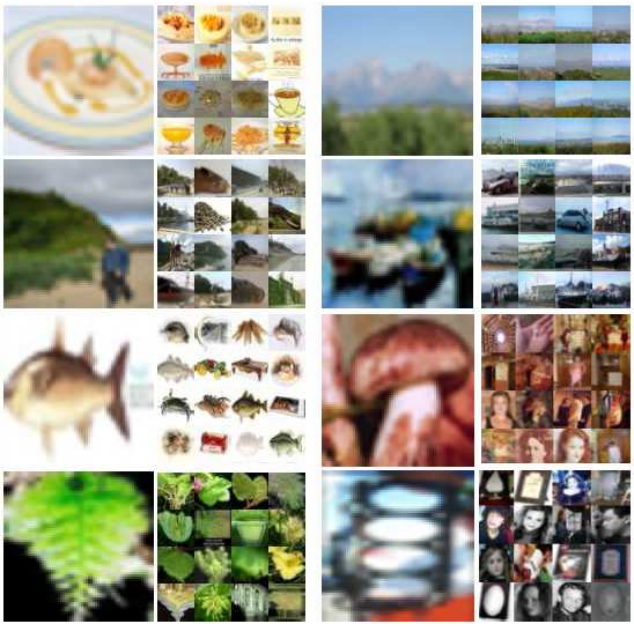


FIGURE B.2 – *Images couleurs de résolution* 32×32*, associées à leurs 16 plus proches voisins. Figure tirée de [178].*

B.3.2 Remarques bibliographiques

Malgré leur naïveté, les K-plus-proches-voisins livrent des résultats très compétitifs, en particulier avec de grandes bases d'images. Makadia et al. [123] proposent une méthode destinée à servir de référence en annotation d'image qui utilise les K-plus-proches-voisins pour assigner des labels aux images. Les voisins sont déterminés et ordonnés par une distance combinant différentes caractéristiques. Les labels sont transférés des voisins à l'image test en deux temps, en s'intéressant prioritairement au plus proche voisin. Torralba et al. [178] explorent les possibilités des K-plus-proches-voisins en utilisant des millions d'images (environ 80×10^6) de très faible résolution (32×32). Les catégories étant organisées hiérarchiquement (vocabulaire WORDNET), le vote est aussi hiérarchique : un nœud vote pour tous ses parents. Avec des millions d'images, ils arrivent à des bonnes performances, et ils montrent que ces performances augmentent avec le nombre d'images n (proportionnellement à $\log(n)$). La figure B.2 montre des images faible résolution et leurs voisins : souvent l'association visuelle correspond à une similarité sémantique, que de simples K-plus-proches-voisins peuvent retrouver. L'idée d'associer les objets à leurs voisins plutôt que de les catégoriser amène Malisiewicz et Efros [124] à s'écarter du schéma classique des K-plus-proches-voisins : au lieu de prendre un nombre fixé de voisins, ils associent une entrée à celles qui sont proches, c'est-à-dire à moins d'une distance donnée. D'une manière générale, la qualité des K-plus-proches-voisins va dépendre de la mesure de distance utilisée (outre la taille de la base). Pour parvenir aux meilleures performances, Torralba et al. [178] calculent la distance sur des images recalées, et à un certain nombre de mouvements de pixels près. Des mesures aussi complexes seraient infaisable sur des images de résolution normale.

B.4 Machines à Vecteurs Supports

B.4.1 Formulation générale

Les Machines à Vecteurs Supports (ou SVMs pour l'anglais *Support Vector Machines*) cherchent à résoudre le problème d'apprentissage par minimisation du risque empirique calculé avec la fonction de coût :

$$\ell(y, y') = [1 - yy']_+ = \max(1 - yy', 0), \tag{B.11}$$

appelée en anglais *hinge loss*. L'utilisation de cette fonction de coût permet d'avoir un problème convexe.

On considère tout d'abord un problème de classification binaire. L'ensemble d'apprentissage est toujours noté $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathcal{L}$, avec, de plus, $\mathcal{L} = \{-1, +1\}$.

B.4.1.1 Espaces vectoriels à noyau reproduisant

Soit $\mathsf{k} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ une fonction symétrique et semi-définie positive, c'est-à-dire

$$\forall J \in \mathbb{N},\ \forall \beta_j \in \mathbb{R},\ \forall x_j \in \mathcal{X},\ \sum_{1 \le j,k \le J} \beta_j \beta_k \mathsf{k}(x_j, x_k) \ge 0, \tag{B.12}$$

ou encore,

$$\forall J \in \mathbb{N},\ \forall x_1, \ldots, x_J \in \mathcal{X},\ \forall u \in \mathbb{R}^J,\ u^T \mathbf{K} u \ge 0, \tag{B.13}$$

où $\mathbf{K} = \big(\mathsf{k}(x_i, x_j)\big)_{1 \le i,j \le J}$.

k s'appelle un *noyau* (ou noyau de Mercer).

Soit $\mathcal{H}$ l'espace vectoriel engendré par les fonctions $\mathsf{k}(x, \cdot)$:

$$\mathcal{H} = vect\{\mathsf{k}(x, \cdot), x \in \mathcal{X}\} \tag{B.14}$$

$$= \{x \mapsto \sum_{j=1}^{J} \beta_j \mathsf{k}(x_j, x); J \in \mathbb{N}, \forall j,\ x_j \in \mathcal{X},\ \beta_j \in \mathbb{R}\}. \tag{B.15}$$

Il peut être muni du produit scalaire défini par : soit $f, f' \in \mathcal{H}$, $\langle \cdot, \cdot \rangle_{\mathcal{H}} : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$,

$$\langle f, f' \rangle = \langle \sum_{1 \le i \le I} \beta_i \mathsf{k}(x_i, \cdot), \sum_{1 \le j \le J} \beta'_j \mathsf{k}(x'_j, \cdot) \rangle_{\mathcal{H}} \tag{B.16}$$

$$= \sum_{i,j} \beta_i \beta'_j \mathsf{k}(x_i, x'_j). \tag{B.17}$$

La fonction k elle-même peut être interprétée comme un produit scalaire sur $\mathcal{H}$ puisque $\langle \mathsf{k}(x, \cdot), \mathsf{k}(x', \cdot) \rangle) = \mathsf{k}(x, x')$.

k est appelé un noyau reproduisant car il vérifie la propriété suivante :

Théorème B.1 *Pour toute fonction $f \in \mathcal{H}$, pour tout x in $\mathcal{X}$,*

$$f(x) = \langle f, \mathsf{k}(x, \cdot) \rangle_{\mathcal{H}}. \tag{B.18}$$

L'espace $\mathcal{H}$ est alors appelé espace vectoriel à noyau reproduisant.

B.4.1.2 Définition des SVMs

Le but est des SVMs est d'estimer une fonction de prédiction de la forme $g = \text{sgn}(f)$ où f est définie dans un espace vectoriel à noyau reproduisant $\mathcal{H}$ muni d'une fonction noyau k en minimisant le coût *hinge loss* :

$$\min_{f \in \mathcal{H}} \lambda ||f||^2_{\mathcal{H}} + \sum_{i=1}^{n} \ell(y_i, f(x_i)), \tag{B.19}$$

où λ est un paramètre de régularisation. f est appelée *fonction de classification*, ou *score* du SVM.

La fonction f étant dans $\mathcal{H}$ prend la forme :

$$f(x) = \sum_{i=1}^{n} \beta_i \mathsf{k}(x_i, x). \tag{B.20}$$

Dans le cas linéaire ($\mathsf{k}(x, x') = \langle x, x' \rangle_{\mathbb{R}}$), et en y ajoutant un offset b, f peut s'écrire $f(x) = \langle w, x \rangle + b$. Le problème non contraint s'écrit alors

$$\min_{w,b} ||w||^2 + C \sum_{i=1}^{n} \ell(y_i, \langle w, x_i \rangle + b), \tag{B.21}$$

où $C = 1/\lambda$.

Dans la littérature, il est généralement présenté sous la forme d'un problème quadratique contraint :

$$\begin{aligned} &\min_{w,b} ||w||^2 \\ &\text{tel que } \forall i \in [n],\ y_i(\langle w, x_i \rangle + b) \geq 1. \end{aligned} \tag{B.22}$$

La fonction f donne l'équation d'un hyperplan H_f séparant les points x_i selon $y_i = -1$ ou $y_i = +1$. La distance d'un point x à H vaut $f(x)/||w||$. Ainsi, la contrainte $y_i f(x_i) \geq 1$ pour tout i est une borne inférieure sur la marge séparant les points "+1" et "-1", qui vaut $2/||w||$, et minimiser $||w||$ revient à maximiser la marge.

Dans le cas où les données ne sont pas séparables, le problème (B.22) n'a pas de solution. Des variables de relaxation sont donc introduites, permettant à certains points de violer la contrainte $y_i(\langle w, x_i \rangle + b) \geq 1$. En pratique, le problème des SVMs s'écrit donc :

$$\begin{aligned} &\min_{w,b} ||w||^2 + C \sum_{i=1}^{n} \xi_i \\ &\text{tel que } \forall i \in [n],\ y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i,\ \xi \geq 0. \end{aligned} \tag{B.23}$$

La variable de relaxation ξ_i mesure de combien l'entrée x_i viole la contrainte. Après résolution du problème, $\xi_i = [1 - y_i f(x_i)]_+ = \ell(y_i, f(x_i))$.

En pratique, la fonction f estimée, sous la forme (B.20), est telle que la plupart des β_i sont nuls : seuls les x_i pour lesquels $\beta_i > 0$ définissent f. Ces vecteurs sont appelés *vecteurs supports*.

B.4.1.3 Cas déséquilibré

Bien que les SVMs permettent de minimiser l'erreur de classification, dans le cas où les données sont déséquilibrées, la frontière favorisela classe dominante – puisque le coût est le même pour les deux classes. Ce genre de situation arrive

souvent – par exemple pour détecter des erreurs. Les erreurs sont peu fréquentes et représentent, par exemple, 5% de la base d'apprentissage. On souhaite pénaliser plus fortement la classification d'une erreur comme correcte que l'inverse. Une manière courante de faire ceci est de définir différemment le coût des faux positifs ℓ_{FP} (dans l'exemple, les erreurs classifiées comme correctes) et le coût des faux négatifs ℓ_{FN}. Les ξ_i seront pondérés différemment selon leur classe, avec, typiquement, $C^+ = C \cdot (1 - P_0)$ et $C^- = C \cdot P_0$ où $P_0 = \ell_{FP}/(\ell_{FP} + \ell_{FN})$:

$$\begin{aligned} &\min_{w,b} ||w||^2 + C^+ \sum_{i,y_i=1} \xi_i + C^- \sum_{i,y_i=-1} \xi_i \\ &\text{tel que } \forall i \in [n],\ y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i,\ \xi \geq 0. \end{aligned} \tag{B.24}$$

Plutôt que de fixer les coûts, il est possible de traiter P_0 comme un paramètre.

B.4.1.4 Cas multiclasse

Nous avons vu la définition des SVMs dans le cas binaire. Il y a plusieurs moyens de traiter le cas multiclasse (K catégories, avec $K \geq 2$). Crammer et Singer [40] proposent une formulation directe des SVMs dans le cas où $\mathcal{L} = [K]$. Cependant ce n'est pas l'approche utilisée en pratique, et nous ne la détaillerons pas ici.

L'autre moyen de traiter le cas multiclasse est de le diviser en un certain nombres de classifications binaires. Ici encore, il y a plusieurs possibilités. Les deux plus courantes sont les approches un-contre-tous et un-contre-un. L'approche un-contre-tous consiste à faire K SVMs binaires où, pour chaque catégorie $\mathcal{C}_k$, le k-ième classifieur prend $y_i^{(k)} = 1$ si $y_i = k$, et $y_i^{(k)} = -1$ sinon. En prenant le vocabulaire des jeux, si chaque catégorie est un joueur, une partie confronte à chaque fois un joueur contre tous les autres réunis. Un exemple x est classifié dans la catégorie donnant le meilleur score $\operatorname{argmax}_k f_k(x)$. Dans l'approche un-contre-un , par contre, chaque joueur est confronté à toutes les autres successivement, et on construit $K(K-1)/2$ SVMs binaires. Chaque classification dans une catégorie $\mathcal{C}_k$ compte comme un vote pour cette catégorie. Un exemple x est classifié dans la catégorie totalisant le plus de votes.

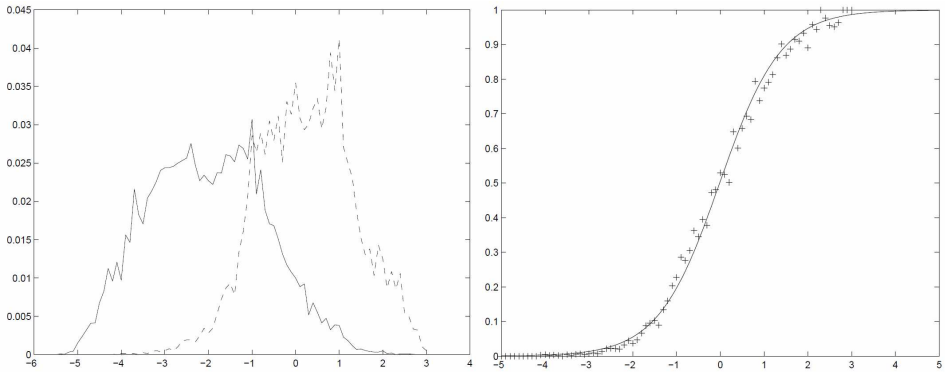
B.4.2 SVM probabilisés

Il est souvent utile d'avoir une information plus riche en sortie de l'algorithme que simplement la catégorie d'appartenance estimée. Le score donné par la fonction f dépend de la distance à la marge et est souvent un bon indicateur de confiance. Il semble naturel d'utiliser ce score pour estimer la probabilité a posteriori : plusieurs travaux s'y intéressent.

B.4.2.1 SVM binaire

Sollich [170] décrit un système permettant d'interpréter les SVM comme des solutions du maximum a posteriori (MAP). La méthode la plus classique est cependant la méthode proposée par Platt [146], qui consiste à approximer la fonction de répartition par une sigmoïde dépendant de la fonction de classification f, sous la forme suivante :

$$P(y = 1|f) = \frac{1}{1 + \exp(Af + B)}. \tag{B.25}$$



(a) (b)

FIGURE B.3 – *La figure B.3a présente les histogrammes obtenus pour $P(f|y=-1)$ (ligne continue) et $P(f|y=1)$ (ligne pointillée) pour un SVM linéaire entraîné une base d'apprentissage (données réelles). La figure B.3b montre la sigmoïde apprise sur la même base. Chaque point marqué "+" correspond à la probabilité a posteriori calculée sur les mêmes intervalles que pour les histogrammes. Figures extraites de [146].*

La figure B.3 montre la forme des densités des distributions $P(f|y=1)$ et $P(f|y=-1)$, et la sigmoïde approchant la probabilité a posteriori $P(y=1|f)$ correspondante.

Les paramètres A et B sont calculés en estimant le maximum de vraisemblance par rapport à un ensemble d'apprentissage (f_i, y_i), en notant $f_i = f(x_i)$. Les sorties y_i sont transformées en probabilités t_i telles que :

$$t_i = \frac{y_i + 1}{2}. \tag{B.26}$$

On cherche à minimiser l'erreur d'entropie croisée :

$$\min -\sum_i t_i \log(p_i) + (1 - t_i)\log(1 - p_i), \tag{B.27}$$

où $p_i = \frac{1}{1+\exp(Af_i+b)} = P(y=1|f_i)$. L'algorithme proposé par Platt a été revu par Lin et al. [118] et est implémenté dans LibSVM (Chang et Lin [36]).

Grandvalet et al. [79] proposent une reformulation des SVMs pour mieux estimer les probabilités a posteriori dans le cas où l'ensemble d'apprentissage est déséquilibré. Soit ℓ_{FN} et ℓ_{FP} les coûts respectifs pour les faux négatifs et les faux positifs. Ayant modifié C^+ et C^- de la même manière qu'en B.4.1.3, ils contraignent de plus la probabilité estimée à être tangente au coût en $f(x) + b = 0$, ce qui les conduit à dévier la marge, et au problème suivant :

$$\begin{aligned}\min_{w,b} ||w||^2 + C \Bigg(& \sum_{i,y_i=+1} [-log(P_0) - (1 - P_0)(\langle w, x_i\rangle + b)]_+ \\ & + \sum_{i,y_i=-1} [-log(1 - P_0) + P_0(\langle w, x_i\rangle + b)]_+ \Bigg)\end{aligned} \tag{B.28}$$

Les performances sont mesurées par un risque R pondérant les erreurs par ℓ_{FN} et ℓ_{FP}.

B.4.2.2 SVM multiclasse

Si on peut estimer les probabilités dans le cas binaire, il n'est pas forcément évident de transposer ces probabilités dans un cas multiclasse. En général, la classification multiclasse par SVM se fait en combinant des SVM binaires. Il est donc

intéressant d'avoir un modèle mathématique permettant de combiner les probabilités issues de tests binaires. Plusieurs solutions sont proposées dans la littérature. Zadrozny [198] utilise des classifieurs Bayésiens Naïfs boostés pour estimer des probabilités calibrées à partir des probabilités binaires obtenues par classification 1-contre-1. Zadrozny et Elkan [199] utilisent les scores des SVM selon un schéma similaire.

Huang et al. [94] proposent d'utiliser des modèles de Bradley-Terry généralisés. Il est assez intuitif de décrire ce modèle avec un vocabulaire sportif. En effet, les modèles de Bradley-Terry sont basés sur des comparaisons deux-à-deux de K individus (ou joueurs). Soit $p_i > 0$ un réel dénotant les capacités d'un joueur i, le modèle suppose que :

$$P(i \text{ bat } j) = \frac{p_i}{p_i + p_j}, \tag{B.29}$$

et que les comparaisons sont indépendantes entre elles. Soit r_{ij} le nombre de fois où i bat j. En notant $\mathbf{p} = \{p_1, \ldots, p_K\}$, l'opposé de la log-vraisemblance vaut alors :

$$l(\mathbf{p}) = -\sum_{i<j}(r_{ij} \log \frac{p_i}{p_i + p_j} + r_{ji} \log \frac{p_j}{p_i + p_j}), \tag{B.30}$$

et il est possible de le minimiser sous la contrainte supplémentaire $\sum_{i=1}^{K} p_i = 1$.

Les auteurs généralisent ce modèle à des groupes de joueurs (ou équipes). Les "parties" ont lieu entre deux équipes I_i^+ et I_i^- et donnent lieu à des résultats enregistrés par r_i et r_i', où r_i est le nombre de fois que I_i^+ bat I_i^- et r_i' le nombre de fois que I_i^- bat I_i^+. Soit m le nombre de parties. Les équipes doivent vérifier pour $i \in [m]$:

$$I_i = I_i^+ \cup I_i^-,\ I_i^+ \neq \emptyset,\ I_i^- \neq \emptyset \text{ et } I_i^+ \cap I_i^- = \emptyset. \tag{B.31}$$

Par ailleurs, on suppose que la capacité d'une équipe est la somme des capacités de ses joueurs. Le modèle est donc le suivant :

$$P(I_i^+ \text{ bat } I_i^-) = \frac{\sum_{j \in I_i^+} p_j}{\sum_{j \in I_i} p_j}. \tag{B.32}$$

On retrouve l'expression à minimiser en posant :

$$q_i \triangleq \sum_{j \in I_i} p_j, \quad q_i^+ \triangleq \sum_{j \in I_i^+} p_j, \quad q_i^- \triangleq \sum_{j \in I_i^-} p_j, \tag{B.33}$$

ce qui donne :

$$l(\mathbf{p}) = -\sum_{i=1}^{m}(r_i \log \frac{q_i^+}{q_i} + r_i' \log \frac{q_i^-}{q_i}), \tag{B.34}$$

tel que $\sum_{j=1}^{K} p_j = 1$, et $\forall j \in [K], p_j \geq 0$.

La solution est obtenue en minimisant $l(\mathbf{p})$ par une procédure itérative. En classification, ce schéma peut être repris directement en interprétant différemment les quantités. Le but est d'estimer pour une entrée x les probabilité $p_k = P(x \in \mathcal{C}_k)$, $k \in [K]$ (plutôt que la *capacité* d'un joueur). La probabilité $P(I_i^+ \text{ bat } I_i^-)$ correspondra à $P(x \in \mathcal{C}_i^+ | x \in \mathcal{C}_i)$, où $\mathcal{C}_i = \{\mathcal{C}_k,\ k \in I_i\}$ et $\mathcal{C}_i^+ = \{\mathcal{C}_k,\ k \in I_i^+\}$.

Les probabilités des parties binaires sont estimées par l'algorithme de Platt (sans que ce soit une contrainte). L'avantage de cette manière d'estimer les probabilités "multiclasse" est qu'elle s'adapte naturellement à n'importe quel type de combinaison de classifieurs binaires.

B.4.3 Implémentation

Quoique le problème quadratique puisse être résolu dans le primal (voir Chapelle [37]), la plupart des algorithmes proposés le résolvent dans le dual, en prenant le lagrangien. Une littérature très nombreuse existe sur ce sujet, et nous ne rentrerons pas dans les détails ici. Nous référons le lecteur à Osuna et al. [140], Platt [147], Joachims [101], Chang et Lin [36] pour les aspects algorithmiques de résolution du problème quadratique.

Les implémentations en C/C++ les plus utilisées sont LibSVM [1] (Chang et Lin [36]) et SVM^{light} [2] (Joachims [101]).

B.4.4 Remarques bibliographiques

Il serait impossible de citer tous les travaux utilisant les SVMs pour l'analyse d'image. Ses bonnes propriétés en ont fait le plus populaire des classifieurs. Avec des caractéristiques images suffisamment riches, un simple classifieur SVM linéaire permet d'avoir des performances très intéressantes. Nous avons déjà vu que Dance et al. [42] introduit les sacs-de-mots en préconisant l'usage de SVMs linéaires, largement plus performants que de simples classifieurs bayésiens naïfs. Vidal-Naquet et Ullman [188] montrent qu'un SVM linéaire fait aussi bien qu'un classifieur plus complexe lorsque les caractéristiques sont sélectionnées pour être les plus informatives possible.

Le choix ap du noyau a aussi une importance majeure. Le noyau linéaire a l'avantage d'être très simple, et rapide à calculer, mais n'est pas forcément le mieux adapté aux données. Barla et al. [13] montre que le noyau intersection d'histogramme, par exemple, est particulièrement intéressant lorsque les images sont représentées par des histogrammes. Lazebnik et al. [113] utilisent aussi le noyau intersection d'histogramme avec les caractéristiques calculées par Spatial Pyramid Matching. Dans ce cas, le noyau intersection d'histogramme bat largement le noyau linéaire. Il est essentiel de déterminer quel est le noyau le mieux adapté aux données pour chaque application.

Les SVMs peuvent encore être combinés avec d'autres classifieurs. Par exemple, Zhang et al. [200] utilisent des SVMs pour affiner la décision de classifieurs K-plus-proches-voisins. Le principe de leur algorithme est illustré figure B.4.

B.5 La régression logistique

B.5.1 Principe

Soit un problème de classification binaire, avec deux classes $\mathcal{C}_1$ et $\mathcal{C}_2$. La probabilité a posteriori de $\mathcal{C}_1$ peut s'écrire comme fonction logistique d'une fonction linéaire du vecteur caractéristique $x \in \mathcal{X}$:

$$p(\mathcal{C}_1|x) = \sigma(\mathbf{w}^T x), \tag{B.35}$$

avec $p(\mathcal{C}_2|x) = 1 - p(\mathcal{C}_1|x)$ et $\sigma(x) = \frac{1}{1+e^{-x}}$. Le nombre de paramètres varie donc linéairement avec le nombre de caractéristiques d (c'est-à-dire la dimension de l'espace $\mathcal{X}$).

Les paramètres sont calculés en utilisant le maximum de vraisemblance.

1. http://www.csie.ntu.edu.tw/~cjlin/libsvm/
2. http://svmlight.joachims.org/

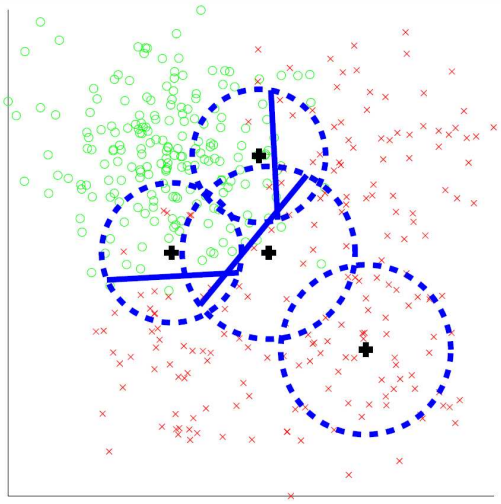


FIGURE B.4 – *Principe des SVM-KNN : un SVM est entrainé sur les 50 plus proches voisins et définit une frontière de décision locale. Figure tirée de [200].*

Soit un ensemble de données $\{x_i, y_i\}$, avec $y_i \in \{0,1\}$,$x_i \in \mathcal{X}$, et $i \in [n]$, la fonction de vraisemblance peut s'écrire :

$$p(\mathbf{y}|\mathbf{w}) = \prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i}, \tag{B.36}$$

avec $\mathbf{y} = (y_1, \ldots, y_n)^T$ et $p_i = p(\mathcal{C}_1|x_i)$. On prend l'opposé du logarithme de cette fonction pour obtenir l'entropie croisée, que l'on cherche à minimiser :

$$l(\mathbf{w}) = -\sum_i y_i \log(p_i) + (1-y_i)\log(1-p_i). \tag{B.37}$$

Cette fonction étant concave, elle offre un minimum global unique. Celui-ci peut être calculé de manière efficace avec un algorithme itératif, type Newton-Raphson.

B.5.2 Lien avec les SVMs

Tout comme les SVMs, la régression logistique est une méthode de classification discriminative. Elle a en outre l'avantage sur les SVMs de présenter une interprétation probabiliste directe. Si on reprend le problème de minimisation du risque empirique (B.19),

$$\min_{f\in\mathcal{H}} \lambda||f||^2_{\mathcal{H}} + \sum_{i=1}^{n} \ell(y_i, f(x_i)),$$

on passe des SVMs à la régression logistique en changeant la fonction de coût. Pour la régression logistique, ℓ est l'*erreur logistique* définie par

$$\ell(y, y') = \log(1 + e^{-yy'}). \tag{B.38}$$

B.5.3 Résolution par IRLS

La régression est généralement résolue par un algorithme itératif des moindres carrés, dont l'équation de mise à jour est :

$$\alpha_{r+1} = \left(\mathbf{\Phi}^T \mathbf{W}_r \mathbf{\Phi} + \lambda \mathbf{R}\right)^{-1} \mathbf{\Phi}^T \mathbf{W}_r \eta_r, \tag{B.39}$$

où :

– le biais b est intégré en ajoutant une dimension dans $\mathbf{\Phi} = [\mathbf{K}\ \mathbb{1}]$,
– $\mathbf{K} = \big(\mathsf{k}(x_i, x_j)\big)_{1\leq i,j\leq n} = \mathbf{X}\mathbf{X}^T$,
– $\lambda\mathbf{R}$ est une terme de régularisation

$$\mathbf{R} = \begin{bmatrix} \mathbf{K} & 0 \\ 0 & 0 \end{bmatrix},$$

– la matrice $\mathbf{W} = \mathrm{diag}(\{\mu_1(1-\mu_1), \ldots, \mu_n(1-\mu_n)\})$, et $\mu_i = \sigma(\mathbf{w}^T x + b)$,
– $\eta_r = \mathbf{X}\mathbf{w}_r + b\mathbb{1} + \mathbf{e}$, où le vecteur $\mathbf{e}$ a pour composantes $e_i = \frac{y_i - \mu_i}{\mu_i(1-\mu_i)}$, $i \in [n]$,
– $\mathbf{w} = \mathbf{X}^T\alpha$.

L'optimisation du vecteur α nécessite de l'inversion de la matrice $H = \mathbf{\Phi}^T\mathbf{W}_r\mathbf{\Phi} + \lambda\mathbf{R}$, qui est symétrique.

B.5.4 Seconde formulation avec noyau

La formulation suivante[3] permet d'appliquer l'algorithme de Newton-Raphson même dans le cas où la matrice de Gram $\mathbf{K}$ est mal conditionnée, et donc que H n'est pas inversible. Pour cela on pose $A = \sqrt{\mathbf{K}}$, puis $\beta_A = A^T\alpha$.

Plus exactement, afin de prendre en compte l'offset b, qui était incorporé précédemment en travaillant sur $\mathbf{\Phi} = [\mathbf{K}\ \mathbb{1}]$. Cette fois-ci, on pose $\mathbf{\Phi} = [A\ \mathbb{1}]$, et $\beta = [\beta_A\ b]^T$.

On cherche à estimer la fonction f :

$$\begin{aligned} f(x) &= \sum_{i=1}^{n} \alpha_i \mathsf{k}(x, xi) + b && \text{(B.40)} \\ &= (\mathbf{K}\alpha)_i + b && \text{(B.41)} \\ &= (AA^T\alpha)_i + b && \text{(B.42)} \\ &= (A\beta_A)_i + b && \text{(B.43)} \\ &= \mathbf{\Phi}\beta && \text{(B.44)} \end{aligned}$$

Le problème à minimiser s'écrit :

$$\begin{aligned} \mathsf{H} &= -\sum_{i=1}^{n} [y_i f(x_i) - \ln(1 + e^{f(x_i)})] + \frac{\lambda}{2}\|f\|^2_{\mathcal{H}_K}, && \text{(B.45)} \\ &= -Y^T\mathbf{\Phi}\beta + \mathbb{1}^T \ln(1 + e^{\mathbf{\Phi}\beta}) + \frac{\lambda}{2}\|\beta\|^2. && \text{(B.46)} \end{aligned}$$

On dérive pour avoir l'optimum, ce qui nous donne les *équations scores* :

$$\begin{aligned} \frac{\partial \mathsf{H}}{\partial \beta} &= -\mathbf{\Phi}^T Y + \mathbf{\Phi}^T\sigma(\mathbf{\Phi}\beta) + \lambda\beta, && \text{(B.47)} \\ &= -\mathbf{\Phi}^T(Y - \mu) + \lambda\beta, && \text{(B.48)} \end{aligned}$$

en posant $\mu = \sigma(\mathbf{\Phi}\beta)$.

On résoud ces équations par l'algorithme de Newton-Raphson. La matrice Hessienne vaut :

$$\frac{\partial^2 \mathsf{H}}{\partial\beta\partial\beta^T} = \mathbf{\Phi}^T W\mathbf{\Phi} + \lambda I \qquad \text{(B.49)}$$

3. Cette formulation n'est pas directement extraite de la littérature, nous l'avons adaptée à partir des formulations sans la racine avant de l'implémenter. (cf 4.3.2)

L'algorithme de Newton-Raphson correspond à l'algorithme 2.

Algorithme 2: Algorithme IRLS pour la régression logistique à noyau

Entrées : $x \in \mathbb{R}^{n \times d}$, $y \in \{0,1\}^n$, fonction noyau k, paramètres λ, ϵ
Sorties : coefficients α et b
Calculer la matrice de Gram $\mathbf{K} = \mathsf{k}(X, X)$;
$A = \sqrt{\mathbf{K}}$;
$\mathbf{\Phi} = [A \; \mathbb{1}]$;
$\alpha_0 = \mathbf{0}$, $b_0 = 0$;
$\beta = [\beta_A \; b]^T$;
$\delta = \inf$, $k = 0$;
tant que $\delta > \epsilon$ **faire**
  Calculer $\mu = \sigma(\mathbf{\Phi} \cdot \beta_k)$;
  $W = \text{diag}(\mu(1-\mu))$;
  Hessienne : $H = \mathbf{\Phi}^T W \mathbf{\Phi} + \lambda I$;
  Gradient : $g = -\mathbf{\Phi}^T (Y - \mu) + \lambda \beta_k$;
  Mettre à jour β : $\beta_{k+1} = \beta_k - H^{-1} g$;
  $k = k + 1$;
$\alpha = (A^T)^{-1} \beta_k(1 : n)$;
$b = \beta_k(n+1)$;

Résultats complémentaires

C

C.1 Signatures

Le tableau C.1 compare le temps de calcul des signatures basées sur les détecteurs avec les SPM. Pour 197 détecteurs, le temps de calcul moyen par image de nos signatures est d'environ 1 minute, ce qui correspond à environ 5 ms par détecteur. Etant donné la relative complexité du système (extraction d'histogrammes, SVM à noyau intersection d'histogrammes), c'est déjà très rapide. Malgré tout, il est possible de réduire encore le temps de calcul, notamment en optimisant la partie classification des fenêtres.

Méthode	**dimension**	**temps de calcul / image**
Détails	197	1,09 mn.
SPM	4200	2,5 s.

Table C.1 – *Temps de calcul moyen des signatures sur les images de la base de voitures.*

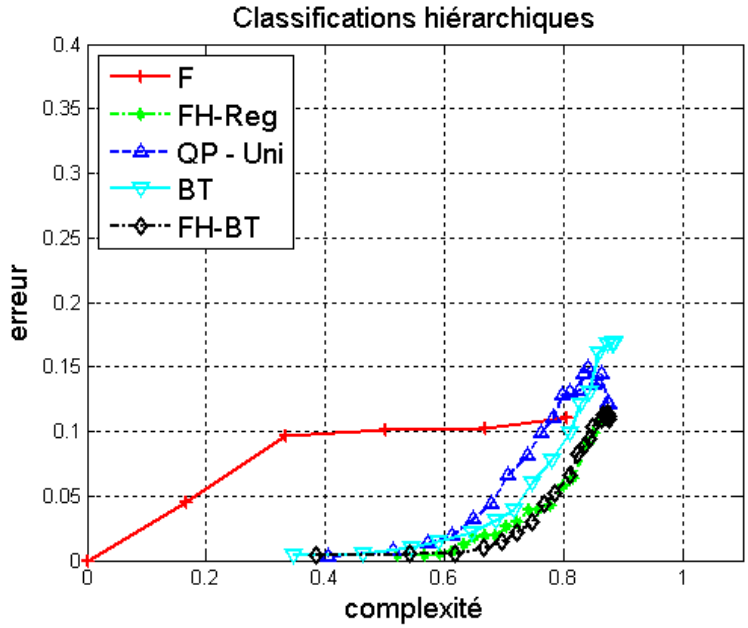
C.2 Annotation multifacette hiérarchique

Le tableau C.2 donne les temps de calcul des annotations par image pour les deux bases. Ce temps dépend de nombreux paramètres : nombre de consets, dimension des vecteurs caractéristiques, noyau, nombre de vecteurs supports. Il est donné pour un calcul non optimisé. Le temps total comprend le temps de classification, de régularisation, ainsi que le calcul des probabilités des nœuds internes et de la courbe erreur/complexité. Comme on pouvait s'y attendre, c'est la classification (SVM) et le calcul des probabilités (Platt) qui sont les plus coûteux. C'est également la partie de calcul la plus évidente à optimiser/paralléliser.

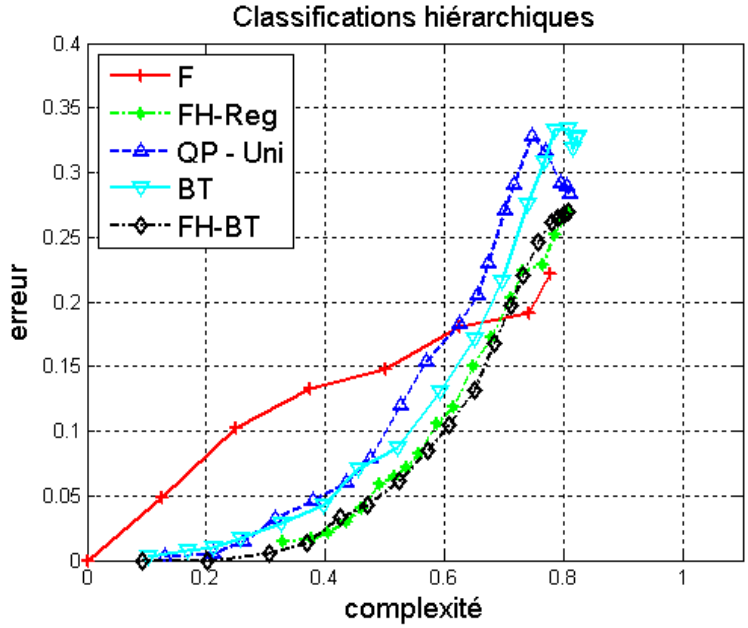
Base	**nombre de consets**	**Total**	**Classification**	**Régularisation**
Voitures	38	15,1 ms.	14,88 ms.	0,13 ms.
Caltech-101	212	2,14 s.	2,13 s.	11,5 ms.

Table C.2 – *Temps de calcul type des annotations (par image, en moyenne).*

C.3 Sélection hiérarchique de caractéristiques

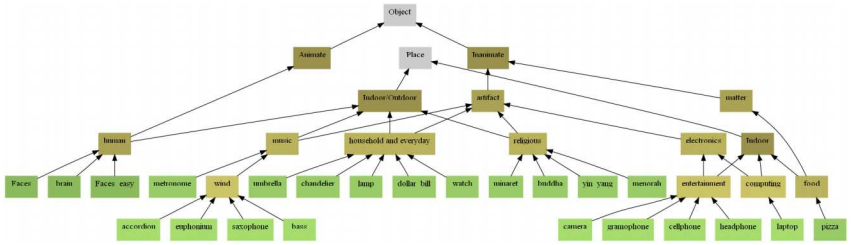


(a) *sous-graphe A*

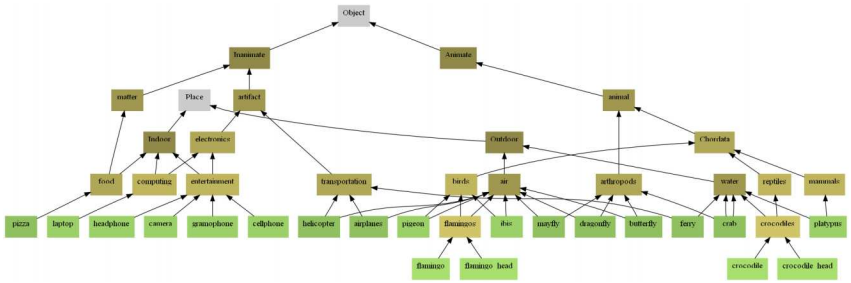


(b) *sous-graphe B*

FIGURE C.1 – *Courbes erreur/complexité obtenues pour quelques sous-graphes du graphe Caltech-101. On remarque les variations de comportement. L'utilisation d'une régularisation sur tous les nœuds du graphe n'est pas avantageuse.*

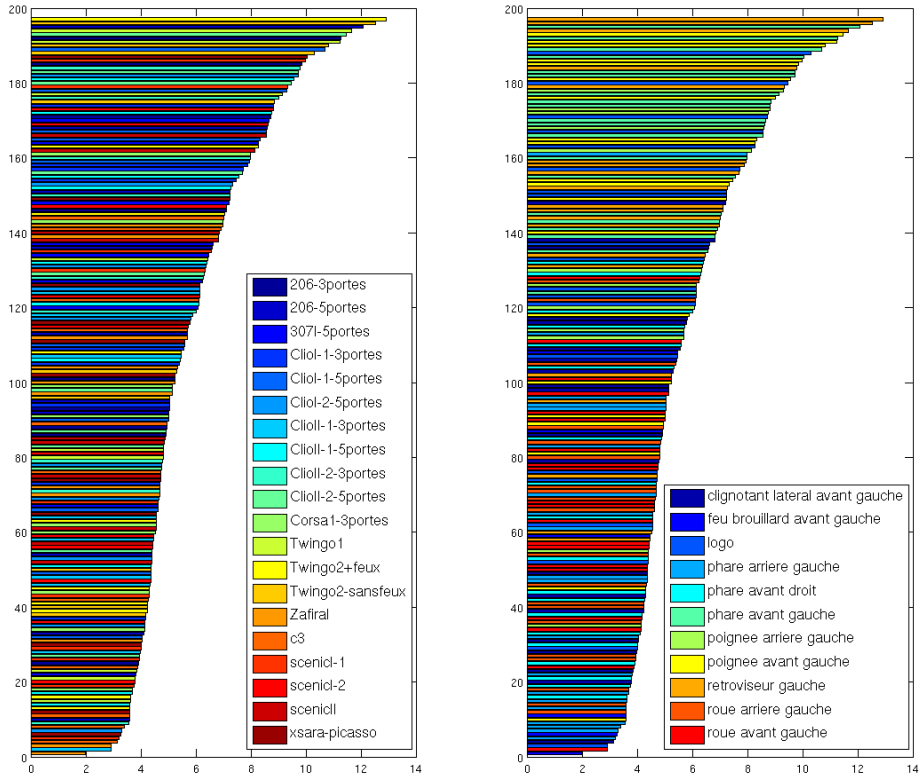


(a) *sous-graphe A*



(b) *sous-graphe B*

FIGURE C.2 – *Sous-graphes* IS-A *extraits du graphe de Caltech-101 utilisés dans les expériences de la section 4.5.4.*



(a) *Par modèles* (b) *Par types*

FIGURE C.3 – *Influence globale de chaque détecteur. Les détecteurs sont ordonnés par rapport à σ calculé par l'équation (5.13), et colorés par modèle (a) et par type de détail (b).*

logo	ClioI-2-5portes
logo	scenicI-1
logo	scenicII
logo	ZafiraI
feu brouillard avant gauche	xsara-picasso
feu brouillard avant gauche	ZafiraI
poignee arriere gauche	ClioI-2-5portes
clignotant lateral avant gauche	c3
clignotant lateral avant gauche	ClioI-1-3portes
clignotant lateral avant gauche	ClioII-1-3portes
clignotant lateral avant gauche	Twing01
clignotant lateral avant gauche	ZafiraI
phare arriere gauche	206-5portes
phare arriere gauche	307I-5portes
phare arriere gauche	c3
phare arriere gauche	ClioI-1-3portes
phare arriere gauche	ClioII-1-5portes
phare arriere gauche	ClioII-2-3portes
phare arriere gauche	ClioII-2-5portes
phare arriere gauche	scenicI-1
phare arriere gauche	Twing02-sansfeux
phare arriere gauche	ZafiraI
poignee avant gauche	c3
poignee avant gauche	Twing02+feux
poignee avant gauche	ZafiraI
retroviseur gauche	ClioII-1-3portes
retroviseur gauche	scenicII
phare avant droit	206-3portes
phare avant droit	scenicI-1
phare avant droit	Twing01
phare avant droit	Twing02+feux
phare avant droit	Twing02-sansfeux
phare avant droit	xsara-picasso
roue avant gauche	307I-5portes
roue avant gauche	c3
roue avant gauche	ClioI-1-3portes
roue avant gauche	scenicI-2
roue avant gauche	scenicII
roue avant gauche	Twing02+feux
roue arriere gauche	c3
roue arriere gauche	ClioI-2-5portes
roue arriere gauche	ClioII-1-3portes
roue arriere gauche	ClioII-2-5portes
roue arriere gauche	Corsa1-3portes
roue arriere gauche	scenicI-1
roue arriere gauche	scenicI-2
roue arriere gauche	Twing02+feux
roue arriere gauche	Twing02-sansfeux
roue arriere gauche	ZafiraI

TABLE C.3 – *Liste des labels des détecteurs qui ne sont sélectionnés pour aucun multilabel avec la méthode* SCORR *et* $K = 20$. *On remarque le nombre important de roues (peu discriminantes) et de phares avant droit et arrière gauches (difficiles à détecter). Aucun phare avant gauche n'apparaît dans cette liste, ce qui montre la pertinence de ce type de détails.*

Glossaire

WordNet Base de données lexicale en langue anglaise, décrivant des liens hiérarchiques entre les mots du vocabulaire. Noms, verbes, adjectifs et adverbes sont regroupés en groupes de synonymes, formant des *synsets*. Ces synsets sont liés entre eux par des relations conceptuelles/sémantiques ou lexicales. WordNet recouvre un vocabulaire très étendu, avec 147 278 mots uniques, et plus de 200 000 synsets [194] .

Annotation Création de métadonnées associées à une image, autrement dit association de données textuelles à l'image, décrivant son contenu sémantique. Généralement associé à la classification multilabel. Correspond souvent à annoter les régions de l'image. Barnard et al. [14] fait la distinction entre *annotation*, lorsque plusieurs labels caractérisent l'image, et *correspondance*, lorsqu'il s'agit d'associer des labels aux différentes parties de l'image.

Catégorisation ou classification ou classification multiclasse. Attribution à l'image d'un unique label parmi une liste de labels donnée..

Classification multilabel Association d'un ensemble de labels (contradictoires ou non) à l'image, ce qui est sensiblement différent de la classification multiclasse.

Correspondance catégorisation des régions de l'image. Parmi les manières d'annoter une image, elle est à distinguer de l'approche multilabel.

Détection Précise si un objet est présent ou non dans une image. Équivalent à une classification binaire objet/fond.

Facette voir point de vue.

Généricité voir aussi spécificité,vocabulaire structuré — Qualité d'un label situé au niveau fondamental et au-dessus. Plus un label se situe haut dans la structure, plus il est dit générique.

Hypernyme Dans une relation Is-A, l'hypernyme est le nœud parent, c'est-à-dire le concept de niveau supérieur. Par exemple, "véhicule" est hypernyme de "voiture".

Hyponyme Dans une relation Is-A, l'hyponyme est le nœud fils. Par exemple, "voiture" est un hyponyme de "véhicule".

Identification Attribution d'un label plus spécifique qu'en catégorisation. Suivant le contexte, il peut s'agir d'une catégorisation en sous-classe (parmi des voitures, une "206 5-portes"), ou de l'identification d'objets particuliers (par exemple "La voiture de M.Dupont"). Nous le réservons à ce dernier usage.

Localisation Recherche de la position et de l'échelle d'un objet dans une image, en général associée à la détection.

Méronyme Dans une relation MEMBER-OF, le méronyme est la partie d'un objet.

Métadonnées Données textuelles décrivant le contenu de l'image. Selon les contextes, les métadonnées peuvent être plus ou moins structurées.

Ontologie Représentation formelle des concepts et des relations existant entre ceux-ci.

Point de vue Sauf exception, nous utilisons le terme de point de vue dans le sens de point de vue descriptif, ou facette.

Reconnaissance d'objets Catégorisation ou Détection ou Identification.

Réseau sémantique Structure de graphe permettant d'encoder les relations entre les concepts et leurs propriétés. Dans ce type de structure, un nœud correspond soit à un concept, soit à une instance de celui-ci (un objet), et une arête correspond à un lien IS-A, HAS-A ou IS-A-KIND-OF, ce dernier type d'arête reliant un nœud objet à un nœud concept.

Spécificité voir aussi généricité,vocabulaire structuré — Qualité d'un label situé sous le niveau fondamental. Dans un vocabulaire structuré, plus un label a d'ancêtres, plus il est spécifique. Opposé à généricité.

Taxonomie Le mot taxonomie ou taxinomie se réfère à l'origine à la structure arborescente utilisée pour la classification du vivant. Plus généralement, ce terme désigne toute structure hiérarchique de type IS-A entre les termes. L'expression "taxonomie visuelle" correspond à un abus de langage et se réfère à une hiérarchie visuelle n'ayant pas nécessairement de signification sémantique.

Thésaurus Un thésaurus est un type de langage documentaire qui consiste en une liste de termes constituant un vocabulaire normalisé sur un domaine de connaissances, reliés entre eux par des relations synonymiques, hiérarchiques et associatives. C'est une sorte de dictionnaire hiérarchisé ; cependant, un thésaurus ne fournit qu'accessoirement des définitions, les relations des termes et leur sélection l'emportant sur la description des significations. (source : Wikipédia, 02/06/09).

Vocabulaire contrôlé voir aussi vocabulaire structuré — Vocabulaire de référence, de taille fixe, utilisé pour l'indexation.

Vocabulaire structuré Vocabulaire contrôlé muni d'une structure, sous forme de thésaurus, de taxonomie, d'ontologie ou plus généralement de réseau sémantique..

Index

Caltech-101, 24, 35, 41, 51, 53, 111, 117

Diagramme de Hasse, 81, 85, 88

Généricité, 9
Gist, 49, 53

Hypergraphe, 81, 88

Métadonnées, 8, 9
Bradley-Terry, modèles de, 94, 98, 107, 111, 169

algorithme de Platt, 83, 90, 106, 167

Régression logistique, 84, 90, 106, 170

Spécificité, 9, 81
SVM, 106, 163, 165
SVM
  un-contre-tous, 48, 74, 167
  un-contre-un, 167
  binaire, 83, 84, 90, 128, 166
  multiclasse, 27, 83, 167
  one-class, 68, 71

Vocabulaire Contrôlé, 17, 18, 29, 82

WordNet, 29, 31, 33, 35, 42

Bibliographie

[1] Emile H. L. Aarts, Jan H. M. Korst, et Peter J.M. van Laarhoven. Simulated annealing. volume Local Search in Combinatorial Optimization de *Discrete Mathematics and Optimization*, Chapitre 4, pages 91–120. Wiley-Interscience, Chichester, England, Juin 1997.

[2] Y. Abramson et Y. Freund. Semi-automatic visual learning (seville) : a tutorial on active learning for visual object recognition. Dans *CVPR'05 : Intl. Conf. on Computer Vision and Pattern Recognition, San Diego*, 2005.

[3] S. Agarwal, A. Awan, et D. Roth. Learning to detect objects in images via a sparse, part-based representation. *PAMI*, 26(11) :1475–1490, November 2004.

[4] N. Ahuja et S. Todorovic. Learning the taxonomy and models of categories present in arbitrary images. Dans *ICCV07*, pages 1–8, 2007.

[5] S. Aksoy, K. Koperski, C. Tusk, G. Marchisio, et J.C. Tilton. Learning bayesian classifiers for scene classification with a visual grammar. *GeoRS*, 43(3) : 581–589, March 2005.

[6] Yonatan Amit, Michael Fink, Nathan Srebro, et Shimon Ullman. Uncovering shared structures in multiclass classification. Dans *ICML '07*, pages 17–24, New York, NY, USA, 2007. ACM Press.

[7] O.D. Arandjelovic et A. Zisserman. Automatic face recognition for film character retrieval in feature-length films. Dans *CVPR05*, pages I : 860–867, 2005.

[8] Linda H. Armitage et Peter G.B. Enser. Analysis of user need in image archives. *Journal of Information Science*, 24(4) :287–299, 1997.

[9] Y. Alp Aslandogan, Chuck Thier, Clement T. Yu, Jon Zou, et Naphtali Rishe. Using semantic contents and wordnet in image retrieval. *SIGIR Forum*, 31 (SI) :286–295, 1997.

[10] Simon Baker et Iain Matthews. Lucas-kanade 20 years on : A unifying framework. *International Journal of Computer Vision*, 56(3) :221 – 255, March 2004.

[11] Gökhan H. Bakir, Thomas Hofmann, Bernhard Schölkopf, Alexander J. Smola, Ben Taskar, et S. V. N. Vishwanathan. *Predicting Structured Data (Neural Information Processing)*. The MIT Press, 2007.

[12] Aharon Bar-Hillel et Daphna Weinshall. Subordinate class recognition using relational object models. Dans *NIPS'06*, pages 73–80, 2006.

[13] A. Barla, F. Odone, et A. Verri. Histogram intersection kernel for image classification. Dans *ICIP03*, volume 3, pages 513–516, 2003.

[14] K. Barnard, P. Duygulu, D.A. Forsyth, N. de Freitas, D.M. Blei, et M.I. Jordan. Matching words and pictures. *JMLR*, 3 :1107–1135, 2003.

[15] K. Barnard et D. Forsyth. Learning the semantics of words and pictures. Dans *ICCV'01 : International Conference on Computer Vision*, volume 2, pages 408–415. Vancouver : IEEE, 2001.

[16] Kobus Barnard, Pinar Duygulu, et David Forsyth. Clustering art. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2 :434, 2001.

[17] Kobus Barnard, Quanfu Fan, Ranjini Swaminathan, Anthony Hoogs, Roderic Collins, Pascale Rondot, et John Kaufhold. Evaluation of localized semantics : Data, methodology, and experiments. *International Journal of Computer Vision*, 77(1-3) :199–217, Aug 2008.

[18] E. Bart, I. Porteous, P. Perona, et M. Welling. Unsupervised learning of visual taxonomies. Dans *CVPR08*, pages 1–8, 2008.

[19] Zafer Barutcuoglu, Robert E. Schapire, et Olga G. Troyanskaya. Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7) :830–836, 2006.

[20] C. Berge. *Graphes et Hypergraphes*. Dunod, Paris, 1977.

[21] I. Biederman. Recognition-by-components : A theory of human image understanding. *Psychological review*, 94(2) :115–147, 1987.

[22] Alexander Binder, Motoaki Kawanabe, et Ulf Brefeld. Efficient classification of images with taxonomies. Dans *ACCV'09*, 2009.

[23] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, August 2006.

[24] Avrim L. Blum et Pat Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1–2) :245–271, 1997.

[25] Jean-Baptiste Bordes. *Inférence de connaissances sémantiques : application aux images satellitaires*. PhD thesis, Télécom ParisTech, 2009.

[26] A. Bosch, A. Zisserman, et X. Munoz. Image classification using random forests and ferns. Dans *ICCV'07 : Proceedings of the 11th International Conference on Computer Vision, Rio de Janeiro, Brazil*, 2007.

[27] Anna Bosch, Xavier Muñoz, et Robert Martí. Which is the best way to organize/classify images by content ? *Image and Vision Computing*, 25(6) :778 – 791, 2007.

[28] Anna Bosch, Andrew Zisserman, et Xavier Munoz. Representing shape with a spatial pyramid kernel. Dans *CIVR '07 : Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 401–408, New York, NY, USA, 2007. ACM.

[29] S. Boughorbel, J.P. Tarel, et N. Boujemaa. Generalized histogram intersection kernel for image recognition. Dans *ICIP05*, pages III : 161–164, 2005.

[30] Matthew R. Boutell, Jiebo Luo, Xipeng Shen, et Christopher M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9) :1757–1771, Septembre 2004.

[31] Lijuan Cai et Thomas Hofmann. Hierarchical document categorization with support vector machines. Dans *CIKM '04 : Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 78–87, New York, NY, USA, 2004. ACM Press.

[32] Gustavo Carneiro, Antoni B. Chan, Pedro J. Moreno, et Nuno Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *PAMI*, 29(3) :394–410, 2007.

[33] V. Černỳ. Thermodynamical approach to the traveling salesman problem : An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1) :41–51, 1985.

[34] Nicolò Cesa-Bianchi, Claudio Gentile, et Luca Zaniboni. Hierarchical classification : combining bayes with svm. Dans *ICML '06*, pages 177–184, New York, NY, USA, 2006. ACM Press.

[35] Nicolo Cesa-Bianchi, Claudio Gentile, et Luca Zaniboni. Incremental algorithms for hierarchical classification. *JMLR*, 7 :31–54, 2006.

[36] Chih-Chung Chang et Chih-Jen Lin. *LIBSVM : a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[37] Olivier Chapelle. Training a support vector machine in the primal. *Neural Computation*, 19(5) :1155–1178, 2007.

[38] V. Cordì, P. Lombardi, M. Martelli, et V. Mascardi. An ontology-based similarity between sets of concepts. Dans F. Corradini, F. De Paoli, E. Merelli, et A. Omicini, éditeurs, *WOA 2005 : Dagli Oggetti agli Agenti. 6th AI*IA/TABOO Joint Workshop "From Objects to Agents"*, pages 16–21. Pitagora Editrice Bologna, 2005.

[39] Thomas M. Cover et Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.

[40] Koby Crammer et Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *J. Mach. Learn. Res.*, 2 :265–292, 2002.

[41] Navneet Dalal et Bill Triggs. Histograms of oriented gradients for human detection. Dans Cordelia Schmid, Stefano Soatto, et Carlo Tomasi, éditeurs, *International Conference on Computer Vision & Pattern Recognition*, volume 2, pages 886–893, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005.

[42] Chris Dance, Jutta Willamowski, Lixin Fan, Cedric Bray, et Gabriela Csurka. Visual categorization with bags of keypoints. Dans *ECCV'04 : International Workshop on Statistical Learning in Computer Vision*, 2004.

[43] Ritendra Datta, Weina Ge, Jia Li, et James Z. Wang. Toward bridging the annotation-retrieval gap in image search by a generative modeling approach. Dans *MULTIMEDIA'06 : Proceedings of the ACM Multimedia Conference*, pages 977–986, Santa Barbara, CA, oct 2006. ACM, ACM Press.

[44] Ritendra Datta, Weina Ge, Jia Li, et James Z. Wang. Toward bridging the annotation-retrieval gap in image search. *IEEE MultiMedia*, 14(3) :24–35, 2007.

[45] Ritendra Datta, Dhiraj Joshi, Jia Li, et James Z. Wang. Image retrieval : Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40(2) :1–60, 2008.

[46] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, et L. Fei-Fei. Imagenet : A large-scale hierarchical image database. Dans *CVPR09*, 2009.

[47] P. Dollar, B. Babenko, S.J. Belongie, P. Perona, et Z.W. Tu. Multiple component learning for object detection. Dans *ECCV'08*, pages II : 211–224, 2008.

[48] Matthijs Douze, Hervé Jégou, Harsimrat Singh, Laurent Amsaleg, et Cordelia Schmid. Evaluation of gist descriptors for web-scale image search. Dans *International Conference on Image and Video Retrieval*. ACM, july 2009.

[49] P. Duygulu, Kobus Barnard, J. F. G. de Freitas, et David A. Forsyth. Object recognition as machine translation : Learning a lexicon for a fixed image vocabulary. Dans *ECCV '02 : Proceedings of the 7th European Conference on Computer Vision-Part IV*, pages 97–112, London, UK, 2002. Springer-Verlag.

[50] John P. Eakins. Towards intelligent image retrieval. *Pattern Recognition*, 35 (1) :3 – 14, 2002.

[51] John P. Eakins, Pamela Briggs, et Bryan Burford. Image retrieval interfaces : A user perspective. Dans *CIVR'04*, pages 628–637, 2004.

[52] Peter Enser. Visual image retrieval : seeking the alliance of concept-based and content-based paradigms. *JIS : Journal of Information Science*, 26(4) :199–210, 2000.

[53] Peter G. B. Enser et Christine J. Sandom. Towards a comprehensive survey of the semantic gap in visual image retrieval. Dans *CIVR'03*, pages 291–299, 2003.

[54] B. Epshtein et S. Ullman. Feature hierarchies for object classification. Dans *ICCV05*, pages I : 220–227, 2005.

[55] Boris Epshtein et Shimon Ullman. Satellite features for the classification of visually similar classes. Dans *CVPR '06*, pages 2079–2086, Washington, DC, USA, 2006. IEEE Computer Society.

[56] Boris Epshtein et Shimon Ullman. Semantic hierarchies for recognizing objects and parts. Dans *CVPR'07 : IEEE Conference on Computer Vision and Pattern Recognition, 2007.*, pages 1–8, 2007.

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

[58] Facebook. Statistics, 2009. http://www.facebook.com/press/info.php?statistics.

[59] Jianping Fan, Yuli Gao, et Hangzai Luo. Hierarchical classification for automatic image annotation. Dans *SIGIR '07 : Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 111–118, New York, NY, USA, 2007. ACM.

[60] Jianping Fan, Yuli Gao, Hangzai Luo, et Ramesh Jain. Mining multilevel image semantics via hierarchical classification. *IEEE Transactions on Multimedia*, 10(2) :167–187, 2008.

[61] J.P. Fan, Y. Gao, et H.Z. Luo. Integrating concept ontology and multitask learning to achieve more effective classifier training for multilevel image annotation. *IP*, 17(3) :407–426, March 2008.

[62] Xiaodong Fan et Donald Geman. Hierarchical object indexing and sequential learning. Dans *ICPR (3)*, pages 65–68, 2004.

[63] Li Fei-Fei, Rob Fergus, et Pietro Perona. A bayesian approach to unsupervised one-shot learning of object categories. Dans *ICCV '03*, Washington, DC, USA, 2003. IEEE Computer Society.

[64] Li Fei-Fei, Rob Fergus, et Pietro Perona. Learning generative visual models from few training examples : An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1) :59 – 70, 2007. Special issue on Generative Model Based Vision.

[65] Li Fei-Fei et Pietro Perona. A bayesian hierarchical model for learning natural scene categories. Dans *CVPR '05 : Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 524–531, Washington, DC, USA, 2005. IEEE Computer Society.

[66] Christiane Fellbaum, éditeur. *WordNet : An Electronic Lexical Database*. The MIT Press, Cambridge, MA ; London, Mai 1998.

[67] S.L. Feng, R. Manmatha, et V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. Dans *CVPR'04. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004.*, volume 2, pages II–1002–II–1009 Vol.2, June-2 July 2004.

[68] Andras Ferencz, Erik G. Learned-Miller, et Jitendra Malik. Learning to locate informative features for visual identification. *Int. J. Comput. Vision*, 77(1-3) : 3–24, 2008.

[69] R. Fergus, P. Perona, et A. Zisserman. Object class recognition by unsupervised scale-invariant learning. *CVPR*, 02 :264, 2003.

[70] Vittorio Ferrari, Loic Fevrier, Frederic Jurie, et Cordelia Schmid. Groups of adjacent contour segments for object detection. *PAMI*, 30(1) :36–51, Jan 2007.

[71] S. Fidler et A. Leonardis. Towards scalable representations of object categories : Learning a hierarchy of parts. Dans *CVPR07*, pages 1–8, 2007.

[72] François Fleuret. Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 5 :1531–1555, Novembre 2004.

[73] C. Olivia Frost, Bradley Taylor, Anna Noakes, Stephen Markel, Deborah Torres, et Karen M. Drabenstott. Browse and search patterns in a digital image database. *Inf. Retr.*, 1(4) :287–313, 2000.

[74] Yuli Gao et Jianping Fan. Incorporating concept ontology to enable probabilistic concept reasoning for multi-level image annotation. Dans *MIR'06 : Multimedia Information Retrieval*, pages 79–88, 2006.

[75] Lars Marius Garshol. Metadata ? Thesauri ? Taxonomies ? Topic Maps ! Making Sense of it all. *Journal of Information Science*, 30(4) :378–391, 2004.

[76] Christian Gerlach. Category-specificity in visual object recognition. *Cognition*, 111(3) :281 – 301, 2009.

[77] Alan Gilchrist. Thesauri, taxonomies and ontologies – an etymological note. *Journal of Documentation*, 59(1) :7–18, 2003.

[78] Philippe Henri Gosselin, Matthieu Cord, et Sylvie Philipp-Foliguet. Combining visual dictionary, kernel-based similarity and learning strategy for image category retrieval. *Computer Vision and Image Understanding*, 110(3) : 403–417, Juin 2008.

[79] Yves Grandvalet, Johnny Mariéthoz, et Samy Bengio. A probabilistic interpretation of svms with an application to unbalanced classification. Dans *NIPS'05*, 2005. IDIAP-RR 05-26.

[80] Kristen Grauman et Trevor Darrell. The pyramid match kernel : Discriminative classification with sets of image features. Dans *ICCV '05 : Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1458–1465, Washington, DC, USA, 2005. IEEE Computer Society.

[81] G. Griffin, A. Holub, et P. Perona. Caltech-256 object category dataset. Rapport Technique 7694, California Institute of Technology, 2007.

[82] Gregory Griffin et Pietro Perona. Learning and using taxonomies for fast visual categorization. Dans *CVPR'08 : IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[83] Abhinav Gupta et Larry S. Davis. Beyond nouns : Exploiting prepositions and comparative adjectives for learning visual classifiers. Dans *ECCV'08*, pages 16–29, 2008.

[84] Isabelle Guyon et André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3 :1157–1182, Mars 2003.

[85] Mark A. Hall. *Correlation-based Feature Selection for Machine Learning*. PhD thesis, Waikato University, Mai 17 1999.

[86] Allan Hanbury. A survey of methods for image annotation. *Journal of Visual Languages & Computing*, 19(5) :617 – 627, 2008.

[87] Jonathon S. Hare, Paul H. Lewis, Peter G. B. Enser, et Christine J. Sandom. Mind the gap : Another look at the problem of the semantic gap in image retrieval. Dans Edward Y. Chang, Alan Hanjalic, et Nicu Sebe, éditeurs, *Multimedia Content Analysis, Management and Retrieval 2006*, volume SPIE Vol. 6073, pages 607309–1. SPIE and IS&T, 2006.

[88] J.S. Hare, P.H. Lewis, P.G.B. Enser, et C.J. Sandom. A linear-algebraic technique with an application in semantic image retrieval. Dans *CIVR'06 : 5th International Conference on Image and Video Retrieval, Tempe, AZ, USA, July 2006*, pages 31–40. Springer-Verlag Heidelberg, 2006.

[89] Trevor Hastie, Robert Tibshirani, et Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[90] Jay Hegdé, Evgeniy Bart, et Daniel Kersten. Fragment-based learning of visual object categories. *Current Biology*, 18(8) :597 – 601, 2008.

[91] Stéphane Herbin, Frédéric Champagnat, Guy Le Besnerais, Anne-Marie Tousch, Antoine Létienne, et Jonathan Guinet. Trois modalités de description des véhicules dans les vidéos. *Revue de l'Electricité et de l'Electronique*, 9 : 41–48, Octobre 2009.

[92] L. Hollink, G. Schreiber, B. Wielemaker, et B. Wielinga. Semantic annotation of image collections. Dans *KCAP'03 : Workshop on Knowledge Markup and Semantic Annotation*, Florida, USA, October 2003.

[93] L. Hollink, G. Schreiber, B.J. Wielinga, et M. Worring. Classification of user image descriptions. *Int. J. Hum.-Comput. Stud.*, 61(5) :601–626, 2004.

[94] Tzu K. Huang, Ruby C. Weng, et Chih J. Lin. Generalized bradley-terry models and multi-class probability estimates. *J. Mach. Learn. Res.*, 7 :85–115, 2006.

[95] Dionysius P. Huijsmans. How to complete performance graphs in content-based image retrieval : Add generality and normalize scope. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(2) :245–251, 2005. Member-Nicu Sebe.

[96] A. Jaimes et S.-F. Chang. Conceptual framework for indexing visual information at multiple levels. Dans G. B. Beretta et R. Schettini, éditeurs, *IS&T/SPIE Internet Imaging*, volume 3964 de *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 2–15, jan 2000.

[97] Alejandro Jaimes. Human factors in automatic image retrieval system design and evaluation. Dans Simone Santini, Raimondo Schettini, et Theo Gevers, éditeurs, *Internet Imaging VII*, volume 6061, page 606103. SPIE, 2006.

[98] J. Jeon, V. Lavrenko, et R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. Dans *SIGIR '03 : Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 119–126, New York, NY, USA, 2003. ACM.

[99] Y. Jin et S. Geman. Context and hierarchy in a probabilistic image model. Dans *CVPR06*, pages II : 2145–2152, 2006.

[100] Yohan Jin, Latifur Khan, Lei Wang, et Mamoun Awad. Image annotations by combining multiple evidence & wordnet. Dans *MULTIMEDIA '05 : Proceedings of the 13th annual ACM international conference on Multimedia*, pages 706–715, New York, NY, USA, 2005. ACM.

[101] Thorsten Joachims. Making large-scale SVM learning practical. Dans Bernhard Schölkopf, Christopher J. Burges, et Alexander J. Smola, éditeurs, *Advances in Kernel Methods – Support Vector Learning*, Chapitre 11, pages 169–184. The MIT Press, Cambridge, US, 1999.

[102] Pierre Jolicoeur, Mark A. Gluck, et Stephen M. Kosslyn. Pictures and names : Making the connection. *Cognitive Psychology*, 16(2) :243 – 275, 1984.

[103] Dhiraj Joshi, James Z. Wang, et Jia Li. The story picturing engine : finding elite images to illustrate a story using mutual reinforcement. Dans *MIR '04 : Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pages 119–126, New York, NY, USA, 2004. ACM.

[104] Corinne Jörgensen. Attributes of images in describing tasks. *Information Processing & Management*, 34(2-3) :161 – 174, 1998.

[105] Frederic Jurie et Bill Triggs. Creating efficient codebooks for visual recognition. Dans *ICCV'05 : International Conference on Computer Vision*, 2005.

[106] Tom Kelly. Revealed : Big brother britain has more cctv cameras than china, August 2009. Daily Mail, http://www.dailymail.co.uk/news/article-1205607/Shock-figures-reveal-Britain-CCTV-camera-14-people--China.html.

[107] S. Kirkpatrick, C. D. Gelatt, et M. P. Vecchi. Optimization by simulated annealing. *Science, Number 4598, 13 May 1983*, 220, 4598 :671–680, 1983.

[108] Ron Kohavi et George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2) :273 – 324, 1997. Relevance.

[109] A. Kushal, C. Schmid, et J. Ponce. Flexible object models for category-level 3d object recognition. Dans *CVPR07*, pages 1–8, 2007.

[110] P. J. M. Laarhoven et E. H. L. Aarts. *Simulated annealing : theory and applications*. Kluwer Academic Publishers, 1987.

[111] T. Lam et R. Singh. Semantically relevant image retrieval by combining image and linguistic analysis. Dans *ISVC06*, pages II : 770–779, 2006.

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

[113] S. Lazebnik, C. Schmid, et J. Ponce. Beyond bags of features : Spatial pyramid matching for recognizing natural scene categories. Dans *CVPR06*, pages II : 2169–2178, 2006.

[114] B. Leibe, A. Ettlin, et B. Schiele. Learning semantic object parts for object categorization. *IVC*, 26(1) :15–26, January 2008.

[115] K. Levi, M. Fink, et Y. Weiss. Learning from a small number of training examples by exploiting object categories. Dans *LCV04*, page 96, 2004.

[116] Jia Li et J.Z. Wang. Real-time computerized annotation of pictures. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(6) :985–1002, June 2008.

[117] L-J. Li, R. Socher, et L. Fei-Fei. Towards total scene understanding :classification, annotation and segmentation in an automatic framework. Dans *CVPR'09 : Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2009.

[118] Hsuan-Tien Lin, Chih-Jen Lin, et Ruby Weng. A note on platt ś probabilistic outputs for support vector machines. *Machine Learning*, 68(3) :267–276, October 2007.

[119] Song Liu, Liang-Tien Chia, et Syin Chan. On the move to meaningful internet systems 2004 : Coopis, doa, and odbase. volume 3291 de *Lecture Notes in Computer Science*, Chapitre Ontology for Nature-Scene Image Retrieval, pages 1050–1061. Springer Berlin / Heidelberg, Oct 2004.

[120] Ying Liu, Dengsheng Zhang, Guojun Lu, et Wei-Ying Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1) :262 – 282, 2007.

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

[122] Nicolas Eric Maillot et Monique Thonnat. Ontology based complex object recognition. *Image and Vision Computing*, 26(1) :102 – 113, 2008. Cognitive Vision-Special Issue.

[123] Ameesh Makadia, Vladimir Pavlovic, et Sanjiv Kumar. A new baseline for image annotation. Dans *ECCV '08 : Proceedings of the 10th European Conference on Computer Vision*, pages 316–329, Berlin, Heidelberg, 2008. Springer-Verlag.

[124] Tomasz Malisiewicz et Alexei A. Efros. Recognition by association via learning per-exemplar distances. Dans *CVPR*, June 2008.

[125] Marjo Markkula et Eero Sormunen. End-user searching challenges indexing practices inthe digital newspaper photo archive. *Inf. Retr.*, 1(4) :259–285, 2000.

[126] Marcin Marszałek et Cordelia Schmid. Semantic hierarchies for visual object recognition. Dans *CVPR'07*, jun 2007.

[127] Marcin Marszałek et Cordelia Schmid. Constructing category hierarchies for visual recognition. Dans *ECCV'08 : European Conference on Computer Vision*, volume IV de *LNCS*, pages 479–491. Springer, oct 2008.

[128] Krystian Mikolajczyk et Cordelia Schmid. A performance evaluation of local descriptors. *PAMI*, 27(10) :1615–1630, 2005.

[129] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, et L. Van Gool. A comparison of affine region detectors. *IJCV*, 65(1/2) :43–72, 2005.

[130] Anuj Mohan, Constantine Papageorgiou, et Tomaso Poggio. Example-based object detection in images by components. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(4) :349–361, 2001.

[131] Aleksandra Mojsilovic, José Gomes, et Bernice Rogowitz. Semantic-friendly indexing and quering of images based on the extraction of the objective semantic cues. *Int. J. Comput. Vision*, 56(1-2) :79–107, 2004.

[132] Frank Moosmann, Bill Triggs, et Frederic Jurie. Fast discriminative visual codebooks using randomized clustering forests. Dans B. Schölkopf, J. Platt, et T. Hoffman, éditeurs, *NIPS'07 : Advances in Neural Information Processing Systems 19*, pages 985–992. MIT Press, Cambridge, MA, 2007.

[133] Emily Moxley, Jim Kleban, et B.S. Manjunath. Spirittagger : A geo-aware tag suggestion tool mined from flickr. Dans *ACM International Conference on Multimedia Information Retrieval (MIR2008)*, Oct 2008.

[134] J.L. Mundy. Object recognition in the geometric era : A retrospective. Dans *CLOR06*, pages 3–28, 2006.

[135] Gregory L. Murphy et Edward E. Smith. Basic-level superiority in picture categorization. *Journal of Verbal Learning and Verbal Behavior*, 21(1) :1 – 20, 1982.

[136] M.-E. Nilsback et A. Zisserman. Automated flower classification over a large number of classes. Dans *Computer Vision, Graphics & Image Processing, 2008. ICVGIP '08. Sixth Indian Conference on*, pages 722–729, Dec. 2008.

[137] Eric Nowak, Frederic Jurie, et Bill Triggs. Sampling strategies for bag-of-features image classification. Dans *ECCV'06*. Springer, 2006.

[138] A. Oliva et A. Torralba. Modeling the shape of the scene : A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42 (3) :145–175, Mai 2001.

[139] A. Oliva et A. Torralba. Building the gist of a scene : the role of global image features in recognition. *Progress in brain research*, 155 :23–36, 2006.

[140] Edgar Osuna, Robert Freund, et Federico Girosi. Support vector machines : Training and applications. Rapport Technique AIM-1602, Aritificial Intelligence Laboratory, MIT, 1997.

[141] G. Th. Papadopoulos, V. Mezaris, Dasiopoulou S., et Kompatsiaris I. Semantic image analysis using a learning approach and spatial context. Dans *SAMT 2006 : 1st International Conference on Semantics And Digital Media Technologies,*, 2006.

[142] D. Parikh et T.H. Chen. Hierarchical semantics of objects (hsos). Dans *ICCV07*, pages 1–8, 2007.

[143] F. Perronnin. Universal and adapted vocabularies for generic visual categorization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30 (7) :1243–1256, July 2008.

[144] M. T. Pham et T. J. Cham. Online learning asymmetric boosted classifiers for object detection. Dans *CVPR*, pages 1–8, 2007.

[145] Axel Pinz. Object categorization. *Foundations and Trends in Computer Graphics and Vision*, 1(4) :255–353, 2005.

[146] J. Platt. Probabilistic outputs for support vector machines and comparison to regularize likelihood methods. Dans *Advances in Large Margin Classifiers*, Chapitre 5, pages 61–74. MIT Press, 2000.

[147] John C. Platt. Fast training of support vector machines using sequential minimal optimization. pages 185–208, 1999.

[148] Jean Ponce, Tamara L. Berg, Mark Everingham, David A. Forsyth, Martial Hebert, Svetlana Lazebnik, Marcin Marszalek, Cordelia Schmid, Bryan C. Russell, Antonio B. Torralba, Christopher K. I. Williams, Jianguo Zhang, et Andrew Zisserman. Dataset issues in object recognition. Dans Jean Ponce, Martial Hebert, Cordelia Schmid, et Andrew Zisserman, éditeurs, *Toward Category-Level Object Recognition*, volume 4170 de *Lecture Notes in Computer Science*, pages 29–48. Springer, 2006.

[149] Jean Ponce, Martial Hebert, Cordelia Schmid, et Andrew Zisserman. *Toward Category-Level Object Recognition (Lecture Notes in Computer Science)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[150] Adrian Popescu, Christophe Millet, et Pierre-Alain Moëllic. Ontology driven content based image retrieval. Dans *CIVR'07*, pages 387–394, 2007.

[151] F. Porikli. Integral histogram : a fast way to extract histograms in cartesian spaces. Dans *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 829–836 vol. 1, June 2005.

[152] Shiyali Ramamrita Ranganathan. *Classification, coding and machinery for search*. UNESCO, 1950.

[153] Manjeet Rege, Ming Dong, et Farshad Fotouhi. Building a user-centered semantic hierarchy in image databases. *Multimedia Systems*, 12(4-5) :325–338, March 2007.

[154] P. Resnik. Semantic similarity in a taxonomy : An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11 :95–130, 1999.

[155] Eleanor Rosch, Carolyn B. Mervis, Wayne D. Gray, David M. Johnson, et Penny Boyes-Braem. Basic objects in natural categories. *Cognitive Psychology*, 8(3) :382 – 439, 1976.

[156] David A. Rosenthal et R. Bajcsy. Visual and conceptual hierarchy : A paradigm for studies of automated generation of recognition strategies. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-6(3) :319–325, May 1984.

[157] Y. Rui, T.S. Huang, et S.F. Chang. Image retrieval : Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation*, 10(1) :39–62, 1999.

[158] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, et William T. Freeman. Labelme : A database and web-based tool for image annotation. *Int. J. Comput. Vision*, 77(1-3) :157–173, 2008.

[159] S. Santini et R. Jain. Beyond query by example. Dans *WMSP'98 : IEEE Second Workshop on Multimedia Signal Processing, 1998*, pages 3–8, Dec 1998.

[160] Simone Santini. Ontology : Use and abuse. Dans *AMR'07 Adaptive Multimedial Retrieval : Retrieval, User, and Semantics : 5th International Workshop, AMR 2007, Paris, France, July 5-6, 2007 Revised Selected Papers*, pages 17–31, Berlin, Heidelberg, 2008. Springer-Verlag.

[161] Simone Santini et Alexandra Dumitrescu. Context and activity games as a non-ontological model of semantics. Dans *SAMT2008*, 2008.

[162] Ansgar Scherp et Ramesh Jain. Towards an ecosystem for semantics. Dans *MS '07 : Workshop on multimedia information retrieval on The many faces of multimedia semantics*, pages 3–12, New York, NY, USA, 2007. ACM.

[163] Henry Schneiderman et Takeo Kanade. A statistical method for 3d object detection applied to faces and cars. Dans *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, volume 1, page 1746, Los Alamitos, CA, USA, 2000. IEEE Computer Society.

[164] Bernhard Schölkopf, John C. Platt, John C. Shawe-Taylor, Alex J. Smola, et Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Comput.*, 13(7) :1443–1471, 2001.

[165] Sara Shatford. Analyzing the subject of a picture : A theoretical approach. *Cataloging & Classification Quarterly*, 6(3) :39–62, Mar 1986.

[166] Sara Shatford Layne. Some issues in the indexing of images. *J. Am. Soc. Inf. Sci.*, 45(8) :583–588, 1994.

[167] J. Sivic, B.C. Russell, A. Zisserman, W.T. Freeman, et A.A. Efros. Unsupervised discovery of visual object class hierarchies. Dans *CVPR08*, pages 1–8, 2008.

[168] J. Sivic et A. Zisserman. Video Google : A text retrieval approach to object matching in videos. Dans *ICCV'03 : Proceedings of the International Conference on Computer Vision*, volume 2, pages 1470–1477, Octobre 2003.

[169] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, et R. Jain. Content-based image retrieval at the end of the early years. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(12) :1349–1380, Dec 2000.

[170] P. Sollich. Probabilistic methods for support vector machines. Dans *NIPS*, volume 12, pages 349–355. MIT Press, 2000.

[171] Von-Wun Soo, Chen-Yu Lee, Chung-Cheng Li, Shu Lei Chen, et Ching-chih Chen. Automated semantic annotation and retrieval based on sharable ontology and case-based learning techniques. Dans *JCDL '03 : Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries*, pages 61–72, Washington, DC, USA, 2003. IEEE Computer Society.

[172] Munirathnam Srikanth, Joshua Varner, Mitchell Bowden, et Dan Moldovan. Exploiting ontologies for automatic image annotation. Dans *SIGIR '05*, pages 552–558, New York, NY, USA, 2005. ACM.

[173] G. Stamou, J. van Ossenbruggen, J.Z. Pan, G. Schreiber, et J.R. Smith. Multimedia annotations on the semantic web. *Multimedia, IEEE*, 13(1) :86–90, Jan 2006.

[174] Erik B. Sudderth, Antonio Torralba, William T. Freeman, et Alan S. Willsky. Learning hierarchical models of scenes, objects, and parts. Dans *ICCV'05*, volume II, pages 1331–1338, 2005.

[175] Ben Taskar, Carlos Guestrin, et Daphne Koller. Max-margin markov networks. Dans Sebastian Thrun, Lawrence Saul, et Bernhard Schölkopf, éditeurs, *NIPS'04*. MIT Press, Cambridge, MA, 2004.

[176] Alexander Thomas, Vittorio Ferrari, Bastian Leibe, Tinne Tuytelaars, et Luc Van Gool. Shape-from-recognition : Recognition enables meta-data transfer. *Computer Vision and Image Understanding*, 2009.

[177] S. Todorovic et N. Ahuja. Learning subcategory relevances to category recognition. Dans *CVPR'08 : Proc. IEEE Comp. Soc. Conf. Computer Vision and Pattern Recognition*, 2008.

[178] A. Torralba, R. Fergus, et W.T. Freeman. Tiny images. Rapport technique, Computer Science and Artificial Intelligence Laboratory Technical Report, Apr 2007.

[179] A. Torralba, R. Fergus, et Y. Weiss. Small codes and large image databases for recognition. Dans *CVPR'08 : IEEE Conference on Computer Vision and Pattern Recognition, 2008.*, pages 1–8, June 2008.

[180] Antonio Torralba, Kevin P. Murphy, et William T. Freeman. Sharing visual features for multiclass and multiview object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5) :854–869, 2007.

[181] Anne-Marie Tousch, Stéphane Herbin, et Jean-Yves Audibert. Semantic lattices for multiple annotation of images. Dans Michael S. Lew, Alberto Del Bimbo, et Erwin M. Bakker, éditeurs, *Multimedia Information Retrieval*, pages 342–349. ACM, 2008.

[182] Martin J. Tovée. *An introduction to the visual system*. Cambridge University Press, 1996.

[183] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, et Yasemin Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6 :1453–1484, 2005.

[184] M. Turk et A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1) :71–86, 1991.

[185] B. Tversky et K. Hemenway. Objects, parts, and categories. *Journal of Experimental Psychology*, 113(2) :169–193, Juin 1984.

[186] S. Ullman, M. Vidal-Naquet, et E. Sali. Visual features of intermediate complexity and their use in classification. *nature neuroscience*, 5(7) :682–687, 2002.

[187] N. Vasconcelos. Image indexing with mixture hierarchies. Dans *CVPR01*, pages I :3–10, 2001.

[188] Michel Vidal-Naquet et Shimon Ullman. Object recognition with informative features and linear classification. Dans *ICCV '03*, page 281, Washington, DC, USA, 2003. IEEE Computer Society.

[189] P. Viola et M.J. Jones. Rapid object detection using a boosted cascade of simple features. Dans *CVPR01*, pages I :511–518, 2001.

[190] J. Vogel et B. Schiele. Semantic modeling of natural scenes for content-based image retrieval. *IJCV*, 72(2) :133–157, April 2007.

[191] Changhu Wang, Lei Zhang, et Hong-Jiang Zhang. Learning to reduce the semantic gap in web image retrieval and annotation. Dans *SIGIR '08 : Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 355–362, New York, NY, USA, 2008. ACM.

[192] Gang Wang, Ye Zhang, et Li Fei-Fei. Using dependent regions for object categorization in a generative framework. Dans *CVPR '06*, pages 1597–1604, Washington, DC, USA, 2006. IEEE Computer Society.

[193] X.J. Wang, W.Y. Ma, et X. Li. Data-driven approach for bridging the cognitive gap in image retrieval. Dans *ICME'04 IEEE International Conference on Multimedia and Expo, 2004.*, volume 3, 2004.

[194] WordNet. Documentation, 2009. http://wordnet.princeton.edu/wordnet/man/wnstats.7WN.html, consultée le 24/09/09.

[195] Changbo Yang, Ming Dong, et Farshad Fotouhi. Learning the semantics in image retrieval - a natural language processing approach. Dans *CVPRW '04*, page 137, Washington, DC, USA, 2004. IEEE Computer Society.

[196] Changbo Yang, Ming Dong, et Farshad Fotouhi. Semantic feedback for interactive image retrieval. Dans *MULTIMEDIA '05 : Proceedings of the 13th annual ACM international conference on Multimedia*, pages 415–418, New York, NY, USA, 2005. ACM.

[197] Ka-Ping Yee, Kirsten Swearingen, Kevin Li, et Marti Hearst. Faceted metadata for image search and browsing. Dans *CHI '03 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 401–408, New York, NY, USA, 2003. ACM.

[198] B. Zadrozny. Reducing multiclass to binary by coupling probability estimates. Dans *NIPS'01*, 2001.

[199] Bianca Zadrozny et Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. Dans *KDD '02 : Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699, New York, NY, USA, 2002. ACM.

[200] Hao Zhang, Alexander C. Berg, Michael Maire, et Jitendra Malik. Svm-knn : Discriminative nearest neighbor classification for visual category recognition. Dans *CVPR '06*, pages 2126–2136, Washington, DC, USA, 2006. IEEE Computer Society.

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

[202] Qiang Zhu, Mei-Chen Yeh, Kwang-Ting Cheng, et Shai Avidan. Fast human detection using a cascade of histograms of oriented gradients. Dans *CVPR '06 : Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1491–1498, Washington, DC, USA, 2006. IEEE Computer Society.

[203] A. Zweig et D. Weinshall. Exploiting object hierarchy : Combining models from different category levels. Dans *ICCV2007*, pages 1–8. IEEE, Oct. 2007.