Apprentissage interactif pour la recherche par le contenu dans les bases multimédias

Philippe-Henri Gosselin

UNIVERSITÉ DE
CERGY - PONTOISE

Unité de Formation et de Recherche Sciences

École doctorale Sciences et Ingénierie

HABILITATION À DIRIGER DES RECHERCHES

Discipline :

Sciences et Technologies de l'Information et de la Communication

présentée et soutenue par :

Philippe-Henri GOSSELIN

Apprentissage interactif pour la recherche par le contenu dans les bases multimédias

---

soutenue à l'Université de Cergy-Pontoise

le 10 novembre 2011 devant le jury :

Président	
Pr Mohamed DAOUDI	Institut Telecom Lille 1
Examinateurs	
Pr Pierre-François MARTEAU	Université de Bretagne Sud
Pr Sylvie PHILIPP-FOLIGUET	École Nationale de l'Électronique et de ses Applications
Rapporteurs	
Pr Michel CRUCIANU	Conservatoire National des Arts et Métiers de Paris
DR Patrick GROS	Centre INRIA de Rennes - Bretagne Atlantique
Pr Sébastien LEFÈVRE	Université de Bretagne Sud

Table des matières

Introduction

Contexte

Au début des années 1990, et d'une manière encore plus prononcée à partir des années 2000, les bases d'images sont devenues de plus en plus grandes et difficiles à exploiter, faute d'outils adaptés. De nombreux chercheurs se sont alors intéressés aux différentes problématiques autour de ces bases, comme le stockage ou l'accès aux images.

En 2002, dans le cadre de ma thèse, j'ai pu m'intéresser à l'une de ces problématiques, à savoir la recherche des images par leur contenu (*Content-Based Image Retrieval*). Ce type de recherche se distingue des autres de par le fait que l'on n'utilise pas d'informations complémentaires aux images (comme des mots clefs), mais on travaille directement sur le contenu même du document. Nous avons travaillé ainsi sur les propriétés visuelles des images, telles que les couleurs et les textures. Ces propriétés sont résumées au sein d'un histogramme, formé à partir d'un dictionnaire visuel calculé sur une base de référence. Une fois munie d'une métrique, cette représentation des images permet de calculer la similarité d'une image requête à un ensemble d'images. Cette première approche permet de répondre à des problèmes de recherche de copie (exactement la même image, mais déformée) et de recherche d'instance (exactement le même objet, mais vu sous un autre angle). Cependant, pour d'autres problèmes de recherche, il est nécessaire d'aller plus loin.

Dans ce contexte, je me suis concentré lors de ma thèse sur la recherche de catégories, où un utilisateur souhaite retrouver un ensemble d'images aux propriétés sémantiques communes. Par exemple, l'ensemble des voitures, des paysages, etc. Dans le but de répondre à ce problème, nous avons opté pour une approche par apprentissage statistique. L'idée est de produire un classifieur qui, sur la base d'un ensemble d'apprentissage, détermine dans quelle mesure une image peut intéresser un utilisateur. En un mot, ces classifieurs doivent faire le lien entre les propriétés visuelles et les propriétés sémantiques. Il existe différentes manière d'entraîner ces classifieurs, par exemple dans le cas de l'apprentissage par lot (*Batch Learning*) on utilise de très larges ensembles d'apprentissage. Dans notre cas, nous nous sommes intéressés à un scénario où un utilisateur interagit avec le système de recherche. La procédure se veut simple et intuitive, avec une interaction dite faiblement supervisée, où l'utilisateur indique si une image est intéressante ou non. Dans ce contexte, l'utilisateur n'a pas besoin de préciser les parties intéressantes de l'image, c'est au système de le deviner.

La première partie du travail a porté sur le classifieur et la métrique associée, où nous avons comparé différentes approches, tels que les mélanges de gaussiennes et les k-Plus Proches Voisins. Parmi celles ci, il s'est avéré qu'une approche par machine à noyau était la plus pertinente, et ce sans compter sur son caractère générique. La deuxième partie du travail a porté sur la gestion de l'interaction avec l'utilisateur. Pour ce faire, nous avons proposé des techniques

basées sur l'apprentissage actif qui déterminent les images les plus intéressantes à faire annoter par l'utilisateur. Enfin, la troisième partie du travail a porté sur l'apprentissage à long terme, où nous avons proposé des méthodes pour optimiser le système en fonction de son utilisation.

Ces travaux de thèse ont permis de dresser un cadre de travail basé sur plusieurs outils, comme les dictionnaires visuels ou les machines à noyaux. C'est sur cette base que j'ai pu proposer de nouvelles pistes pour améliorer la recherche de catégories, mais aussi pour traiter d'autres médias comme les objets en trois dimensions. Les différents projets que j'ai pu mener autour de ces thèmes ont donné lieu à un certain nombre de contributions, dont les plus abouties sont présentées dans ce mémoire.

Dans ce contexte général, nous présentons au chapitre suivant une vue plus détaillée des outils que nous avons utilisés pour la conception d'un système de recherche interactive. Puis, nous présentons lors d'une première partie des techniques de recherche qui utilisent des noyaux sur ensembles, où l'objectif est d'obtenir une machine à noyau basée sur le principe de la mise en correspondance. La deuxième partie du mémoire est consacré à des techniques de représentation dynamique, où l'on apprend en ligne la représentation des documents.

Architecture d'un système de recherche interactive

La conception d'un système de recherche interactive nécessite de nombreux composants qu'il faut coordonner pour obtenir un ensemble pertinent. Nous présentons dans la figure 1 un exemple d'architecture utilisant des dictionnaires visuels et des machines à noyaux. Cet exemple d'architecture est une base sur laquelle nous nous appuierons pour proposer différentes solutions, en modifiant certains composants. Cette architecture est divisée en deux grandes parties : l'une hors ligne, consacrée au pré-traitement de la base, et l'autre en ligne, consacrée au traitement des requêtes utilisateurs.

Indexation

La phase hors ligne du système de recherche permet d'obtenir des représentations pour les documents, ainsi que les métriques associées. Notons que l'on parle aussi de signature des documents. L'ensemble des structures de données associées est aussi appelé index de la base, résultat du processus d'indexation de la base. Pour parvenir à ces représentations, on passe par plusieurs étapes, en commençant par l'extraction de descripteurs visuels.

Descripteurs visuels

Pour les images, un processus d'extraction courant est le passage d'un espace colorimétrique pertinent pour l'acquisition (par exemple, RVB) à un espace colorimétrique pertinent sur le plan psycho-visuel (par exemple, $L^\star a^\star b^\star$). On peut aussi procéder par filtrage, en calculant les gradients ou en appliquant des transformées. Parmi les différentes techniques de filtrage, on peut noter celles par banc de filtres de Gabor [55], les motifs binaires locaux (*Local Binary Pattern*) [62], les ondelettes de Haar [87], complexes [41] ou quaternioniques [5]. Pour les objets 3D représentés sous la forme de mailles, on extrait généralement des propriétés telles que les directions des normales ou les courbures.

Un document peut être partitionné en régions en utilisant un processus de segmentation. La manière la plus simple est d'utiliser une segmentation statique, par exemple en découpant une image en blocs réguliers. Il est aussi possible de procéder de manière adaptative, en s'appuyant sur les propriétés visuelles du document, comme les gradients des images ou l'orientation des normales des surfaces d'un objet 3D. Pour ce faire, il existe les méthodes basées sur les contours

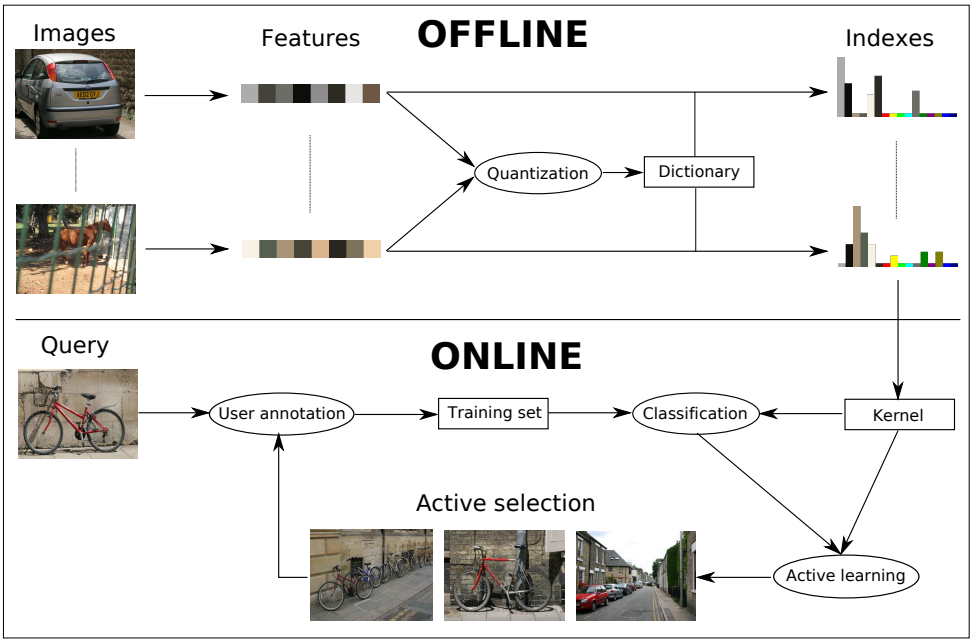


FIGURE 1 – Exemple d'architecture de système de recherche interactive.

actifs [39] et sur la montée des eaux (*watershed*) pour les images [86] et les objets 3D [69]. Les régions extraites sont ensuite décrites en utilisant des processus similaires à ceux utilisés pour décrire un document entier.

Une caractérisation locale qui connaît beaucoup de succès dans le domaine de la vision par ordinateur est celle par points d'intérêt. L'idée ici est de reproduire le système de vision rétinien, qui se focalise sur les parties du document les plus riches en hautes fréquences. La méthode la plus populaire pour les images est celle des SIFT [49], qui est basée sur des différences de gaussiennes. Puis, on trouve de nombreuses propositions dans ce domaine, comme les détecteurs Harris-Affine qui offrent une invariance aux transformations affines [59], les MSER qui se focalisent sur les régions stables de l'image [57], ou encore les SURF qui permettent d'extraire des points d'intérêt très rapidement [2]. Notons qu'il existe aussi des méthodes pour les objets 3D [58] et les vidéos [44]. Une fois les points d'intérêt détectés, ils sont décrits en fonction de la région qui entoure le point, généralement avec un soucis d'invariance aux transformations visuelles [60].

Dictionnaires visuels

Travailler sur l'ensemble des descripteurs extraits est généralement difficile compte tenu de leur nombre. Une solution courante est d'utiliser un dictionnaire visuel pour produire une représentation compacte des documents, comme les histogrammes, qui contiennent le nombre d'occurrences de chaque mot visuel dans le document.

Pour produire le dictionnaire, on peut procéder par quantification statique des descripteurs, par exemple en découpant de manière régulière l'espace des couleurs [79]. Du côté des objets

en trois dimensions, on peut par exemple projeter l'orientation des faces sur une gaussienne qui enveloppe l'objet [32, 36]. On peut aussi procéder de manière adaptative, en déterminant automatiquement les prototypes de descripteurs visuels. L'idée est d'analyser les descripteurs visuels présents dans une base représentative pour en extraire des prototypes, les "mots visuels". Pour ce faire, on utilise généralement un algorithme de K-Means [15] ou ELBG [65], mais aussi des algorithmes par reconstruction linéaire locale [91]. Diverses techniques ont été proposées pour ce faire sur le principe des "Bags of Words", par exemple à l'aide de pyramides [27], avec les noyaux de Fisher [67], par forêts aléatoires [61], ou encore par codage linéaire local [91]. Certains chercheurs proposent aussi de concevoir les représentations des documents comme l'approximation explicite d'un processus de mise en correspondance, par exemple avec une encapsulation de Hamming [33].

Index rapides

La pertinence d'une représentation est importante pour pouvoir retrouver les documents d'intérêt, mais pour traiter de très grandes bases, il faut aussi s'intéresser aux coûts matériels d'une représentation. Un des thèmes les plus étudiés dans ce cadre est celui de la recherche rapide des k plus proches voisins dans un espace vectoriel. Il existe de nombreuses méthodes qui partitionnent l'espace pour accélérer la recherche, souvent au prix d'une approximation [4, 70]. Nous pouvons noter les méthodes par hachage local ("Locality Sensitive Hashing"), qui permettent d'accélérer considérablement les temps de recherche [11]. Le cas de très grandes bases pose aussi des problèmes en terme de stockage, et c'est la raison pour laquelle des solutions sont proposées pour prendre en compte les temps d'accès sur disques [47], ou tout simplement pour réduire considérablement la taille des représentations pour qu'elles tiennent en mémoire [34].

Métriques

Une fois que chaque document de la base est représenté, il est nécessaire de disposer d'une notion de similarité (ou de distance) pour pouvoir effectuer les recherches.

Distances

Dans le cas où les représentations des documents sont des vecteurs, on se tourne souvent vers les distances comme la distance Euclidienne et plus généralement celle de Minkowski, ou encore les distances quadratiques généralisées. Dans le cas où les représentations vectorielles s'apparentent à des histogrammes ou des distributions, on utilise la distance du χ^2, ou pour beaucoup de méthodes que nous présentons dans ce mémoire, celle du χ^1 :

$$d_{\chi^1}(\mathbf{x}, \mathbf{y}) = \sum_{r=1}^{p} \frac{|x_r - y_r|}{x_r + y_r} \tag{1}$$

avec $\mathbf{x} \in \mathbb{R}^p$ et $\mathbf{y} \in \mathbb{R}^p$.

Fonctions noyaux

De nombreux types de représentation des documents ne se prêtent pas naturellement aux méthodes d'apprentissage les plus éprouvées. À défaut d'adapter ou de concevoir des méthodes d'apprentissage dédiées à un type de représentation, une stratégie est de ramener tout type de représentation à un type bien connu et maîtrisé. Plus précisément, on introduit une opération supplémentaire de transformation d'espace, qui injecte les représentations des documents dans un nouvel espace vectoriel muni d'un produit scalaire. Lorsque le produit scalaire est défini, on parle d'*espace Hilbertien*, et lorsqu'il est indéfini, on parle d'*espace de Krein*.

Il existe plusieurs approches pour effectuer une telle injection, dont celle par fonctions noyaux, qui présente de nombreux avantages [75, 76]. Parmi ceux-ci, on peut noter la possibilité de travailler dans des espaces de très grandes dimensions sans avoir à les expliciter. Plus formellement, cette approche s'intéresse à la fonction noyau $k(x, y)$ qui correspond au produit scalaire $\langle ., . \rangle$ dans le nouvel espace :

$$k(x, y) = \langle \phi(x), \phi(y) \rangle \tag{2}$$

avec $\phi(x)$ l'injection qui, à une représentation x, fait correspondre un vecteur dans le nouvel espace. Que le produit scalaire soit défini ou non, le nouvel espace est appelé *espace induit par la fonction noyau* k.

Il existe de nombreux algorithmes compatibles avec les noyaux définis, notamment grâce au "truc du noyau", qui permet d'utiliser n'importe quel algorithme dès lors que l'on peut l'écrire uniquement à l'aide de produits scalaires. Pour le cas des noyaux indéfinis, comme certains que nous présentons pour traiter les sacs et graphes, il existe des méthodes pour les utiliser, qui en un mot ramènent le cas indéfini à un cas défini [63, 29, 28, 66].

Classification active

Les processus hors ligne terminés, chaque document de la base est représenté et une métrique est choisie pour comparer les documents. L'utilisateur présente alors un document requête, et le système affiche les plus proches voisins. Si l'utilisateur est satisfait, le processus s'arrête là. Dans le cas contraire, il peut annoter des documents dans la base, et le système utilise alors ces annotations pour produire un meilleur résultat.

Classification

Une approche courante pour exploiter les annotations utilisateur est de fabriquer une fonction de classification en deux classes : la classe des documents recherchés, et la classe des autres documents. Notons qu'il existe d'autres approches, comme la modification de requête [73], la pondération d'attributs [1], l'optimisation d'un modèle probabiliste autour de la requête [13, 18], ou encore des modèles bayésiens [8].

Pour former la fonction de classification, dès lors que l'on dispose directement ou indirectement d'une représentation vectorielle des documents, on trouve de nombreuses méthodes qui sont utilisées en multimédia, comme le critère de Bayes [84], les k-Plus Proches Voisins,

les K-Means, les mélanges de gaussiennes [25], les discriminants de Fisher, les Séparateurs à Vaste Marge [6], etc. Chaque méthode a ses avantages et inconvénients, et le choix dépend des contraintes de l'application. Dans notre cas de recherche interactive, nous travaillons le plus souvent avec les SVMs, car la méthode est compatible avec les fonctions noyaux, et l'algorithme d'entraînement est convexe et très rapide.

Apprentissage actif

L'apprentissage actif est l'étape de sélection des documents non annotés les plus intéressants à faire annoter par l'utilisateur. En d'autres termes, un système actif évalue chaque document non annoté selon l'impact qu'il pourrait avoir s'il était annoté. Pour ce faire, il existe deux familles de méthodes : une première que l'on peut qualifier de "pessimiste" et une deuxième d'"optimiste".

La première famille active, dite "pessimiste", sélectionne en assurant qu'un minimum de gain soit obtenu, quelle que soit l'annotation de l'utilisateur (positive ou négative). Pour atteindre ce but, on raisonne généralement dans l'espace des versions, qui représente l'ensemble des annotations que l'utilisateur peut formuler. L'idée est alors de réduire cet espace le plus rapidement possible, par exemple en modélisant l'espace des versions, puis en choisissant le document qui, une fois annoté, coupera au mieux en deux le modèle de l'espace des versions [83]. En fin de compte, ces techniques convergent vers l'idée que le choix le plus pertinent est celui des documents dont la classification est la plus incertaine. Le plus simple pour ce faire est de sélectionner les documents les plus proches de la frontière du classifieur.

La deuxième famille active, dite "optimiste", sélectionne les documents qui, s'ils sont annotés d'une certaine manière, offriront un gain important. Une manière de suivre ce principe est d'ajouter artificiellement une annotation, entraîner le système, puis évaluer le résultat [72]. Ce processus est répété pour tous les documents et toutes les annotations possibles, et on choisit le document qui offre le meilleur gain. Si l'utilisateur choisit l'annotation attendue, on obtient une belle progression, mais dans le cas contraire, le résultat peut ne pas évoluer du tout. Notons que les approches optimistes sont, d'une manière générale, beaucoup plus coûteuses en calculs que les approches pessimistes.

Aucune des deux familles n'est meilleure que l'autre, le choix le plus judicieux étant de faire un mélange [24]. On commence par présélectionner un certain nombre de documents avec une méthode pessimiste, puis on applique un critère optimiste au sein de la pré-sélection. Ainsi, on assure un minimum de gain via la pré-sélection, tout en s'autorisant à trouver des documents qui peuvent améliorer grandement le résultat.

Dans le cas où l'on souhaite sélectionner plusieurs documents non annotés à chaque itération du processus de bouclage, une approche courante pour améliorer les résultats est de s'assurer que l'on sélectionne des documents différents. Dans le cas contraire, on pourrait sélectionner plusieurs copies (ou presque) d'un document, et n'obtenir que le gain équivalent à la sélection d'un seul de ces documents. Pour s'assurer de la diversité des documents sélectionner, un critère pertinent est celui de la diversité des angles [17].

Principales contributions

Noyaux sur sacs et graphes

Dans le but de rechercher des objets dans les bases d'images, une approche pertinente est de travailler sur les mises en correspondance entre descripteurs visuels. Dans ce cadre, les systèmes par votes permettent d'obtenir de très bons résultats pour la recherche par similarité, où l'on recherche les images les plus proches d'une requête. Cependant, ces systèmes ne sont pas capables d'exploiter des exemples d'apprentissage, et par là même de résoudre les problèmes de recherche d'objets les plus difficiles.

Afin de concilier apprentissage et mise en correspondance, j'ai travaillé sur les fonctions noyaux sur ensembles, principalement pendant mon post-doctorat. Nous nous sommes tout d'abord intéressés aux noyaux sur sacs de descripteurs, présentés au ***Chapitre 1***. Ce cadre permet de déplacer le problème de mise en correspondance entre les descripteurs visuels dans un espace induit par une fonction noyau, où chaque point représente une combinaison de descripteurs. Ces fonctions permettent ainsi de reproduire les systèmes par votes, mais en respectant les propriétés mathématiques nécessaires à l'utilisation des classifieurs comme les SVMs. Dans ce contexte, nous avons proposé des solutions innovantes qui permettent de corriger certains défauts de méthodes proposées dans la littérature.

Les fonctions noyaux sur sacs ne prennent pas en compte les positions des descripteurs visuels dans l'image. Dans le but d'intégrer cette information, et dans le cadre des thèses de J.E. Haugeart et J. Lebrun, nous avons étudié l'approche par noyaux sur graphes, présentée au ***Chapitre 2***. Dans ce cas, le graphe représente les différentes primitives visuelles (régions, points d'intérêt) réparties dans l'image. Il existe différentes approches pour ce faire, et parmi celles-ci, nous avons choisi de travailler sur les noyaux sur sacs de chemins, où l'on considère un ensemble de chemins extraits de chaque graphe. Outre les problèmes d'énumération des chemins et de métriques pour comparer les chemins, le principe est très similaire à celui des noyaux sur sacs. En effet, on injecte les chemins dans un espace induit, puis on effectue la mise en correspondance dans cet espace. Dans ce contexte, nous avons proposé de nouvelles fonctions qui, en plus d'être performantes, ont un coût calculatoire inférieur aux méthodes de référence. Les résultats obtenus ont été validés sur des bases d'images, mais aussi sur des objets en trois dimensions dans le cadre du projet ANR EROS 3D, ainsi que pour la détection des fenêtres dans les façades de Paris pour le projet ANR iTowns.

Noyaux dynamiques

Une approche qui connait beaucoup de succès en recherche de catégories d'images est celle basée sur des dictionnaires visuels. Pour ce faire, on construit un dictionnaire de mots visuels, généralement avec un algorithme K-Means, puis on décrit chaque image à l'aide d'un histogramme. De nombreuses méthodes ont été proposées pour construire ces dictionnaires, et dans presque tous les cas, il est nécessaire de passer par une phase d'étalonnage pour régler les paramètres du dictionnaire en fonction des catégories susceptibles d'être recherchées. Or, dans un contexte de recherche interactive, on ne dispose que de très peu d'a priori sur la nature des recherches qui vont être menées par les utilisateurs. De plus, les centres d'intérêt des utilisateurs peuvent évoluer au cours du temps, et par conséquent les paramètres optimaux aussi.

Dans le but de réduire cette dépendance à des paramètres globaux, j'ai étudié une approche qui consiste à effectuer le calcul du dictionnaire non plus au préalable, mais pendant la session de recherche. Pour régler le dictionnaire, je me base sur les annotations fournies par l'utilisateur. Ainsi, chaque session de recherche et même chaque itération de bouclage dispose d'un dictionnaire unique et adapté. Une approche naïve pour ce faire est de refaire l'ensemble des calculs habituellement effectués hors ligne. Une telle approche n'est cependant pas pertinente étant donnée la grande quantité de calculs requis. Dans ce contexte, je présente au ***Chapitre 3*** une méthode pour apprendre le dictionnaire tout en respectant les contraintes de temps de calcul. Plus précisément, je propose d'apprendre une combinaison de fonctions noyaux de base, où chaque noyau de base correspond à un mot visuel. Le résultat de cette combinaison est alors équivalent à un dictionnaire visuel. Le processus est itératif, et permet ainsi d'apprendre à faible coût, tout en mutualisant de nombreuses ressources d'une itération de bouclage à l'autre, voire d'une session à l'autre, grâce à des systèmes de cache.

Boosting interactif

Toutes les techniques précédentes travaillent dans le cadre des fonctions noyaux. Ce cadre dispose de bien des avantages, mais on peut toutefois lui faire un reproche. En effet, tout repose sur des fonctions noyaux pertinentes pour le problème abordé. Or, pour construire ces fonctions, il faut impérativement respecter un certain nombre de propriétés mathématiques. Ainsi, pour traiter de nouveaux types de structures pour représenter les documents, un important travail de conception doit être réalisé. C'est la raison pour laquelle nous nous sommes tournés vers le boosting, qui demande moins d'efforts de conception. Ce travail est mené dans le cadre de la thèse de A. Lechervy, et les résultats seront utilisés lors du stage du R. Negrel.

Le cadre du boosting est un processus itératif de sélection de classifieurs faibles dans le but de former un classifieur fort pour classer la base. Le premier intérêt de ce cadre est de n'imposer que de faibles hypothèses sur les classifieurs faibles, ce qui rend leur conception aisée. Un autre intérêt du boosting est qu'il permet aussi d'apprendre la représentation des documents en même temps que le classifieur fort, étant donné que chaque classifieur faible est une manière différente d'observer les données. Notons que ce processus est, sur le fond, très proche des noyaux dynamiques que nous avons proposés précédemment, et nous retrouvons

beaucoup d'avantages communs aux deux approches.

Le boosting a cependant été introduit dans la littérature pour faire de l'apprentissage par lots, où l'on dispose d'un grand nombre d'exemples d'apprentissage. Autrement dit, tout le contraire de la recherche interactive. Cependant, il existe des moyens d'utiliser ce cadre avec peu d'annotations. En effet, nous présentons au ***Chapitre 4*** nos propositions pour ce faire. Parmi les idées maitresses qui permettent de gérer les contraintes de la recherche interactive, nous montrons celle qui consiste à travailler avec des ensembles de classifieurs faibles dynamiques, adaptés et optimisés au fur et à mesure des itérations de bouclage. Un autre avantage du boosting est qu'il permet de produire des classifieurs forts moins coûteux en calculs, étant donné qu'il sélectionne uniquement les classifieurs faibles pertinents pour la catégorie recherchée. Ainsi, on n'effectue pas de calcul sur des descriptions ou représentations inutiles voire bruitées, contrairement aux méthodes à représentation statique où l'intégralité de l'information doit être utilisée.

Autres travaux

Ce mémoire ne présente en détails que mes contributions après la thèse qui ont atteint un certain niveau de maturité. Je travaille néanmoins sur d'autres thèmes présentés rapidement dans cette section.

Dans le cadre de la thèse en cotutelle France Brésil de Jefersson Alex dos Santos, je travaille sur l'apprentissage de fonctions capables de segmenter des images. On commence par fournir au système une ou plusieurs images exemples, où chaque pixel est annoté comme appartenant à une région d'intérêt ou non. Dans le cadre de la thèse, les régions d'intérêt sont par exemple les champs de café dans les images satellite. Sur la base de ces exemples, un algorithme basé sur le principe du Boosting conçoit une fonction de décision capable de déterminer si le pixel de n'importe quelle image fait partie d'une région d'intérêt ou non. Bien que la décision finale soit exprimée au niveau des pixels, tout le processus de décision est basé sur un ensemble de descriptions des régions de l'image à différentes échelles.

Je travaille aussi sur les descripteurs de mouvements dans les vidéos. L'objectif est de concevoir des techniques capables de produire des descripteurs de mouvements utilisables dans un contexte de recherche dans les bases de vidéos, où l'on s'intéresse aux actions humaines (courir, manger, ...) ou aux évènements (accident, mouvement de foule, ...). Le travail a débuté dans le cadre du stage Master de Virginia Fernandez, sur la base des travaux de Martin Druon [14] qui modélise le flot optique sous la forme d'une combinaison linéaire de polynômes. Ces travaux préliminaires ont permis de créer des descripteurs de mouvement global simples et performants. Ces travaux ont ensuite été repris dans le cadre du stage Master de Romain Négrel, avec qui je conçois une version capable de contenir des descriptions de mouvements locaux. Le but à terme de ces travaux est de les combiner avec les méthodes d'apprentissage que développe Alexis Lechervy dans sa thèse.

Première partie

Noyaux sur Ensembles

Chapitre 1

Noyaux sur sacs

De nombreuses méthodes ont été proposées ces dernières années pour permettre la recherche d'objets dans les bases d'images. Ces techniques permettent de gérer les différents problèmes liés à ce type de recherche, comme la robustesse aux changements de pose et aux occultations. De premières méthodes reposent sur un système de vote, et comptent en quelque sorte le nombre de mises en correspondance que l'on peut faire entre les descripteurs visuels d'une requête et ceux des documents d'une base. Puis, pour permettre une recherche plus large et/ou plus robuste aux variations de l'objet, on peut suivre une approche basée sur la création d'un classifieur fonction d'un ensemble d'apprentissage. Pour ce faire, on peut distinguer deux grandes familles.

Une première famille est basée sur une approche par dictionnaire visuel, dont les premières méthodes sont les sacs de mots visuels ("Bags of Words"). Dans tous les cas, on commence par construire un dictionnaire de mots visuels, chaque mot visuel étant généralement apparenté à un descripteur visuel. Puis, chaque image est représentée sous la forme d'un vecteur, dont chaque composante représente une probabilité de présence d'un mot visuel. Avec une telle représentation, il est par conséquent très facile d'utiliser des classifieurs. Diverses méthodes ont été proposées dans cet esprit, par exemple par K-Means[27] ou codage[53]. Chaque méthode se différencie d'une autre dans la manière de calculer le dictionnaire et les vecteurs associés aux images, mais dans tous les cas on observe une forte dépendance au dictionnaire. Cela peut expliquer le succès des méthodes de combinaison de noyaux ("Multiple Kernel Learning[43]"), compte tenu de leur capacité à déterminer et combiner les meilleurs dictionnaires visuels. Cependant, hormis pour certaines propositions très récentes, les techniques de combinaison reposent sur un large ensemble d'apprentissage pour être pleinement fonctionnelles. Ainsi, il va s'en dire que, dans un contexte de recherche interactive où les exemples sont rares, ces approches par dictionnaire sont difficiles à utiliser sans un travail d'adaptation.

Une deuxième famille de méthodes capables de concilier recherche d'objet et classification est celle basée sur les fonctions noyaux[90]. Dans ce cas, on revient aux sources de la recherche d'objet, à savoir la mise en correspondance de descripteurs visuels. Dans ce contexte, il n'est plus question de dictionnaire visuel : on conserve l'ensemble des descripteurs visuels extraits, ou au pire on réduit leur nombre en supprimant les redondances. Puis les images sont comparées en comptant en quelque sorte le nombre de descripteurs visuels communs. Notons que dans ce cas, les comparaisons sont toujours relatives aux images comparées, et non plus relatives à un

référentiel commun (le dictionnaire). Autrement dit, nous sommes dans une approche moins dépendante d'un processus lourd de validation croisée. De plus, cette approche se distingue des autres par les contraintes imposées à la métrique pour comparer les images. En effet, on souhaite assurer que la comparaison entre deux ensembles de descripteurs corresponde à un produit scalaire dans un espace vectoriel, autrement dit une fonction noyau. Une fois muni d'une telle fonction, toutes les méthodes à noyaux s'offrent à nous, de la PCA aux SVM.

1.1 Noyaux et mise en correspondance

Nous supposons que chaque document i de la base est représenté sous la forme d'un sac $B_i \in \mathcal{B}$ de descripteurs visuels $\mathbf{b}_{ri} \in \mathcal{F}$, ainsi que d'une fonction noyau $k(.,.)$ définie sur $\mathcal{F}^2$. Nous utilisons le plus souvent des descripteurs $\mathbf{b}_{ri}$ vectoriels ($\mathcal{F} = \mathbb{R}^p$), cependant d'un point de vue théorique n'importe quelle forme est possible dès lors que l'on dispose de la fonction $k(.,.)$. Notons que nous désignons par la suite le noyau $k(.,.)$ comme étant le *noyau mineur*.

Les fonctions noyaux que nous présentons dans ce chapitre s'appuient sur l'hypothèse que les descripteurs visuels $\mathbf{b}_{ri}$ et la fonction noyau mineure associée $k(.,.)$ permettent de mettre en correspondance les parties d'un même objet dans des images différentes. Par exemple, si on a plusieurs photographies d'un même objet pris sous un angle proche, on suppose que la fonction noyaux mineure renvoie des valeurs très élevées entre les descripteurs visuels pris aux mêmes endroits de l'objet. Cette hypothèse, que nous appelons par la suite "hypothèse de mise en correspondance" est essentielle pour le bon fonctionnent des méthodes qui suivent, et sans laquelle rien ne peux être appris.

1.1.1 Un noyau simple

Un bon moyen d'introduire la notion de noyau sur sacs est de présenter celui proposé par Shawe-Taylor[76] :

$$K(B_i, B_j) = \frac{1}{|B_i|}\frac{1}{|B_j|}\sum_r \sum_s k(\mathbf{b}_{ri}, \mathbf{b}_{sj}) \tag{1.1}$$

Notons que nous désignons par la suite la fonction noyau K comme étant le *noyau majeur*.

Littéralement, cette fonction calcule le produit scalaire moyen entre les éléments des deux sacs. A première vue, il n'y a pas de notion de mise en correspondance. Il faut cependant garder à l'esprit que le produit scalaire est calculé dans un espace induit par le noyau mineur k. En outre, les fonctions noyaux sont des *injections* – autrement dit l'espace induit a une dimension plus grande que l'espace initial, souvent bien plus grande. Un noyau courant est le noyau gaussien, qui induit un espace de dimension infinie. La motivation derrière cette injection est d'obtenir un espace où les données sont linéairement séparables [80].

Sachant cela, les valeurs des produits scalaires mineurs de la fonction de l'équation (1.1) sont généralement soit très faibles si les descripteurs sont un peu différents, soit très grands (ou proches de la valeur maximale) si les descripteurs sont identiques ou presque identiques. Par

conséquent, et sous réserve que la taille des sacs n'est pas trop grande, cette fonction noyau effectue une sorte de comptage des descripteurs communs aux deux sacs.

1.1.2 L'espace induit

Toujours dans le but de présenter les idées maîtresses derrière les noyaux sur sacs, développons la fonction de l'équation (1.1) pour mieux comprendre ce qui se passe dans l'espace induit. Notons tout d'abord $\phi : \mathcal{X} \rightarrow \mathbb{H}$ la fonction injection mineure associée au noyau mineur k, soit :

$$k(\mathbf{b}_{ri}, \mathbf{b}_{sj}) = \langle \phi(\mathbf{b}_{ri}), \phi(\mathbf{b}_{sj}) \rangle$$

Il s'ensuit que la fonction d'injection majeure $\Phi : \mathcal{B} \rightarrow \mathbb{H}$ associée au noyau majeur K est :

$$\Phi(B_i) = \frac{1}{|B_i|} \sum_r \phi(\mathbf{b}_{ri}) \tag{1.2}$$

étant donné que :

$$\begin{aligned} K(B_i, B_j) &= \Big\langle \frac{1}{|B_i|}\Phi(B_i), \frac{1}{|B_j|}\Phi(B_j) \Big\rangle \\ &= \frac{1}{|B_i|}\frac{1}{|B_j|} \langle \sum_r \phi(\mathbf{b}_{ri}), \sum_s \phi(\mathbf{b}_{sj}) \rangle \\ &= \frac{1}{|B_i|}\frac{1}{|B_j|} \sum_r \sum_s \langle \phi(\mathbf{b}_{ri}), \phi(\mathbf{b}_{sj}) \rangle \\ &= \frac{1}{|B_i|}\frac{1}{|B_j|} \sum_r \sum_s k(\mathbf{b}_{ri}, \mathbf{b}_{sj}) \end{aligned}$$

Cela signifie que la fonction K compare les barycentres des descripteurs injectés dans l'espace induit. En conséquence, on peut se demander comment des mises en correspondance peuvent avoir lieu entre deux ensembles d'éléments réduits à leurs moyennes respectives. Une fois encore, l'astuce de l'injection dans un espace de très grande dimension est l'explication.

En effet, si l'espace induit est de très grande dimension, voire infini, chaque descripteur "possède" sa propre dimension, tous les descripteurs un tant soit peu différents sont alors quasi orthogonaux entre eux. Si nous prenons un cas extrême avec un noyau mineur gaussien, chaque vecteur $\phi(\mathbf{b}_{ri})$ dans l'espace induit est principalement composé de 0, sauf sur une dimension où la valeur est 1 (la dimension que "possède" le descripteur). Ainsi, l'image $\Phi(B_i)$ du sac B_i dans l'espace induit est un vecteur principalement nul, sauf sur les dimensions correspondant aux descripteurs $\mathbf{b}_{ri}$ qu'il contient. Les images des sacs évoluent ainsi dans l'espace des combinaisons de descripteurs. Dans ce cas extrême, le produit scalaire entre les images des sacs est bel et bien un comptage des descripteurs identiques. Naturellement, dans la pratique un tel cas n'est pas désirable si l'on souhaite avoir un minimum de capacité à généraliser – capacité que l'on peut régler par ailleurs en jouant sur les paramètres du noyau mineur (largeur du noyau gaussien par exemple).

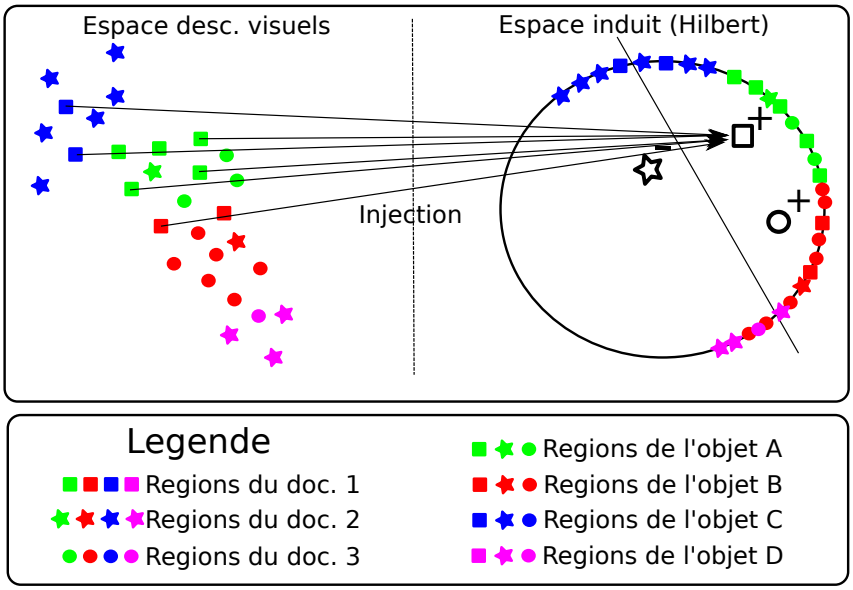


FIGURE 1.1 – Interprétation géométrique des noyaux sur sacs

1.1.3 Classification faiblement supervisée

Ces fonctions noyaux permettent de faire encore plus que reproduire des comptages de mises en correspondance. En effet, elles permettent de travailler dans un espace où l'on peut séparer linéairement les sacs contenant des objets d'intérêt ($\Leftrightarrow$ une certaine combinaison de descripteurs) des autres sacs, et ce uniquement sur la base d'annotations sur les sacs. Pour illustrer ces propos, nous proposons de nous appuyer sur la figure 1.1. Dans cette figure, nous avons représenté un cas avec 3 sacs et quatre objets (A,B,C et D). Les éléments des sacs sont des vecteur de dimension 2 (partie gauche de la figure) représentés sous la forme d'un symbole et d'une couleur. Tous les éléments de même symbole (carré, rond ou étoile) sont dans le même sac, et tous les éléments de même couleur (rouge, vert, bleu ou mauve) sont supposés appartenir au même objet. De plus, nous avons choisi un noyau mineur qui injecte les descripteurs dans une hypersphère (comme les noyaux gaussiens ou triangulaire).

Ainsi, l'image dans l'espace induit d'un descripteur est sur l'hypersphère, et l'image d'un sac dans l'hypersphère. L'intérieur de l'hypersphère est le lieu des combinaisons descripteurs. A présent, on suppose que les sacs "carré" et "rond" ont été ajoutés positivement, et le sac "étoile" négativement, puis calculé un hyperplan séparateur. On peut constater en observant la partie droite de la figure que l'on a aussi classifié les éléments des sacs, et plus particulièrement on peut observer que les descripteurs de l'objet vert sont au cœur de la classe positive. Autrement dit, sur la base d'annotations sur les sacs, nous sommes en mesure de produire des classifieurs linéaires capables de déterminer les composantes d'un objet d'intérêt.

Cette propriété est très intéressante, et tout particulièrement pour la recherche interactive. En effet, les utilisateurs préfèrent effectuer le moins possible de tâche d'annotation. Par conséquent, un système capable de retrouver les objets sans avoir à les détourer ($\Leftrightarrow$ annoter les éléments des sacs), mais simplement en annotant simplement les images ($\Leftrightarrow$ annoter les sacs)

est bien plus intéressant. Dans ce contexte, nous parlerons d'apprentissage *faiblement supervisé*, où les éléments d'intérêt non pas besoin d'être annotés, mais seulement leur contenant. En conséquence, les noyaux sur sacs combinés à des classifieurs par hyperplan font parti des méthodes d'apprentissage avec instances multiples ("Multiple Instance Learning").

Toutes les propriétés dépendent énormément de la manière dont va être construit l'espace induit. Pour ce faire, on peut travailler sur le noyau mineur, mais aussi sur la forme du noyau majeur, comme nous allons le présenter dans la section suivante.

1.2 Quelques noyaux de la littérature

La fonction de l'équation (1.1) a un principal défaut : lorsque le nombre d'éléments dans les sacs comparés est très grand, elle a tendance à "noyer" les produits scalaires forts au milieu des produits scalaires faibles. Par exemple, supposons qu'il existe une mise en correspondance de produit scalaire 1 entre deux éléments de deux sacs B_i et B_j, et que toutes les autres valeurs de produit scalaire sont approximativement égales à ε, alors la valeur finale de cette fonction est $1+(|B_i||B_j|-1)\varepsilon \simeq 1+|B_i||B_j|\varepsilon$. Ainsi, le poids des ε devient rapidement plus important que la valeur de mise en correspondance. Cela se produit même dans le cas de valeurs d'amplitudes relativement faibles, par exemple avec $|B_i| = |B_j| = 100$ et $\varepsilon = 0.01$, la valeur cumulée des non-mises en correspondance est 100 fois plus grande que la valeur de mise en correspondance !

Dans le but de pallier ce problème, Lyu [51] a proposé une amélioration du noyau de l'équation (1.1) :

$$K_{lyu}(B_i, B_j) = \frac{1}{|B_i|}\frac{1}{|B_j|}\sum_r \sum_s k(\mathbf{b}_{ri}, \mathbf{b}_{sj})^p \tag{1.3}$$

avec $p \geq 1$ pour conserver un noyau de Mercer. Notons que dans le cas d'un noyau mineur gaussien, cela revient à considérer des gaussiennes plus fines.

D'autres chercheurs ont proposé des solutions plus radicales, comme Wallraven [90] :

$$K_{avmax}(B_i, B_j) = \frac{1}{|B_i|}\sum_r \max_s k(\mathbf{b}_{ri}, \mathbf{b}_{sj}) + \frac{1}{|B_j|}\sum_s \max_r k(\mathbf{b}_{ri}, \mathbf{b}_{sj}) \tag{1.4}$$

Dans ce cas, l'idée de mise en correspondance est plus évidente, étant donné que l'on compte le nombre de descripteurs du sac B_i proches d'un des descripteurs du sac B_j, et vice-versa. Malheureusement, cette fonction n'est pas de Mercer [51] – il faut se tourner vers des méthodes à noyaux indéfinis comme nous l'avons décrit dans la section à la page 14.

Notons que des solutions encore plus radicales peuvent être utilisées, par exemple :

$$K_{max}(B_i, B_j) = \max_r \max_s k(\mathbf{b}_{ri}, \mathbf{b}_{sj}) \tag{1.5}$$

Dans ce cas, la fonction renvoie la valeur de la meilleure mise en correspondance. Ce noyau est aussi indéfini.

Des propositions ont aussi été faites pour se baser sur une modélisation plus complexe que la moyenne des éléments sur sacs (Eq. 1.2), par exemple en modélisant chaque sac B_i par une gaussienne de moyenne μ_i et de variance Σ_i. Dans ce cas, un noyau de Bhattacharyya peut être utilisé pour comparer les modèles [42] :

$$K_{kondor}(B_i, B_j) = |\Sigma_i|^{-\frac{1}{4}} |\Sigma_j|^{-\frac{1}{4}} |\Sigma^\star|^{\frac{1}{2}} e^{\left(-\frac{1}{4}\mu_i^\top \Sigma_i^{-1}\mu_i - \frac{1}{4}\mu_j^\top \Sigma_j^{-1}\mu_j + \frac{1}{2}\mu^{\star\top}\Sigma^\star\mu^\star\right)} \tag{1.6}$$

avec $\Sigma^\star = \left(\frac{1}{2}\Sigma_i^{-1} + \frac{1}{2}\Sigma_j^{-1}\right)^{-1}$ et $\mu^\star = \frac{1}{2}\Sigma_i^{-1}\mu_i + \frac{1}{2}\Sigma_j^{-1}\mu_j$. Cette formule travaille dans l'espace initial des descripteurs, mais il est tout à fait possible d'effectuer les mêmes opérations dans un espace induit par une fonction noyau. Pour ce faire, Kondor [42] propose d'effectuer une analyse en composantes principales noyautée (KPCA), mais qui implique une dépendance à une forme de dictionnaire visuel.

1.3 Propositions

1.3.1 Noyaux majeurs

Le noyau proposé par Lyu [51] (Eq. 1.3) met bien en valeur les mises en correspondances, mais de grandes valeurs de p tendent à le rendre beaucoup trop discriminant. L'auteur montre dans son article que son noyau tend vers la forme suivante lorsque $p \to +\infty$:

$$K_{lyu}(B_i, B_j) \to \frac{1}{|B_i|}\sum_r \max_s k(\mathbf{b}_{ri}, \mathbf{b}_{sj})^p + \frac{1}{|B_j|}\sum_s \max_r k(\mathbf{b}_{ri}, \mathbf{b}_{sj})^p \tag{1.7}$$

Ainsi, en augmentant la puissance p, on se rapproche du noyau de l'équation (1.4), mais avec des noyaux mineurs $k^p(.,.)$ qui tendent vers l'identité.

Dans le but de réduire cet effet secondaire, nous avons fait les propositions suivantes [ICIP07] :

$$K_{power}(B_i, B_j) = \frac{1}{|B_i|}\frac{1}{|B_j|}\left(\sum_r\sum_s k(\mathbf{b}_{ri}, \mathbf{b}_{sj})^p\right)^{\frac{1}{p}} \tag{1.8}$$

$$K_{avpower}(B_i, B_j) = \frac{1}{|B_i|}\sum_r\left(\sum_s k(\mathbf{b}_{ri}, \mathbf{b}_{sj})^p\right)^{\frac{1}{p}} + \frac{1}{|B_j|}\sum_s\left(\sum_r k(\mathbf{b}_{ri}, \mathbf{b}_{sj})^p\right)^{\frac{1}{p}} \tag{1.9}$$

L'idée derrière ces propositions est de se rapprocher du noyau de l'équation (1.4), mais tout en conservant le degré de liberté offert par le paramètre p. En effet, dans le cas de l'équation (1.8), si $p = 1$ nous avons le noyau de Shawe-Taylor (Eq. 1.1) qui généralise le plus. Si $p = +\infty$, nous tendons vers le noyau de l'équation (1.5), qui repose sur une et une seule mise en correspondance (la meilleure). Enfin, entre ces deux extrêmes, nous avons une large gamme de possibilités.

Notons que nous n'avons pas réussi à prouver que ces deux fonctions noyau sont de Mercer, sans pour autant avoir trouvé un contre-exemple qui prouverait qu'elles ne le sont pas. La ressemblance avec la distance de Minkowski nous invite à penser que ces fonctions sont de Mercer, et qu'il reste à trouver une preuve pour le démontrer.

1.3.2 Gestion des redondances

Jusqu'à présent, nous n'avions pas traité le problème des éléments redondants dans les sacs. En effet, étant donné que nous regroupons les descripteurs dans des sacs (et non des ensembles), un même descripteur peut être présent plusieurs fois, ou une copie très proche. Cela pose des problèmes dans un contexte de comptage de mises en correspondance. Par exemple, si une image possède de nombreuses régions de ciel, elle va être très proche de toutes les images possédant au moins une région de ciel.

Pour traiter ce problème, nous proposons d'associer à chaque descripteur un poids qui représente sa singularité au sein du sac, autrement dit une valeur qui représente la similarité d'un descripteur aux autres descripteurs du même sac. La première chose à faire est donc de déterminer le poids qui représente la singularité du descripteur. Pour ce faire, nous nous appuyons sur une autre propriété des noyaux sur sacs que nous n'avons pas encore présentée. En effet, bien que ces fonctions soient conçues pour comparer des sacs, elles fonctionnent aussi très bien pour comparer un unique élément avec tout un sac, tout simplement en comparant le sac au singleton. Pour se faire une idée de la signification de cette opération, on peut retourner sur la figure 1.1. Sur cette base, la pondération α_{ri} associée au descripteur $\mathbf{b}_{ri}$ du sac que nous proposons est la suivante [CIVR07a] :

$$\alpha_{ri} = \frac{1}{K(\{\mathbf{b_{ri}}\}, B_i)} \tag{1.10}$$

avec $K(\{b_{ri}\}, B_i)$ la valeur de la fonction noyau majeure entre le sac et l'un de ses descripteurs. Dans la plupart des cas cela s'interprète comme le comptage des mises en correspondances entre le descripteur et tous les autres.

Cette pondération est ensuite appliquée à l'image des descripteurs $\mathbf{b}_{ri}$ dans l'espace induit par la fonction mineure k :

$$\hat{\phi}(\mathbf{b}_{ri}) = \alpha_{ri}\phi(\mathbf{b}_{ri}) \tag{1.11}$$

La fonction noyau majeure pondérée $\hat{K}$ est la fonction K utilisée avec le noyau mineur $\hat{k}(x, y) = \langle\hat{\phi}(x), \hat{\phi}(y)\rangle$. Ce processus peut être utilisé avec n'importe quelle fonction noyau majeure K, par exemple dans le cas de l'équation (1.1) nous avons :

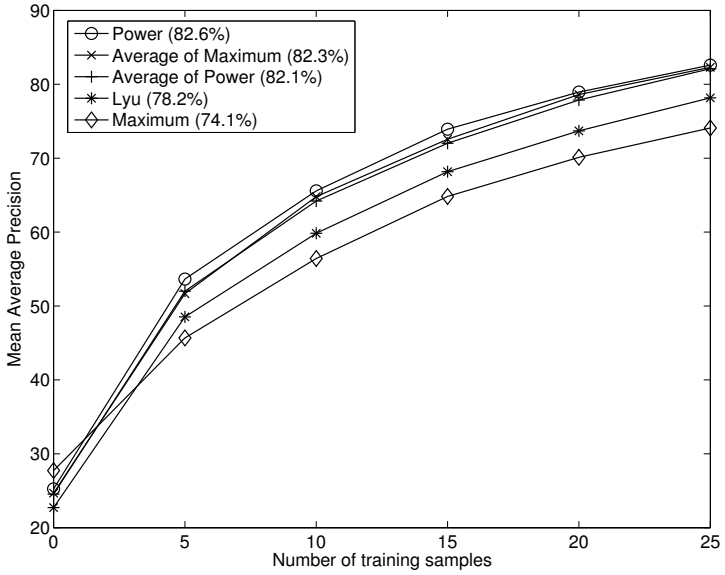
$$
\begin{aligned}
\hat{K}(B_i, B_j) &= \langle \hat{\Phi}(B_i), \hat{\Phi}(B_j) \rangle \\
&= \left\langle \sum_r \hat{\phi}(\mathbf{b}_{ri}), \sum_s \hat{\phi}(\mathbf{b}_{sj}) \right\rangle \\
&= \sum_r \sum_s \langle \alpha_{ri} \phi(\mathbf{b}_{ri}), \alpha_{sj} \phi(\mathbf{b}_{sj}) \rangle \\
&= \sum_r \sum_s \alpha_{ri} \alpha_{sj} k(\mathbf{b}_{ri}, \mathbf{b}_{sj})
\end{aligned}
$$

1.4 Expériences

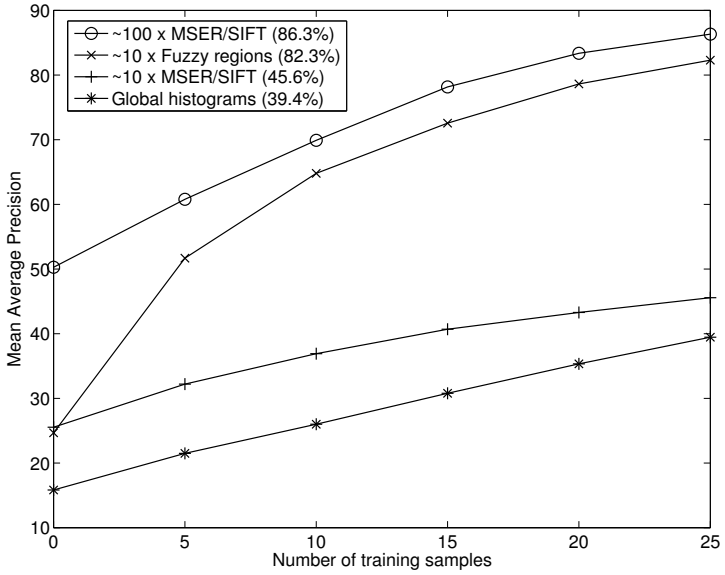
Nous présentons ici des résultats sur une base jouet formée de 600 objets sur fonds aléatoires (cf section A.2.1). Cette base a été conçue pour permettre de vérifier la capacité des différentes techniques à gérer les différents points qui nous intéressent. Chaque image est construite à partir d'un fond (un paysage) choisi aléatoirement dans une base d'images de paysage, et d'un objet issu de la base COIL-100 pris sous un certain angle de vue. Les objets ont été choisis pour permettre d'assurer l'hypothèse de mise en correspondance, et sont pris sous différents points de vue pour tester la capacité des méthodes à généraliser. Les fonds ont été ajoutés pour évaluer la capacité des méthodes à différencier les bonnes et les mauvaises mises en correspondance. Les fonds sont généralement très semblables, et il y a un bon nombre de doublons. Les résultats sont produits à partir de simulations de recherches interactives (cf section A.1.1).

Les premières expériences menées utilisent une représentation des images sous la forme de sacs d'histogrammes couleur et gradient, où chaque histogramme représente une région floue de l'image. Nous avons testé les fonctions suivantes : "Lyu" (Eq. (1.3)) avec $p = 1.4$, "Maximum" (Eq. (1.5)), "Average of Maximum" (Eq. (1.4)), "Power" (Eq. (1.8)) avec $p = 1.6$ et "Average of Power" (Eq. (1.9)) avec $p = 5$. Toutes ces fonctions utilisent un noyau mineur gaussien χ^2 et la pondération proposée. Nous présentons les résultats sur la figure 1.2(a) avec le meilleur paramétrage trouvé par cross-validation pour des valeurs de p entre 1 et 5. Tout d'abord, dans tous les cas, les fonctions sont en mesure de trouver les caractéristiques des objets. Remarquons que le plus mauvais noyau est le noyau "Maximum", ce qui conforte l'idée qu'une seule mise en correspondance n'est pas suffisante pour obtenir les meilleurs résultats, même dans un cas d'école. Puis, nous pouvons observer le gain apporté par nos propositions une fois comparées au noyau de Lyu. Nos propositions ont, au final, les mêmes performances que le noyau "Average of Maximum" en dépit du paramètre p que l'on peux régler. Cela est peut être dû au fait que, pour cette base, le meilleur paramétrage est celui qui conduit à une équivalence entre les noyaux proposés et le noyau "Average of Maximum".

Nous avons aussi mené des expériences avec d'autres descripteurs visuels, dont les résultats sont présentés dans la figure 1.2(b). Nous présentons tout d'abord les résultats obtenus en utilisant une description globale par histogramme couleur et texture ("Global histograms"). Si on compare aux noyaux sur sacs avec 10 régions floues en moyenne par image ("Fuzzy regions"), on peut constater le gain très important qu'offre une approche locale pour ce type de problème. Nous avons aussi comparé aux régions MSER [57] décrites par des SIFT [49]. Les résultats sont présentés avec le meilleur choix de noyau, pour 10 ou 100 régions MSER en moyenne par image. Les meilleures performances sont obtenues avec 100 régions MSER, cependant au prix d'une plus grande complexité. Si on compare à complexité égale, à savoir 10 régions en moyenne par image, les régions floues sont plus intéressantes.



(a) Comparaison de différents noyaux avec les meilleurs paramètres.



(b) Comparaison de différents descripteurs visuels.

FIGURE 1.2 – Mean Average Precision(%) sur la base d'objets sur fond aléatoire, en fonction du nombre d'annotations.

1.5 Conclusion

Les noyaux sur sacs offrent un cadre formel très intéressant pour combiner mise en correspondance et apprentissage. D'un point de vue théorique, ce cadre offre la possibilité de travailler de manière implicite dans l'espace des combinaisons de descripteurs. Cette propriété permet de comparer les sacs l'un relativement à l'autre, sans avoir à souffrir d'une forte dépendance à un macro-paramètre global, comme un dictionnaire visuel. Nous avons présenté des solutions proposées dans la littérature, ainsi que des améliorations que nous proposons [ICIP07,CIVR07a]. Ces techniques ont été évaluées sur un problème de recherche interactive d'objet dans une base d'image. Les résultats ont montré la capacité de ces techniques à gérer avec succès une telle situation.

Ces travaux effectués lors de mon post-doctorat ont forgé les bases des études qui ont suivi sur les noyaux sur graphes. En effet, les techniques que nous avons présentées ne tiennent pas compte de la répartition dans le documents des caractéristiques visuelles. Une approche pertinente pour intégrer cette information est de modéliser les documents sous la forme d'un graphe, puis de concevoir des fonctions noyaux sur ces graphes. C'est dans ce contexte qu'a été proposé le sujet de la thèse de Justine Lebrun, qui s'est focalisée sur les aspects théoriques de la mise en œuvre des noyaux sur graphes. Peu de temps après, la thèse de Jean-Emmanuel Haugeard a démarré sur le même thème, mais cette fois ci avec une approche plus applicative, dans le cadre du projet ANR iTowns.

Chapitre 2

Noyaux sur graphes

Nous avons montré dans le chapitre précédent l'intérêt d'une représentation sous la forme d'un ensemble de descripteurs, pour la résolution de problèmes avec des mises en correspondance, comme la recherche d'un objet. Il est néanmoins possible d'être encore plus performant en ajoutant la répartition des descripteurs dans l'espace du document. Pour ce faire, il existe plusieurs formes de modélisation, et parmi celles-ci nous nous sommes intéressés à celle par graphe. L'idée ici est de représenter chaque primitive visuelle (région, point d'intérêt, etc.) sous la forme d'un sommet d'un graphe, puis chaque relation spatiale (à gauche, en haut, perpendiculaire, etc.) sous la forme d'une arête entre deux sommets du graphe.

Plus particulièrement, nous nous intéressons aux fonctions noyaux sur graphes, dans le but de pouvoir les utiliser dans un contexte d'apprentissage. Un bon point de départ pour présenter un rapide historique des noyaux sur graphes est de s'intéresser aux noyaux de convolution [30]. Ce modèle très généraliste est la base de nombreuses méthodes [23]. D'autres noyaux ont aussi été proposés par la suite, et peuvent être répartis par familles en fonction du type de structure qu'ils considèrent. L'une de ces familles de noyaux est basée sur des ensembles de chemins issus du graphe. L'idée est de sommer les similarités entre les différents chemins de deux graphes. En pratique l'intégralité des chemins n'est pas considérée, et diverses approches sont proposées pour choisir le meilleur tirage. Par exemple, on peut considérer que les chemins sont issus de marches aléatoires [38, 81, 16], ou encore s'appuyer sur le produit direct entre graphes [3, 88].

Une autre famille compare les graphes en sommant les similarités entre les sous-graphes élémentaires (*graphlets*) des deux graphes [78]. Ces méthodes sont motivées par les limites des chemins aléatoires, qui ne peuvent discerner certains graphes [52]. Notons que ces cas arrivent surtout lorsque les graphes ont des sommets dont les similarités ont des valeurs binaires. Des propositions ont aussi été faites en considérant des arbres élémentaires [52, 77].

Beaucoup de noyaux de la littérature ont été construits pour la chimie ou les applications bio-informatiques, où les sommets et les arêtes contiennent peu d'information, généralement une étiquette voire un vecteur de faible dimension (moins de 4). De plus, ces méthodes ont été proposées pour des graphes de petit taille, mises à part certaines comme [78], mais qui ne fonctionnent que sur des sommets non étiquetés. Dans le contexte de la recherche multimédia, ces méthodes ont besoin d'être adaptées.

Ces travaux ont été menés tout d'abord dans le cadre de la thèse de Justine Lebrun. Cette thèse avait pour but la conception de méthodes à noyau pour la mise en correspondance des graphes. Ces premiers travaux étaient plus théoriques, et ont fait l'objet de publications [ICPR08,EGC08,IVC11]. Peu de temps après le démarrage de cette thèse, une seconde sur un thème similaire a commencé avec Jean-Emmanuel Haugeard, mais cette fois-ci avec une approche plus applicative. En effet, l'objectif ici était de produire un détecteur d'objet dans les façades de Paris, dont l'un des composants est un classifieur sur graphes (cf section 2.3.3). Les modèles de noyaux sur graphes de la première thèse ont été utilisés comme base pour fabriquer les noyaux adaptés aux graphes produits à partir des façades. Ces travaux ont aussi fait l'objet de publications [ICPR10b,PRL]. Enfin, le framework que nous avons mis en place a aussi été utilisé dans le cadre du stage master de Mathias Fuzier, pour classer des objets en trois dimensions [ACM10].

Notons que nous ne présentons ici que les principaux résultats de la thèse de J. Lebrun.

2.1 Principe

Dans ce chapitre, nous notons $G = (V, E)$ un graphe composé d'un ensemble de sommets V et d'un ensemble d'arêtes E.

2.1.1 Sacs de chemins

L'approche que nous proposons de suivre est d'interpréter tout graphe comme étant un ensemble de chemins. Pour représenter cette transformation, nous considérons une fonction $H(G)$ qui à un graphe G fait correspondre un sac de chemins. La nature du sac renvoyé dépend des types de chemins désirés – par exemple ceux avec ou sans cycles. Une fois cette fonction choisie, les fonctions noyaux que nous considérons sont basées sur une mise en correspondance entre les chemins issus de chaque graphe selon H. Les graphes étant vus comme des sacs, on peut utiliser un noyau sur sacs, par exemple celui de l'équation (1.1) :

$$K_H(G, G') = \sum_{h \in H(G)} \sum_{h' \in H(G')} K_C(h, h') \tag{2.1}$$

avec $K_C(h, h')$ une fonction noyau qui compare deux chemins h et h'. Un chemin h est une séquence de n sommets $v_1, \ldots, v_n$ issus du graphe G, chaque sommet v_i étant relié au sommet suivant v_{i+1} par l'arête e_i.

Les fonctions noyaux sur chemins les plus répandues sont des produits entre les similarités de sommets et d'arêtes :

$$K_C(h, h') = K_V(v_1, v'_1) \prod_{i=2}^{n} K_V(v_i, v'_i) K_E(e_{i-1}, e'_{i-1}) \tag{2.2}$$

avec K_V une fonction noyau qui compare les descripteurs des sommets et K_E une fonction noyau qui compare les descripteurs des arêtes.

L'idée derrière cette métrique est assez intuitive : deux chemins sont similaires si et seulement si tous leurs sommets et arêtes sont similaires.

2.1.2 Quelques exemples de la littérature

Une méthode qui illustre bien le principe des noyaux sur sacs de chemins est celle de Kashima [37]. Cette méthode fait partie des méthodes qui s'appuient sur des marches aléatoires pour modéliser puis comparer les graphes. Elle fait suite aux travaux de Tsuda [35], Vishwanathan [89] et Smola [22].

Le noyau $K_{kashima}(G, G')$ est la somme pondérée des comparaisons entre tous les chemins h de G et h' de G' de même longueur :

$$K_{Kashima}(G, G') = \sum_{n=1}^{\infty} \sum_{\substack{h \in H(G) \\ |h|=n}} \sum_{\substack{h' \in H(G') \\ |h'|=n}} K_C(h, h') p(h|G) p(h'|G') \tag{2.3}$$

avec $|h|$ la longueur du chemin h.

Kashima utilise des noyaux sur sommets et arêtes qui renvoient des valeurs entre 0 et 1. Lorsque les descripteurs sont des vecteurs, l'utilisation de noyaux gaussiens L^2 est préconisée. Cependant, il reste tout à fait possible d'utiliser d'autres noyaux dès lors qu'ils renvoient des valeurs entre 0 et 1.

La probabilité $p(h|G)$ associée à un chemin h est calculé de la manière suivante :

$$p(h|G) = p_s(v_1) \prod_{i=2}^{n} p_t(v_i \to v_{i+1}) p_q(v_n)$$

avec $p_s(v)$ la probabilité (uniforme) de trouver le sommet v dans G, $p_t(v_i \to v_{i+1})$ la probabilité de transition de v_i à v_{i+1}, et $p_q(v)$ la probabilité que le chemin s'arrête au sommet v définie comme suit, avec $|V|$ le nombre de sommets dans le graphe :

$$\forall v_i \in V, \quad \sum_{j=1}^{|V|} p_t(v_j \to v_i) + p_q(v_i) = 1$$

En théorie, le noyau proposé doit comparer tous les chemins jusqu'à une longueur infinie. Cependant, le caractère récursif de l'expression permet d'exprimer le problème sous la forme d'un système linéaire fini. De plus, les valeurs des probabilités et celles des noyaux sur sommets et arêtes étant inférieures à 1, la convergence du calcul est assurée [37].

Cependant, comme le montre Suard [81], cette méthode a une forte tendance à considérer de nombreuses fois les mêmes arêtes. Cet aspect est peut-être peu gênant pour des graphes de molécules, mais dans le contexte des graphes issus d'images, un très grand nombre de petites valeurs de similarité sont ainsi sommées, et finissent par diluer les similarités les plus fortes. Cela pose aussi des problèmes en terme de complexité calculatoire. Dans le but de résoudre ces

problèmes, Suard propose une version approximée de la méthode de Kashima :

$$K_{Suard}(G,G') = \frac{1}{2}\left(\sum_{h\in H(G)} \max_{\substack{h'\in H(G')\\|h'|=|h|}} K_C(h,h') + \sum_{h'\in H(G')} \max_{\substack{h\in H(G)\\|h|=|h'|}} K_C(h,h')\right) \tag{2.4}$$

Cette fonction est composée de deux parties pour assurer la symétrie. Le premier terme (et de même le second) somme les similarités entre un chemin h de G et son meilleur appariement de même longueur dans G'. Cela permet d'éviter de sommer trop de petites valeurs de similarités.

La méthode propose aussi de considérer un ensemble de chemins moins vaste, à savoir l'ensemble des plus courts chemins entre deux sommets du graphe. Ainsi, il y a au plus $|V|^2$ (resp. $|V'|^2$) chemins considérés dans G (resp. G'), et donc au plus $|V|^2 \times |V'|^2$) comparaisons de chemins.

Au final, pour des graphes d'ordre n, cette méthode permet de passer d'une complexité $O(n^6)$ à $O(n^4 + 2n\log(n))$. Les expériences menées montrent des performances légèrement inférieures à celles de Kashima, mais pour un temps de calcul bien meilleur.

2.2 Propositions

2.2.1 Noyaux sur sacs de chemins

Notre objectif dans cette section est de proposer des fonctions noyaux adaptées aux particularités des graphes issus des images [ICPR08,EGC08,IVC11]. En effet, dans ce contexte, les valeurs des similarités entre les sommets et arêtes évoluent de manière assez uniforme dans un intervalle. Prenons l'exemple des noyaux mineurs basés sur des gaussiens χ^2, dont les valeurs possibles sont entre 0 et 1. Dans la pratique, il est très rare de trouver deux sommets dont la similarité est égale à 1, et de même pour les valeurs proches de 0. Avec un paramétrage classique, deux sommets visuellement proches ont une similarité de l'ordre de 0.9, et deux sommets visuellement différents ont une similarité de l'ordre de 0.1. Puis, pour toutes les autres possibilités, la notion de similitude est toute relative. Par conséquent, suivre une stratégie où l'on accumule un très grand nombre de similarités n'est pas efficace dans ce domaine. Les problèmes dont souffrent certains noyaux comme celui de Kashima s'en trouvent donc amplifiés.

Dans le but de répondre à cette problématique, nous proposons de nous intéresser à une famille de fonctions noyaux qui va effectuer une sorte de filtrage sur les similarités à accumuler. Plus précisément, nous proposons de ne considérer que les couples de chemins issus des deux graphes dont la similarité entre les premiers sommets est élevée. Cela peut s'exprimer de la manière suivante :

$$K(G,G') = \sum_{v\in V}\sum_{v'\in V'} K_A(v,v')\hat{K}\left(H_v(G), H_{v'}(G')\right) \tag{2.5}$$

avec :

– $G = (V,E)$ et $G' = (V',E')$ les deux graphes à comparer ;

- $K_A(v, v')$ une fonction qui va déterminer si le couple de sommets (v, v') doit être pris en compte ;
- $\hat{K}(H, H')$ une fonction qui compare deux ensembles de chemins H et H' ;
- $H_v(G)$ une fonction qui renvoie un ensemble de chemins issus de G dont le premier sommet est v.

L'idée est alors de jouer sur la fonction $K_A(v, v')$ pour ne prendre en compte que les similarités pertinentes.

Notons que cette formulation intègre les formules précédentes. Par exemple, si on choisit $K_A = 1$, mises à part les probabilités, on peut retrouver une formule similaire à l'équation (2.3) avec :

$$\hat{K}_{somme}(H, H') = \sum_{n=1}^{\infty} \sum_{\substack{h \in H \\ |h|=n}} \sum_{\substack{h' \in H' \\ |h'|=n}} K_C(h, h') \tag{2.6}$$

Dans le cas où l'on peut déterminer via un seuil θ si deux sommets sont similaires ou non (comme dans le cas des points d'intérêt), on peut utiliser la fonction K_A suivante :

$$K_{A_{seuil}}(v, v') = \begin{cases} 1 & \text{si } K_V(v, v') < \theta \\ 0 & \text{sinon} \end{cases} \tag{2.7}$$

Si on a la possibilité de construire un dictionnaire de sommets, on peut alors utiliser la fonction suivante :

$$K_{A_{dict}}(v, v') = \sum_r a_r(v) a_r(v') \tag{2.8}$$

avec $a_r(v)$ une fonction qui renvoie 1 si v correspond au mot r du dictionnaire, 0 sinon.

Enfin, si on ne peut ni fixer un seuil ni calculer un dictionnaire, on peut alors se tourner vers une fonction qui ne sélectionne que les couples les plus proches :

$$K_{A_{ppv}}(v, v') = \frac{1}{|V|} p_{v'}(v) + \frac{1}{|V'|} p_v(v') \tag{2.9}$$

avec

$$p_v(v') = \begin{cases} 1 & \text{si } v' \in \mathrm{ppv}_k(v) \\ 0 & \text{sinon} \end{cases} \tag{2.10}$$

et $\mathrm{ppv}_k(v)$ une fonction qui renvoie les k plus proches sommets de v dans G'.

Toujours dans le but de réduire le nombre de similarités accumulées, il est aussi possible de jouer sur la fonction $\hat{K}(H, H')$ qui compare deux ensembles de chemins. Pour ce faire, nous proposons de calculer la valeur de la plus grande similarité entre les chemins des deux ensembles :

$$\hat{K}(H, H') = \max_{n \in [1,N]} \max_{\substack{h \in H \\ |h|=n}} \max_{\substack{h' \in H' \\ |h'|=n}} K_C(h, h') \tag{2.11}$$

Avec une telle fonction, il y autant de valeurs de similarité accumulées dans l'équation (2.5) que de couples de sommets sélectionnés par la fonction $K_A(v, v')$. Un autre avantage de

cette formule est la recherche d'une valeur maximale. En effet, avec des noyaux sur chemins $K_C(h, h')$ récursifs, il est possible d'utiliser des algorithmes de calcul rapide.

La formulation finale que nous proposons est la suivante (synthèse des équations (2.5), (2.9) et (2.11)) :

$$\begin{aligned} K_{new}(G, G') &= \frac{1}{|V|} \sum_{\substack{v \in G \\ v' \in \mathrm{ppv}_k(v)}} \max_{h \in H_v(G)} \max_{\substack{h' \in H_{v'}(G') \\ |h'|=|h|}} K_C(h, h') \\ &+ \frac{1}{|V'|} \sum_{\substack{v' \in G \\ v \in \mathrm{ppv}_k(v')}} \max_{h' \in H_{v'}(G')} \max_{\substack{h \in H_v(G) \\ |h|=|h'|}} K_C(h', h) \end{aligned} \tag{2.12}$$

Cette fonction n'est pas une fonction de Mercer, mais peut cependant être utilisée dans des algorithmes d'apprentissage spécialisés (voir page 14).

2.2.2 Noyaux sur chemins

Pour compléter un noyau sur un ensemble de chemins, il est nécessaire de définir un noyau sur ces chemins : $K_C(h, h')$. Dans cette section nous nous intéressons à ces noyaux et proposons diverses solutions possibles. Ces noyaux sont basés sur des noyaux sur sommets K_V et sur arêtes K_E. Nous supposons que ce sont des noyaux qui retournent des valeurs comprises entre 0 et 1.

Un noyau sur chemins très utilisé dans la littérature effectue le produit entre toutes les similarités des sommets et arêtes composant les deux chemins (Eq. 2.2) :

$$K_{C_{mul}}(h, h') = K_V(v_1, v'_1) \times \prod_{i=2}^{|h|} K_V(v_i, v'_i) K_E(e_{i-1}, e'_{i-1}) \tag{2.13}$$

Lorsque les noyaux mineurs retournent des valeurs entre 0 et 1 la fonction décroit toujours avec l'augmentation de la taille du chemin, et par conséquent les similarités des longues chaînes ont de très faibles valeurs. Lorsqu'il est utilisé avec un noyau sur sacs de chemins qui somme l'ensemble des similarités (comme dans l'équation (2.3)), cette fonction ne pose pas de problèmes. Cependant, lorsqu'elle est utilisée avec un noyau sur sacs de chemins qui recherche les meilleures similarités (comme les équations (2.4) ou (2.12)), seules sont accumulées les similarités des chemins les plus courts.

Un autre type de noyau sur chemins a été proposé dans la littérature, comme celui qui somme les similarités entre sommets pondérés par les similarités entre arêtes [68] :

$$K_{C_{som}}(h, h') = K_V(v_1, v'_1) + \sum_{i=2}^{|h|} K_V(v_i, v'_i) K_E(e_{i-1}, e'_{i-1}) \tag{2.14}$$

Cette fonction augmente de façon systématique avec la taille des chemins si les noyaux mineurs sont positifs. Les similarités des chemins courts seront inférieures aux mêmes chemins prolongés. Cependant, cela n'est pas le plus gênant étant donné qu'il est de plus en plus difficile de trouver des chemins de grande longueur dont tous les sommets sont très similaires. Le

principal reproche que l'on pourrait faire est que, contrairement au noyau précédent, quelques sommets/arêtes très similaires suffisent pour considérer que les deux chemins sont similaires.

Dans le but de pallier aux problèmes de ces fonctions noyaux, mais aussi de mieux répondre aux particularités des graphes issus des images, nous proposons les deux fonctions suivantes.

Nous proposons tout d'abord le noyau suivant, dont le premier but est d'améliorer celui de l'équation (2.14) :

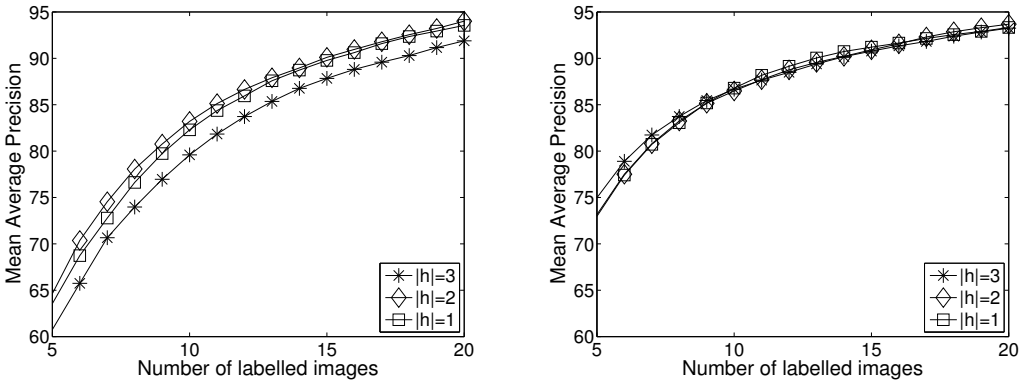
$$K_{C_{new1}}(h, h') = K_V(v_1, v'_1) \times \prod_{i=1}^{|h|} \Big(1 + K_V(v_i, v'_i) K_E(e_{i-1}, e'_{i-1})\Big) \tag{2.15}$$

À l'aide d'une telle formule, nous obtenons une augmentation de la similarité avec la longueur des chemins comme dans (2.14). Cependant, en utilisant un produit, nous pénalisons fortement tout couple de chemin qui a au moins un couple sommet/arête non similaire.

Étant donné que les descripteurs sur arêtes sont généralement de plus faible dimension (et dont les similarités sont par conséquent plus franches), il peut être intéressant de les traiter différemment :

$$K_{C_{new2}}(h, h') = K_V(v_1, v'_1) \times \prod_{i=1}^{|h|} \Big(1 + K_V(v_i, v'_i)\Big) \times K_E(e_{i-1}, e'_{i-1}) \tag{2.16}$$

Contrairement aux autres, cette fonction n'est pas monotone avec la taille des chemins. Dans le cas où les chemins h et h' représentent des régions disposées dans l'image de la même manière, cette fonction va augmenter avec la taille des chemins. Dans l'autre cas, où au moins une des régions est disposée différemment, la similarité va chuter de manière significative. Ainsi, seules les ensembles de régions disposés de manière similaires seront similaires. En plus de donner un poids important aux relations spatiales, cette fonction va aussi permettre des calculs plus rapides étant donné que davantage de comparaisons de chemins seront ignorées par les algorithmes rapides.



(a) Noyau de Kashima (2ms/50ms/700ms) (b) Noyau proposé (2ms/2.5ms/3ms)

FIGURE 2.1 – Mean Average Precision(%) sur la base d'objets sur fond aléatoire, en fonction du nombre d'annotations. Les temps entre parenthèses (t1/t2/t3) sont les durées moyennes pour calculer une valeur de la fonction noyau utilisée pour des longueurs de chemins de longueur 1, 2 et 3.

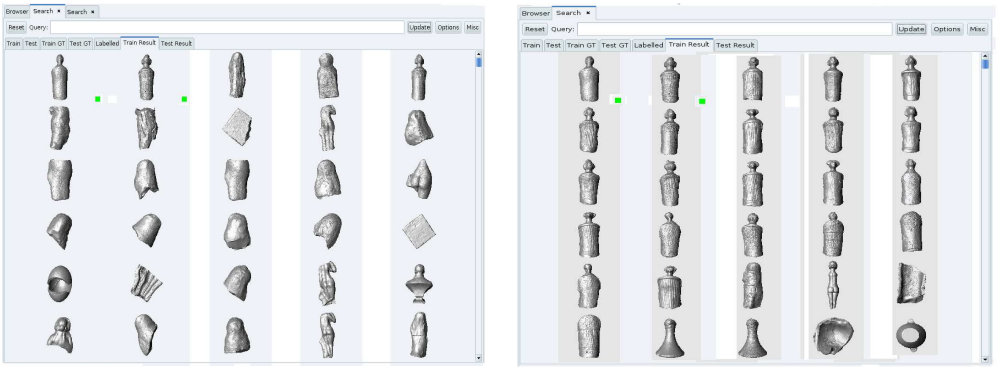
2.3 Expériences et applications

2.3.1 Recherche interactive dans les bases d'images

Nous avons mené des premières expériences sur la base d'objets sur fonds aléatoires (cf section A.2.1) dans le but de valider les noyaux proposés [ICPR08,EGC08,IVC11]. Les régions, les descripteurs visuels et le protocole sont les mêmes que dans le cas des noyaux sur sacs du chapitre précédent. Les résultats sont présentés dans la Figure 2.1 pour deux noyaux sur graphes. Le premier est celui de Kashima (Eq. (2.3)) qui fait office de référence, et celui que nous proposons (Eq. (2.12)) avec le noyau sur chemin de l'Eq. (2.16).

Tout d'abord, notons le gain d'environ 10% que l'on obtient par rapport aux noyaux sur sacs, avec 20 annotations. Ce résultat montre ainsi l'intérêt d'exploiter les relations spatiales entre les primitives d'une image. Si on observe l'évolution des performances du noyau de Kashima en fonction de la longueur des chemins, on peut constater que l'on progresse de 1 à 2, mais que l'on régresse en passant à une longueur de 3. Cela est le résultat du très grand nombre de similarités de chemins que somme le noyau de Kashima, qui finit pas "noyer" les fortes similarités ($\Leftrightarrow$ mises en correspondance). De son côté, le noyau que nous proposons ne souffre pas de ce problème. Notons qu'il n'y a pas de différence de performance entre les différentes longueurs de chemin. Cela est très certainement dû au fait que les ensembles de paires de régions adjacentes sont suffisantes pour décrire un objet.

Outre la qualité des résultats, il existe une importante différence entre les deux méthodes en temps de calcul. Par exemple, pour une longueur de 2, le noyau de Kashima est en moyenne 20 fois plus long à calculer que celui proposé. La différence est encore plus importante pour une longueur de 3, avec un temps de calcul 230 fois plus coûteux. Cette différence s'explique par la possibilité d'utiliser un algorithme de Branch and Bound avec notre noyau, ce qui permet d'éviter une très grande partie des calculs.



(a) Descripteurs globaux (b) Graphes

FIGURE 2.2 – Résultats de recherche interactive avec 2 annotations positives et 7 annotations négatives.

2.3.2 Recherche dans les bases d'objet 3D

Les noyaux sur graphes que nous avons présentés dans ce chapitre sont utilisés pour la recherche interactive dans les bases d'objet 3D, dans le cadre du projet ANR EROS 3D. La base utilisée est une collection de 750 modèles 3D non texturés d'objets anciens (vases grecs, figurines gallo-romaines, etc.). Ces modèles ont été fournis par le Centre de Recherche et de Restauration des Musées de France (C2RMF).

FIGURE 2.3 – Exemples de segmentation d'objets en trois dimensions

Nous avons mené des expériences pour déterminer l'intérêt des noyaux sur graphes dans ce contexte (Stage M. Fuzier [ACM10]). Les premières expériences utilisent une description globale des objets 3D, sont la forme d'un histogramme calculé selon la méthode CEGI. Nous présentons un résultat avec cette description globale en Figure 2.2(a). Les secondes expériences utilisent une description par graphe, où chaque sommet du graphe représente une région surfacique de l'objet, et chaque arête l'adjacence entre deux régions. Les régions ont été obtenues en utilisant un algorithme par montée des eaux (*watershed*), dont des exemples sont présentés en Figure 2.3. Les sommets/régions sont représentés avec des histogrammes calculés selon la méthode CEGI. Nous présentons un résultat avec cette description par graphe en Figure 2.2(b). Si nous comparons au résultat précédent, nous pouvons apprécier le gain offert par la méthode que nous proposons.

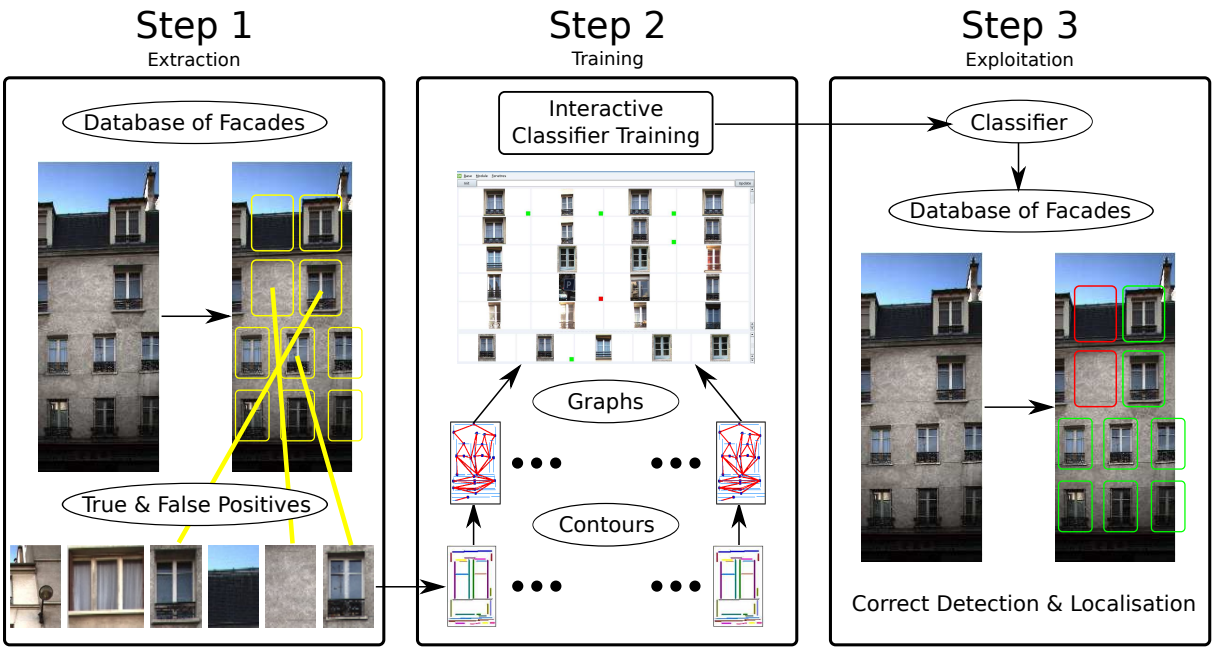


FIGURE 2.4 – Détection des fenêtres dans les façades de Paris.

2.3.3 Recherche dans les façades de Paris

Les noyaux sur graphes que nous avons présentés dans ce chapitre sont utilisés pour la détection des fenêtres dans les façades de Paris (Thèse J-E. Haugeart [ICPR10b,PRL]), dans le cadre du projet ANR iTowns.

Le processus se déroule en 3 étapes résumées dans la figure 2.4.

Etape 1 Des hypothèses de fenêtres sont extraites des images de façades de manière automatique en utilisant une amélioration de la méthode de Lee et Nevatia [45]. La méthode est conçue pour produire très peu de faux négatifs.

Etape 2 L'ensemble des hypothèses sont regroupées pour former une base d'image. On extrait les contours dans chaque image, puis on forme un graphe dont chaque sommet représente un contour et chaque arête la proximité entre deux contours. Les graphes sont ensuite comparés en utilisant un noyau sur graphe proche de celui de l'équation (2.12), mais avec des noyaux sur chemins adaptés à la description par contours. La recherche interactive est alors utilisée pour fabriquer un classifieur capable de distinguer les bonnes des mauvaises hypothèses de fenêtre.

Etape 3 Le classifieur d'hypothèses de fenêtres produit à l'étape 2 est introduit dans le processus final de détection. Ce processus n'est composé que de l'extraction automatique des hypothèses et du raffinement par le classifieur.

2.4 Conclusion et perspectives

Les fonctions noyaux sur graphes sont une extension naturelle des noyaux sur sacs, et tout particulièrement si on travaille avec des noyaux sur sacs de chemins. Nous avons présenté dans ce chapitre des propositions qui permettent de produire des fonctions noyaux pertinentes pour le contexte de la recherche interactive par le contenu. Outre le gain que nous avons obtenu par rapport aux noyaux sur sacs, nous avons aussi réussi à produire des fonctions noyaux avec un temps de calcul raisonnable.

Un autre intérêt de l'approche que nous avons suivie est sa capacité à pouvoir traiter différentes problématiques. En effet, bien qu'initialement conçus pour rechercher dans des bases d'images représentées par des régions, les outils que nous proposons s'adaptent facilement à d'autres cas, comme la recherche d'objets en trois dimensions ou la recherche dans des graphes de contours.

Les techniques par noyaux sur ensembles, que ce soit sacs ou graphes, restent parmi les plus performantes pour la recherche par mise en correspondance. En effet, elles reproduisent d'une part les mécanismes des systèmes de vote, qui sont encore parmi les meilleurs pour la recherche par similarité. D'autre part, les techniques par noyaux sur ensembles sont compatibles avec les méthodes avancées d'apprentissage, ce qui permet d'obtenir des performances toujours plus importantes grâce à un ensemble d'exemples.

Cependant, ces techniques souffrent d'un principal défaut : leur coût calculatoire. Bien que certaines fonctions permettent de réduire les temps de calculs grâce à des algorithmes comme le Branch and Bound, les temps restent prohibitifs pour traiter des bases de très grandes taille (au delà du million). Pour résoudre ce problème, nous présentons au chapitre suivant une solution basée sur les dictionnaires dynamiques. Parallèlement à ceci, une piste possible est de traduire le calcul sur ensemble de descripteurs en un calcul sur vecteurs, par exemple en utilisant des séries de Taylor. Pour effectuer ce travail, on doit bien entendu s'assurer de conserver au mieux les propriétés des noyaux sur ensembles, comme la robustesse au paramétrage, au risque de retomber dans les solutions par dictionnaires.

Deuxième partie

Représentations Dynamiques

Chapitre 3

Noyaux Dynamiques

Une approche très courante pour rechercher dans les bases multimédia est d'utiliser des dictionnaires visuels. Comme nous l'avons présentée dans les chapitres précédents, cette approche est problématique dans le cas de la recherche interactive, lorsque le dictionnaire est déterminé une fois pour toutes, pendant la phase de pré-traitement de la base (phase "hors ligne", telle que présentée dans la figure 3.1). Nous avons présenté des solutions basées sur les fonctions noyaux sur ensembles pour traiter ce problème pour la recherche d'objet. Ces techniques fonctionnent très bien pour ce type de recherche grâce à leur fort pouvoir discriminant. Cependant, pour des recherches plus généralistes, comme la recherche de catégorie, cette forte capacité à discriminer n'est plus aussi intéressante.

Dans le but de concilier recherche interactive et dictionnaires visuels, nous proposons de déplacer la construction du dictionnaire de la phase "hors ligne" à celle "en ligne", tel que présenté dans la figure 3.2. L'idée est de s'appuyer sur les annotations fournies par l'utilisateur pour construire le dictionnaire le plus adapté. Nous nous détachons ainsi d'un processus lourd de réglage d'un dictionnaire général, et bien plus encore, étant donné que chaque session de recherche aura son propre dictionnaire.

Pour ce faire, nous devons faire face aux contraintes de la recherche interactive. La première concerne le faible nombre d'annotations fournies par l'utilisateur. Par conséquent, il n'est plus possible de raisonner comme dans le cas où les exemples sont très nombreux, par exemple on ne peut plus faire d'hypothèses fortes sur la capacité à généraliser des classifieurs. La deuxième contrainte qu'il faut gérer est le peu de temps de calcul dont on dispose en recherche interactive. En effet, il n'est pas envisageable de faire patienter l'utilisateur plusieurs dizaines de minutes entre chaque itération du processus de bouclage de pertinence. Ainsi, une approche naïve qui consiste à reproduire l'intégralité du processus de calcul du dictionnaire tel qu'il est effectué hors-ligne n'est pas possible. D'une manière générale, il faut se tourner vers des solutions capables de modifier à moindre frais le dictionnaire, et par conséquent les signatures et les similarités qui en découlent.

La problématique de l'apprentissage d'un dictionnaire visuel dans un contexte de recherche interactive est très peu explorée. Nous pouvons toutefois noter les améliorations qui ont été proposées dans la littérature pour réduire le temps de calcul d'un dictionnaire. Par exemple, on trouve des méthodes rapides pour calculer des dictionnaires pour la représentation parci-

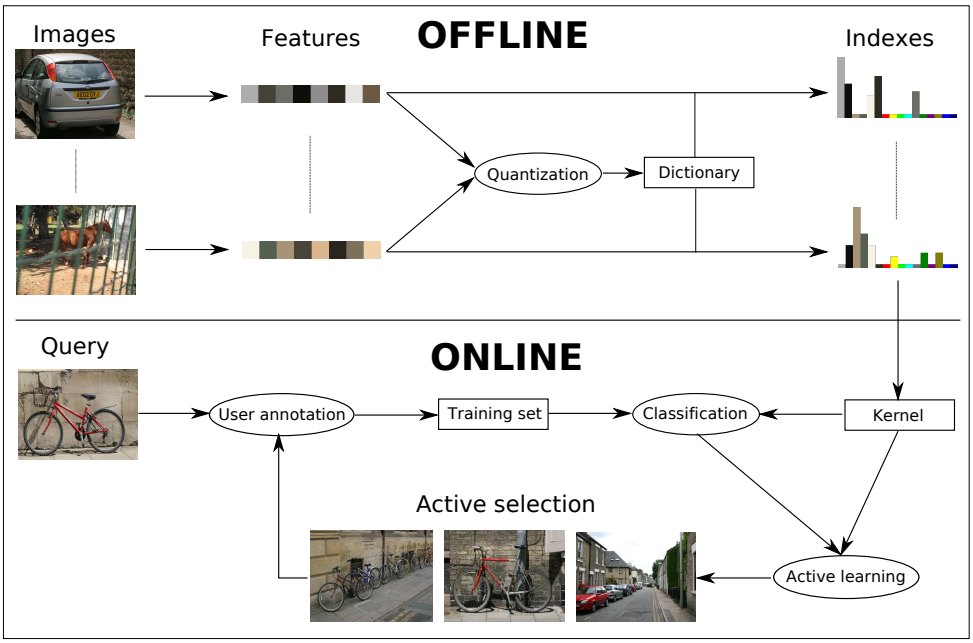


FIGURE 3.1 – Architecture habituelle des systèmes de recherche interactive, où le calcul du dictionnaire est effectué hors ligne.

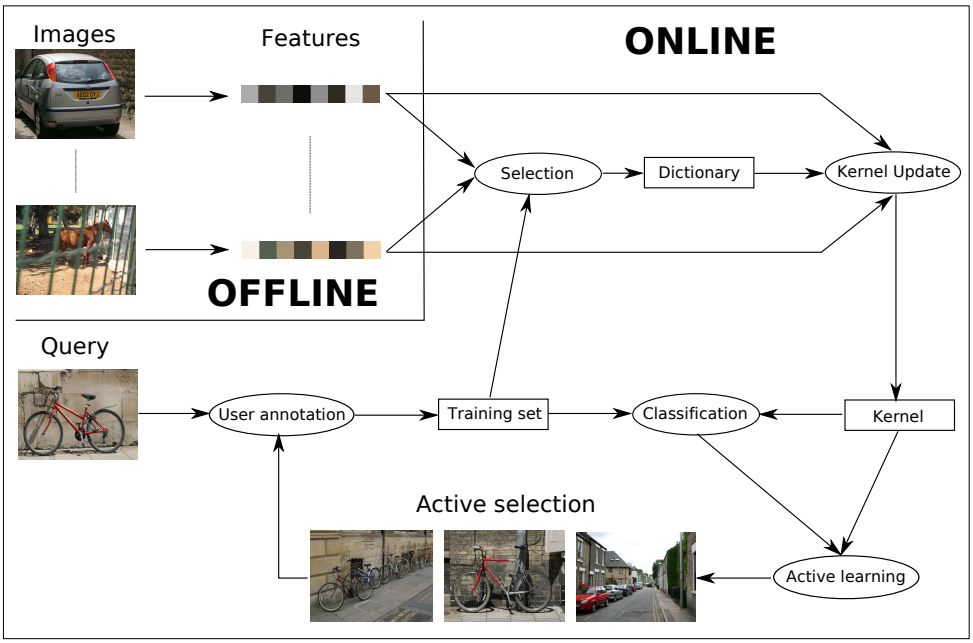


FIGURE 3.2 – Architecture proposée pour un système de recherche interactive, où le calcul du dictionnaire est effectué en ligne.

monieuse des images, certaines de manière non-supervisée [46] et d'autres de manière supervisée [54, 53]. Cependant, bien que ces méthodes soient bien plus rapide que les précédentes, leur complexité reste encore trop importante pour la recherche interactive.

3.1 Principe

3.1.1 Noyaux et dictionnaires

Dans le but de trouver le dictionnaire le plus adapté à la recherche courante, nous proposons de nous appuyer sur les combinaisons de fonctions noyaux :

$$K = \sum_{t=1} \beta_t k_t \tag{3.1}$$

avec k_t des fonctions noyaux dit *de base* prises dans un ensemble $\mathcal{K}$, et $\beta_t \in \mathbb{R}$.

L'idée est la suivante. Si chaque noyau de base correspond à un mot visuel, l'ensemble des noyaux de base inclus dans la combinaison correspond à un dictionnaire visuel. Ainsi, en optimisant l'équation (3.1) en fonction des annotations utilisateurs, nous formons à la fois une métrique et un dictionnaire adaptés.

Cette approche entre dans la catégorie des méthodes de combinaisons de noyaux ("Multiple Kernel Learning") [43], méthodes qui connaissent un franc succès ces dernières années. La plupart des techniques ont été proposées dans un contexte d'apprentissage par lots, cependant certaines ont attiré notre attention de par leur caractère incrémental, et tout particulièrement celles basées sur le boosting [9, 31]. Ces techniques sélectionnent les noyaux de base l'un après l'autre, tant qu'il existe un noyau de base intéressant. De plus, elles fabriquent les noyaux de base pendant le processus de sélection, contrairement aux autres techniques de combinaisons qui travaillent sur un ensemble de noyaux de base pré-établi.

3.1.2 Ensemble dynamique de noyaux de base

L'approche que nous proposons de suivre est de sélectionner de manière itérative les noyaux de base à ajouter à la combinaison finale. Cependant, nous n'utilisons ni un ensemble de noyaux de base pré-établi, ni un ensemble formé pendant le processus de sélection, mais une solution intermédiaire.

L'utilisation d'un ensemble pré-établi est intéressant du point de vue de la phase de classification, étant donné qu'il devient possible de pré-calculer les valeurs des fonctions de base. Cependant, plus l'ensemble de noyaux de base est grand, plus nous avons de choix, et donc de chances de trouver une combinaison intéressante. Plus particulièrement, dans le cas où chaque noyau de base représente un mot visuel, nous avons constaté qu'il faut des dizaines, voire des centaines de milliers de noyaux de base dans un ensemble pré-établi pour obtenir de bons résultats. Avec une telle configuration, la phase de recherche de la meilleure combinaison devient extrêmement coûteuse en temps de calcul.

La formation de l'ensemble de noyaux de base pendant le processus de sélection ne dépend plus d'un ensemble de noyaux de base pré-établi, mais d'un ensemble de familles de noyaux

de base. À chaque itération de sélection, on forme un noyau de base à partir de chaque famille, avec pour objectif l'amélioration du critère de sélection. Le nombre de familles est généralement faible, mais la formation de noyaux de base est souvent très coûteuse. De plus, étant donné que l'on ne peut prévoir à l'avance les noyaux ainsi formés, on ne peut pas pré-calculer leurs valeurs, ce qui pose problème lors de la phase de classification de la base.

Dans le but de répondre aux contraintes de la recherche interactive, nous proposons une gestion particulière de l'ensemble des noyaux de base. Tout d'abord, nous formons des noyaux de base à partir de chaque image. En d'autres termes, pour chaque image i de la base, nous formons un ensemble $\mathcal{K}_i$, par exemple en utilisant les principales couleurs de l'image i. Puis, lors de la phase interactive, nous formons un ensemble $\mathcal{K}$ de noyaux de base avec l'ensemble des noyaux de base des images annotées positivement. Autrement dit, $\mathcal{K} = \cup_{y_i>0}\mathcal{K}_i$, avec $y_i = 1$ si l'image i est annotée comme pertinente, et $y_i = -1$ dans le cas contraire. De cette manière, nous construisons une combinaison de noyaux en considérant un ensemble réduit de noyaux de base, et donc une complexité elle aussi réduite. De plus, cela permet de se focaliser sur des mots visuels pertinents pour la recherche actuelle, et réduit les mauvaises sélections dues au faible nombre d'exemples. En outre, les noyaux de base sont connus à l'avance, et via un système de cache, leurs valeurs peuvent être soit calculées à la volée, soit prises dans le cache si elles ont déjà été calculées aux bouclages ou aux sessions de recherche précédents.

3.1.3 Alignement du noyau

Dans le but de déterminer l'intérêt d'une combinaison de noyaux, nous utilisons l'alignement du noyau [10], qui est une mesure de similarité entre une matrice noyau à évaluer (dans notre cas la combinaison) et une matrice noyau cible (la matrice parfaite) :

$$\mathcal{A}_L(K) = \frac{\langle K, L\rangle_F}{||K||_F||L||_F} \tag{3.2}$$

avec K la matrice noyau $n \times n$ à évaluer, L la matrice noyau $n \times n$ cible, $\langle K, L\rangle_F$ le produit scalaire de Frobenius, et $||K||_F$ la norme de Frobenius.

Dans [10], Cristianini propose d'utiliser le noyau cible $\mathbf{y}\mathbf{y}^\top$, avec $y_i \in \{-1, 1\}$ l'annotation du document i. Ce choix est pertinent pour des ensembles d'apprentissage équilibrés, où il y a autant d'annotations positives que négatives. Pour les autres cas, un tel noyau cible donne davantage de poids à la classe la plus peuplée. Dans le cas de la recherche interactive, où la classe négative est bien plus importante que la classe positive, les algorithmes qui utilisent ce noyau cible apprennent davantage la classe négative – celle qui n'intéresse pas l'utilisateur.

Dans le but de gérer ce déséquilibre des annotations, il est possible d'utilise le noyau cible $\tilde{\mathbf{y}}\tilde{\mathbf{y}}^\top$, avec :

$$\tilde{y}_i = \begin{cases} \frac{1}{n^+} & \text{si } y_i > 0 \\ -\frac{1}{n^-} & \text{si } y_i < 0 \\ 0 & \text{sinon} \end{cases}$$

et n^+ et n^- les tailles respectives des classes positives et négatives.

Une approche encore plus intéressante est d'utiliser l'alignement centré [40] :

$$\mathcal{A}_L^H(K) = \frac{\langle HKH, HLH \rangle_F}{||HKH||_F ||HLH||_F} \tag{3.3}$$

avec :

$$\begin{aligned} L_{ij} &= \begin{cases} 1 & \text{si } y_i = y_j \\ 0 & \text{sinon} \end{cases} \\ H &= I - \tfrac{1}{n}\mathbf{1}\mathbf{1}^\top \end{aligned}$$

et I la matrice identité de taille n, $\mathbf{1}$ le vecteur colonne de taille n dont toutes les valeurs sont égales à 1.

L'alignement centré gère le déséquilibre de la même manière qu'avec le noyau cible $\tilde{\mathbf{y}}\tilde{\mathbf{y}}^\top$, mais possède aussi des invariances :

- Invariance à l'échelle : $\mathcal{A}_L^H(\lambda K) = \mathcal{A}_L^H(K)$
- Invariance à la translation : $\mathcal{A}_L^H(K + \lambda \mathbf{1}\mathbf{1}^\top) = \mathcal{A}_L^H(K)$

3.2 Modèle proposé

Nous présentons dans cette section le modèle que nous proposons, avec deux versions possibles pour la recherche de la combinaison de noyaux. Dans tous les cas, nous supposons qu'un ensemble $\mathcal{K}$ de noyaux de base est formé à partir des ensembles $\mathcal{K}_i$ de noyaux de base issus des images i annotées positivement.

3.2.1 Apprentissage rapide de combinaisons

Méthode simple

La première méthode que nous proposons est assez simple. On évalue l'intérêt de chacun des noyaux de base dans $\mathcal{K}$ à l'aide de l'alignement centré du noyau, on élimine ceux plus mauvais que le hasard (alignement négatif), on sélectionne les T meilleurs, puis on somme le tout pour former le noyau final :

$$K = \sum_{k \in S} k \tag{3.4}$$

avec

$$S = \underset{k \in \mathcal{K}, \mathcal{A}_L^H(k) > 0}{\operatorname{argmax}} \mathcal{A}_L^H(k), \quad |S| \leq T$$

Le principal intérêt de cette méthode est son faible coût en calculs lors de l'apprentissage, coût principalement constitué par le calcul des alignements. Cependant, cette méthode ne gère pas les redondances dans $\mathcal{K}$. Si on trouve plusieurs noyaux de base identiques ou presque identiques dans $\mathcal{K}$, la méthode les sélectionnera tous. Le choix du nombre T de noyaux de base est délicat. Une solution simple est de poser $T = +\infty$, autrement dit de prendre tous les noyaux de base meilleurs que le hasard. Notons que, tant que l'ensemble $\mathcal{K}$ n'est pas trop grand, cette méthode fonctionne bien, et même bien plus que l'on ne pourrait l'imaginer.

Méthode avancée

La deuxième méthode que nous proposons effectue véritablement une sélection et pondération incrémentales. Nous démarrons avec un noyau $K_0 = 0$, puis sélectionnons le noyau de base $k_1^\star$ qui maximise l'alignement avec $\beta_1^\star = 1$. Les suivants sont alors sélectionnés en recherchant le meilleur couple (poids, noyau) qui améliore l'alignement :

$$K_t^\star = K_{t-1} + \beta_t^\star k_t^\star \tag{3.5}$$

avec

$$\beta_t^\star, k_t^\star = \underset{\beta>0, k\in\mathcal{K}}{\operatorname{argmax}} \mathcal{A}_L^H(K_{t-1} + \beta k) \tag{3.6}$$

La résolution de l'équation (3.6) peut être obtenue en utilisant des techniques d'algèbre linéaire [40]. D'un point de vue calculatoire, le principal coût de cet algorithme est le calcul du Frobenius entre chaque noyau de base sélectionné et l'ensemble des noyaux de base de $\mathcal{K}$. Tout le reste est négligeable ou peut être pré-calculé. Cette complexité est à comparer aux algorithmes proposés en apprentissage par lots qui, d'une manière grossière, nécessitent le calcul des Frobenius entre tous les noyaux de base de $\mathcal{K}$.

Notons qu'au fur et à mesure des itérations de sélection, on trouve de moins en moins de solutions à l'équation (3.6), pour au final ne plus en trouver du tout, étant donné que l'ensemble $\mathcal{K}$ ne change pas au cours des itérations de sélection.

Cette méthode peut être rapprochée des méthodes de construction de noyaux par boosting [9, 31]. La principale différence est que l'on ne construit pas à chaque itération de sélection un nouveau noyau de base qui améliore à coup sûr l'alignement. Or cette fonctionnalité est la garantie que, dans le cadre du boosting, l'algorithme converge dans tous les cas, en supposant que l'on n'a pas de limites matérielles. En supprimant cette fonctionnalité pour réduire les temps de calculs, nous faisons porter un lourd poids à l'ensemble $\mathcal{K}$. En effet, si cet ensemble était infini, nous aurions toutes les possibilités de noyaux de base, dont celles construites pendant les itérations de sélection, et donc une garantie de convergence. Naturellement, dans la pratique, un ensemble $\mathcal{K}$ infini n'est pas possible. Cependant, nous pouvons nous en approcher autant que possible en travaillant finement sur les noyaux de base mis dans $\mathcal{K}$.

3.2.2 Les noyaux de base

Pour pouvoir former notre combinaison de noyaux de manière itérative, il nous faut des noyaux de base qui se combinent bien de cette manière. Rappelons que notre objectif initial est la formation d'un dictionnaire visuel, autrement dit la sélection d'un ensemble de mots visuels, où chaque noyau de base est associé à un mot visuel. Or d'expérience nous savons que, pour assurer un maximum de sélection, un tel choix est bien plus pertinent que celui où chaque noyau de base représente quelque chose de complexe, comme un dictionnaire complet. En effet, dans le cas de noyaux de base "complexes", il est très difficile d'en sélectionner un deuxième lorsqu'on en a déjà choisi un premier. Pour s'en convaincre, rappelons que cette approche est similaire à celle du boosting, qui repose sur un ensemble de classifieurs "faibles" [74]. Dans

notre cas, l'équivalent de ces classifieurs "faibles" sont nos noyaux de base, qui doivent donc être aussi "faibles".

Pour ce faire, nous commençons par extraire et quantifier des descripteurs visuels dans chaque image i de la base. La quantification est effectuée à l'aide de la méthode de quantification vectorielle ELBG [65] avec une distance d. Nous obtenons ainsi un ensemble P_i d'attributs $\mathbf{p}_{ri}$ avec :

$$\mathbf{p}_{ri} = (\mathbf{f}_{ri}, h_{ri}, \theta_{ri}) \tag{3.7}$$

où $\mathbf{f}_{ri}$ est le centre du cluster r, h_{ri} la taille du cluster r, et θ_{ri} la distance de $\mathbf{f}_{ri}$ au centre du cluster le plus proche $r' \neq r$. Notons bien que ces ensembles P_i sont calculés indépendamment pour chaque image i. La quantification sert principalement à définir des mots visuels sous la forme d'hypersphères de centres $\mathbf{f}_{ri}$ et de rayons θ_{ri}, il n'y a pas de notion de dictionnaire à ce stade de la méthode.

Puis, lors d'une session de recherche, nous formons un nouvel ensemble de mots visuels candidats avec l'ensemble des ensembles P_i des images annotées positivement. Chacun de ces mots visuels candidats $\hat{\mathbf{p}}$ donne naissance à un nouveau noyau de base $k_{\hat{\mathbf{p}}}$. Enfin, les noyaux de base $k_{\hat{\mathbf{p}}}$ sont regroupés pour former un ensemble noté $\mathcal{K}$:

$$\mathcal{K} = \{k_{\hat{\mathbf{p}}} | \hat{\mathbf{p}} \in P_i \text{ et } y_i > 0\} \tag{3.8}$$

Les noyaux de base sont alors définis comme suit :

$$k_{\hat{\mathbf{p}}}(P_i, P_j) = \delta_{\chi^1}(e_{\hat{\mathbf{p}}}(P_i), e_{\hat{\mathbf{p}}}(P_j)) \tag{3.9}$$

avec

$$\delta_{\chi^1}(x, y) = \begin{cases} 1 - \frac{|x-y|}{x+y} & \text{si } x > 0 \text{ et } y > 0 \\ 0 & \text{sinon} \end{cases} \tag{3.10}$$

$$e_{\hat{\mathbf{p}}}(P_i) = \begin{cases} h_{r^\star i} & \text{si } d(\hat{\mathbf{f}}, \mathbf{f}_{r^\star i}) \leq \hat{\theta} \\ 0 & \text{sinon} \end{cases} \tag{3.11}$$

$$r^\star = \underset{r}{\operatorname{argmin}}\, d(\hat{\mathbf{f}}, \mathbf{f}_{ri}) \tag{3.12}$$

$$\hat{\mathbf{p}} = (\hat{\mathbf{f}}, \hat{h}, \hat{\theta}) \tag{3.13}$$

Chacun de ces noyaux de base peut être vu comme un noyau triangulaire avec une distance χ^1 calculée sur le bin d'un histogramme. Nous commençons par calculer le descripteur $\mathbf{f}_{r^\star i}$ de l'image P_i le plus proche du descripteur $\hat{\mathbf{f}}$ associé au noyau de base (Eq. (3.12)). Puis, si ce descripteur est dans l'hypersphère de centre $\hat{\mathbf{f}}$ et de rayon $\hat{\theta}$, nous renvoyons le nombre d'occurrences $h_{r^\star i}$ du descripteur $\mathbf{f}$ dans l'image P_i (Eq. (3.11)). Un calcul similaire est effectué pour l'image P_j, et pour finir les nombres d'occurrences dans chaque image sont comparés (Eq. (3.10)).

Ce faisant, la combinaison finale de tels noyaux de base est très proche d'un noyau triangulaire appliqué à des histogrammes issus d'un dictionnaire général. En effet, si on défini un vecteur $\mathbf{x}_i = (e_{\hat{\mathbf{p}}_1}(P_i), \ldots, e_{\hat{\mathbf{p}}_T}(P_i))$ pour chaque image i, alors :

$$
\begin{aligned}
K(P_i, P_j) &= \sum_t \beta_t k_{\hat{\mathbf{p}}_t} \\
&= \sum_t \beta_t \left(1 - \frac{|e_{\hat{\mathbf{p}}_t}(P_i) - e_{\hat{\mathbf{p}}_t}(P_j)|}{e_{\hat{\mathbf{p}}_t}(P_i) + e_{\hat{\mathbf{p}}_t}(P_j)}\right) \\
&= \left(\sum_t \beta_t\right) - \sum_t \beta_t \frac{|x_{ti} - x_{tj}|}{x_{ti} + x_{tj}} \\
&\simeq 1 - \left(\sum_t \beta_t\right)^{-1} d^{\boldsymbol{\beta}}_{\chi^1}(\mathbf{x}_i, \mathbf{x}_j)
\end{aligned}
$$

On obtient un noyau triangulaire avec $\sigma = \sum_t \beta_t$ et une distance du χ^1 pondérée par $\boldsymbol{\beta}$. Ce noyau est appliqué sur les vecteurs $\mathbf{x}_i$, qui regroupent le nombre d'occurrences des mots visuels, autrement dit un histogramme.

3.2.3 Apprentissage actif

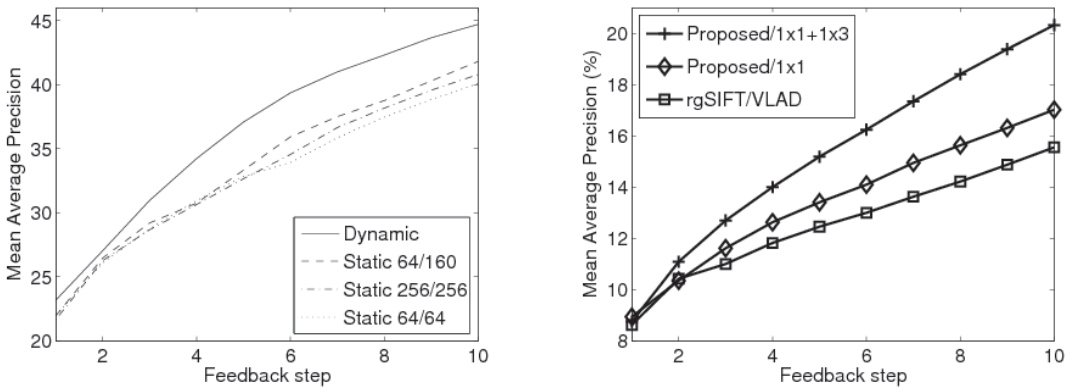
Pour gérer la sélection des images à faire annoter par l'utilisateur, nous optons ici aussi pour une approche de pré-sélection pessimiste suivie d'une approche optimiste. La pré-sélection est effectuée en utilisant la méthode de [IP08], une amélioration de la méthode SVM_{active}[83].

La sélection au sein de cette pré-sélection est propre à la méthode que nous proposons ici. Compte tenu de l'importance de l'ensemble des noyaux de base, nous proposons de sélectionner les images qui contiennent les mots visuels candidats les plus prometteurs. Dans cet esprit, la fonction de sélection $a(P_i)$ de l'apprentissage actif est définie comme la valeur du meilleur alignement que l'on peut obtenir avec les descripteurs visuels contenus dans l'image i :

$$
a(P_i) = \underset{\mathbf{p} \in P_i}{\operatorname{argmax}} \mathcal{A}^H_L(k_{\mathbf{p}}) \tag{3.14}
$$

En utilisant une telle fonction de sélection, nous augmentons nos chances de former un bon ensemble de noyaux de base.

Notons qu'une alternative est d'estimer le gain que pourrait apporter l'ajout d'une nouvelle annotation en tenant compte de l'ensemble du processus de combinaison de noyau. Autrement dit, pour chaque image pré-sélectionnée, nous supposons qu'elle est annotée positivement, nous effectuons tout le calcul de combinaison, et enfin nous notons l'alignement obtenu. Cette approche doit probablement être plus efficace, mais compte tenu de son coût calculatoire, nous ne l'avons pas étudiée.



(a) Base VOC2006 (b) Base VOC2007

FIGURE 3.3 – Mean Average Precision(%) en fonction du nombre d'itérations de bouclage. Initialisation avec 1 image, 10 bouclages de pertinence, 5 annotations par bouclage.

3.3 Expériences et applications

3.3.1 Recherche interactive dans les bases d'images

Comparaison statique/dynamique. Nous avons mené des premières expériences pour comparer l'approche par dictionnaire/noyau dynamique à une approche équivalente par dictionnaire/noyau statique. Ces expériences ont été menées sur la base VOC 2006 (cf section A.2.2) en suivant le protocole de recherche interactive présenté en section A.1.1. Dans tous les cas, les descripteurs visuels sont des couleurs $L^\star a^\star b^\star$ (lab) et des textures par ondelettes quaternioniques (qw).

Pour l'approche statique, nous avons construit toutes les combinaisons de dictionnaire avec 32, 64,. . .ou 256 couleurs et 32, 64,. . .ou 256 textures. Chacun de ces dictionnaires a été évalué en recherche interactive avec un noyau gaussien χ^1 et un classifieur SVM. Nous présentons sur la figure 3.3(a) les résultats de trois combinaisons : 64 couleurs et 64 textures (un paramétrage naturel compte tenu de la taille de la base), 256 couleurs et 256 textures (le plus grand dictionnaire testé), et 64 couleurs et 160 textures (le meilleur paramétrage). Notons bien que pour déterminer le meilleur dictionnaire, nous avons utilisé la vérité terrain, une information qui n'est pas disponible dans la réalité. On peut remarquer qu'un plus grand dictionnaire ne conduit pas nécessairement à de meilleures performances, et qu'un petit dictionnaire peut aussi être très performant. Par exemple, pour le cas de la couleur, 64 mots est le meilleur choix.

Pour l'approche dynamique, nous avons suivi la méthode simple présentée en section 3.2.1. On a extrait et quantifié 64 couleurs et 64 textures de chaque image (cf section 3.2.2). Les résultats sont présentés sur la figure 3.3(a), où l'on peut observer le gain notable de l'approche dynamique. Rappelons que, contrairement à l'approche statique, de tels résultats ne nécessitent pas le réglage d'un dictionnaire ! Nous avons aussi testé l'influence du nombre de noyaux de base sélectionnés (le paramètre T dans les formules). Si nous ne fixons pas de limite, la méthode sélectionne en moyenne 200 noyaux de base à la fin de chaque session de recherche. Si on

diminue le nombre maximal de noyaux de base sélectionnés, on observe une légère perte jusqu'à $T = 96$ (MAP 43%), puis on descend assez rapidement (MAP 34% avec $T = 16$). Notons que la méthode avancée (section 3.2.1) donne des résultats assez proches, cependant elle offre ces performances avec beaucoup moins de noyaux de base (environ 30-35 noyaux de base sont sélectionnés en fin de session).

Comparaison avec une méthode de référence. Nous avons aussi comparé l'approche que nous proposons à une méthode de référence qui bénéficie aussi d'une bonne robustesse aux paramètres globaux. Pour trouver une telle méthode, nous avons testé de nombreuses techniques proposées dans la littérature, et avons sélectionné la méthode par "Vector of Locally Aggregated Descriptors (VLAD)" [34], une approximation des noyaux de Fisher [67]. Pour chaque image, cette méthode fabrique une description vectorielle à partir de ses descripteurs visuels et d'un dictionnaire visuel global de faible taille. Toute la robustesse de la méthode provient de la faible taille du dictionnaire visuel (entre 16 et 64 mots), qui n'a que peu d'influence sur les résultats. Les résultats de cette méthode sont présentés en figure 3.3(b) sur la base VOC2007 (cf section A.2.3). Nous avons choisi cette base pour ces expériences de par son succès dans la littérature. Les résultats de la méthode avancée (cf section 3.2.1) sont aussi présentés dans cette figure, avec deux jeux de descripteurs visuels. Le premier (noté "1x1" sur la figure) est le même que celui des expériences précédentes, et le second (noté "1x1+1x3" sur la figure) est aussi le même, auquel nous avons ajouté des descripteurs (couleurs et textures) extraits uniquement dans une des trois parties horizontales des images (tiers supérieur, tiers central et tiers inférieur), pour un total de 8 différents types de descripteurs visuels. Ces résultats montrent l'intérêt de l'approche que nous proposons, mais aussi sa capacité à gérer plusieurs types de descripteurs visuels.

Temps de calcul. Sur une machine dotée d'un processeur iCore7 et sans cache partagé entre les sessions, toutes les méthodes présentées ici ont un temps d'apprentissage (noyau et classifieur) inférieur à 250ms, et un temps de classification inférieur à 200ms par tranche de 5000 images. Côté apprentissage, les méthodes statiques sont les plus rapides (moins de 1ms). Côté classification, les méthodes sont d'autant plus rapides que le dictionnaire est petit, plaçant la méthode avancée que nous proposons parmi les meilleures (environ 25ms par tranche de 5000 images). Notons que, dès lors qu'un cache est mis en place pour partager les valeurs entre les sessions de recherche, tous les temps de classification chutent très rapidement quelle que soit la méthode, pour atteindre quelques millisecondes.

3.3.2 Recherche dans les façades de Paris

L'approche dynamique que nous proposons a été utilisée pour proposer une alternative rapide aux noyaux sur graphes, dans le cadre de la recherche dans les façades de Paris que nous avons présentée en section 2.3.3. Tout le processus reste identique, hormis la fonction noyau qui est dynamique.

Les descripteurs visuels considérés sont des chemins issus des graphes correspondant à chaque image, et les noyaux de base comparent non plus un nombre d'occurrences, mais la similarité à un chemin associé au noyau de base. Plus précisément, la fonction $e_{\hat{\mathbf{p}}}(P_i)$ de l'équation (3.11) renvoie la similarité au chemin $\hat{\mathbf{p}}$ le plus proche dans l'ensemble des chemins

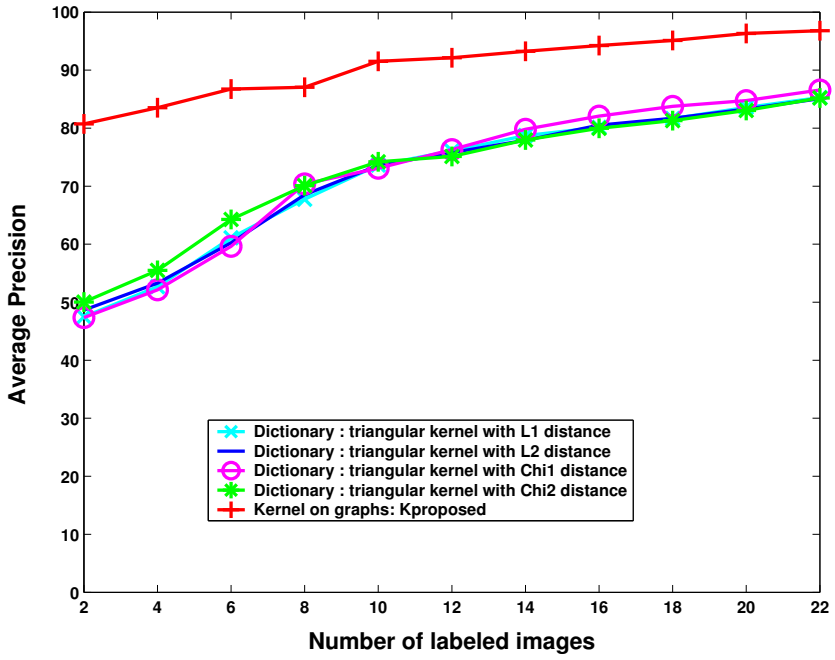


FIGURE 3.4 – Average Precision(%) en fonction du nombre d'annotations, avec différents noyaux.

P_i issus du graphe de l'image i.

Les résultats sont présentés dans la figure 3.4, pour le meilleur noyau sur graphes et 4 noyaux dynamiques avec différentes distances (fonction δ dans l'équation (3.9)). On peut observer que nous avons une perte en terme de performances – les noyaux sur graphes restent, dans le cas des problèmes de mise en correspondance, l'une des meilleures approches. Cependant, on observe un important gain en temps de calcul, si l'on en juge par le temps moyen pour calculer une valeur de la fonction noyau (ici avec des chemins de longueur 8) :

Sommets dans le graphe	26	30	40	124
Noyau sur graphe	40ms	96ms	184ms	511ms
Noyau dynamique	5ms	6ms	9ms	48ms

3.3.3 Apprentissage par lots dans les bases d'images

Bien que ce ne soit pas l'objectif initial, nous avons utilisé les noyaux dynamiques pour faire de l'apprentissage par lots. Le but dans ce cas est de construire un classifieur pour une catégorie donnée sur la base d'un grand ensemble d'apprentissage. Pour ce faire, nous avons suivi le protocole officiel pour la base VOC2006 (cf section A.2.2). Les résultats sont présentés sur la figure 3.1 avec le critère d'évaluation utilisé lors de la campagne.

Signatures	Non supervisé		Supervisé	Supervisé + Non supervisé	
Attributs	lab/qw(a)	vlad(b)	lab/qw(c)	lab/qw+vlad(d)	Fisher(e)
bicycle	90	90	91	95	94
bus	95	95	96	97	98
car	93	94	95	96	97
cat	80	89	87	91	93
cow	82	92	92	93	94
dog	73	81	82	87	87
horse	84	87	86	92	92
motorbike	93	92	95	97	96
person	77	81	81	87	86
sheep	88	92	90	94	95
tout	85	89	89	93	93

TABLE 3.1 – Area under ROC Curve(%) sur l'ensemble de test de la base VOC2006. lab = couleurs $L^\star a^\star b^\star$, qw = textures par ondelettes quaternioniques, vlad = Vector of Locally Aggregated Descriptors avec Harris-Laplace/rgSIFT.

Nous avons commencé par évaluer les méthodes avec dictionnaires produits de manière non supervisée, l'une avec des couleurs et textures (colonne (a)) et l'autre avec des VLADs (colonne (b)). Dans tous les cas, nous présentons les résultats avec les meilleurs dictionnaires.

Puis, nous avons utilisé l'approche proposée dans ce chapitre pour produire des signatures supervisées, dans le sens où chaque image est représentée par une signature produite en utilisant les annotations sur l'ensemble d'entrainement. Pour les obtenir, nous avons lancé de nombreuses simulations de recherche interactive sur l'ensemble d'entraînement de la base VOC2006. A la fin de chaque recherche simulée, nous avons conservé la fonction de classification produite. Suite à cela, nous avons obtenu des centaines de classifieurs pour chaque catégorie. Ces classifieurs ont été appliqués aux images des ensembles d'apprentissage et de test pour produire les signatures supervisées, sous la forme d'un vecteur qui regroupe l'ensemble des sorties des classifieurs. Ces attributs, une fois utilisés avec un noyau gaussien L^2 et un classifieur SVM, produisent les résultats de la colonne (c). Une fois comparés aux résultats de la colonne (a), on peut constater le gain apporté par l'approche.

Enfin, nous avons combiné les attributs sémantiques de toutes les catégories avec les VLADs pour former une nouvelle description des images, dont les résultats sont présentés dans la colonne (d). Nous pouvons observer qu'ils permettent d'obtenir des résultats aussi bons que ceux de la méthode avec les noyaux de Fisher, dont les résultats officiels ont été reportés dans la colonne (e).

3.4 Conclusion et perspectives

Dans ce chapitre, nous avons présenté une approche d'apprentissage de noyau "en ligne" pour la recherche de catégories, qui satisfait les contraintes de la recherche interactive. Cette approche permet de proposer des solutions plus robustes à une paramétrisation globale d'un système de recherche, sans avoir à recourir à de forts a priori. De plus, elle peut être utilisée pour d'autres applications, comme une alternative rapide aux problèmes de mise en correspondance de graphes, ou encore l'apprentissage par lots.

De par sa conception, l'approche que nous proposons se prête à un passage à l'échelle, autrement dit au traitement de très grandes bases (plusieurs millions). En effet, étant donné que l'ensemble des noyaux de base utilisés pour la combinaison est connu à l'avance, il est possible de travailler sur la création d'index rapides, dans le but d'obtenir une classification de la base avec une complexité sous-linéaire. L'approche peut aussi être améliorée en effectuant la combinaison des noyaux en ciblant la qualité du classifieur final. En effet, dans le contexte de l'apprentissage par lots, il existe des méthodes de combinaison de noyaux qui construisent à la fois la combinaison et le classifieur en même temps. Par conséquent, Il nous semble tout à fait envisageable de proposer des méthodes similaires, capables de gérer les contraintes de l'apprentissage interactif.

La conception du framework des noyaux dynamiques a été conçu de manière indépendante, en dehors de tout projet ou thèse, et a fait l'objet de publication [PR10]. En outre, ce framework a été utilisé dans le cadre de la thèse de Jean-Emmanuel Haugeard [ICPR10b,PRL], dans le but de réduire les temps de calcul, comme nous l'avons présenté dans la section 3.3.2. Notons que ces travaux sont aussi à l'origine des études autour du boosting que nous présentons dans le chapitre suivant.

Chapitre 4

Boosting Interactif

Dans tous les chapitres précédents, nous avons présenté des méthodes basées sur les fonctions noyaux. Comme nous l'avons présentée, cette approche est motivée (entre autres) par la possibilité d'utiliser des classifieurs très performants tels que les SVMs. Cependant, le cadre des fonctions noyaux est très contraignant, et nécessite beaucoup de travail de conception.

Il existe cependant d'autres approches d'apprentissage qui permettent de faire de la recherche dans les bases multimédias. Parmi celles-ci, il existe celle par Boosting, qui est à la fois similaire au couple SVM/Noyau d'un certain point de vue, mais aussi très différente d'un autre point de vue. En effet, le Boosting est, au même titre que l'approche noyau, une technique de changement d'espace. On passe ainsi d'un espace quelconque (celui des index, histogrammes, graphes, etc.) à un espace vectoriel, mais non nécessairement Hilbertien. Puis, on forme un hyperplan séparateur dans cet espace. Par contre, la philosophie du Boosting est très différente en ce qui concerne la manière d'apprendre. En effet, on ne cherche plus à résoudre le problème en s'appuyant sur des métriques très performantes, mais sur un ensemble de métriques de mauvaise qualité – une approche pour le moins surprenante, mais dont les fondements sont bien éprouvés.

Outre l'intérêt d'une approche très différente, le Boosting est un cadre beaucoup moins contraignant que celui des machines à noyaux. En effet, ce cadre repose sur un ensemble de métriques de mauvaise qualité – autrement dit des métriques bien plus faciles à concevoir que des métriques très performantes ! Le Boosting est une technique qui a déjà fait ses preuves en apprentissage par lots, par exemple pour la détection de visage. Cependant, pour la recherche interactive, cette approche est encore peu explorée. La principale motivation derrière cette étude est donc la perspective d'une approche capable d'offrir la même qualité de résultat que le couple SVM/Noyau pour la recherche interactive, mais avec un coût de conception très inférieur.

Ce travail fait partie de la thèse d'Alexis Lechervy, dont le but final est la conception de techniques interactives et collaboratives pour la recherche dans les bases vidéos. Les résultats que nous proposons sont une première étape pour atteindre ce but [RFIA10,ICPR10a].

4.1 Boosting

4.1.1 Principe

Le principe du Boosting a été proposé par Freund en 1995, puis a été popularisé par la méthode AdaBoost (*Adaptive Boosting*) l'année suivante [19]. Cette approche repose sur un ensemble $\mathcal{H}$ de classifieurs h_t dit *faibles*, dont les performances sont supposées meilleures que le hasard. L'objectif du Boosting est alors la formation d'une combinaison linéaire de ces classifieurs faibles, appelée classifieur fort :

$$H_T = \sum_{t=1}^{T} \alpha_t h_t \tag{4.1}$$

Cette combinaison est formée de manière itérative, en ajoutant à chaque itération un nouveau terme à la somme :

$$H_{T+1} = H_T + \alpha_{T+1} h_{T+1} \tag{4.2}$$

avec

$$\alpha_{T+1}, h_{T+1} = \underset{\alpha>0, h\in\mathcal{H}}{\operatorname{argmax}} C(H_T + \alpha h) \tag{4.3}$$

La fonction $C(.)$ est le critère réel (i.e. non empirique) à maximiser qui dépend du problème à résoudre. Naturellement, en pratique on approche ce critère sur la base des exemples fournis, généralement en effectuant une minoration. Le plus courant est d'utiliser un critère de classification binaire, mais d'autres choix peuvent être faits, comme dans le cas du chapitre précédent, où nous avons utilisé l'alignement du noyau.

Avant de poursuive plus avant dans la formation de la combinaison, notons bien que cette approche est une approche de changement d'espace de représentation. En effet, l'hyperplan H_T évolue dans un espace vectoriel, où les coordonnées de chaque vecteur sont les valeurs des classifieurs faibles, i.e. la méthode induit un espace par une fonction ϕ définie par $\phi(\mathbf{x}) = (h_1(\mathbf{x}), \ldots, h_T(\mathbf{x}))$. De plus, cette transformation se fait de manière supervisée, étant donné que le critère C est lié aux exemples d'apprentissage. En d'autres termes, cette approche permet d'apprendre l'hyperplan séparateur ***et*** l'espace de représentation.

Toute la réussite de l'approche réside dans la capacité à trouver une solution à l'équation (4.3), autrement dit à trouver un classifieur faible qui, une fois ajouté à H_T, améliore le critère C. Pour ce faire, on considère généralement une famille de classifieurs faibles, et un "apprenant faible" (*weak learner*) sélectionne au sein de cette famille les classifieurs faibles qui améliorent le critère C. Par exemple, la famille peut être celle des hyperplans séparateurs dans l'espace des descripteurs visuels, et l'"apprenant faible" détermine celui qui maximise C. De plus, on introduit souvent une notion de pondération fonction de la bonne ou mauvaise classification des exemples pour aider l'"apprenant faible" dans sa recherche.

4.1.2 Les variantes du Boosting

Suite au succès d'AdaBoost [19], de nombreuses variantes sont apparues au sein de la littérature. On peut noter les améliorations de ce premier algorithme, comme Real AdaBoost [74], Gentle AdaBoost [21], ou Modest AdaBoost [85]. Puis, d'autres propositions étendent cette approche vers d'autres problèmes que la classification binaire, comme le classement avec RankBoost [20], les arbres de décision avec Reweight Boost [71], les fonctions noyaux [9, 31], mais aussi d'une manière générique dès lors que le problème peut être exprimé sous la forme d'une descente de gradient [56].

Des propositions ont aussi été faites dans un contexte proche de la recherche interactive. Les premières méthodes que l'on peut rattacher à ce contexte sont les méthodes qui fonctionnent avec peu d'exemples, comme [12, 92]. On peut noter tout particulièrement celles qui s'appuient sur l'apprentissage actif. Par exemple, Li [48] propose de raisonner dans l'espace des versions pour alimenter petit à petit AdaBoost en exemples, ce qui revient à sélectionner les exemples les plus proches de la frontière. Collins [7] utilise aussi ce principe d'incertitude pour sélectionner les documents à faire annoter, mais intègre aussi une idée intéressante qui consiste à réduire la famille de classifieurs faibles à celle engendrée par les annotations. Notons que des propositions permettent d'intégrer l'utilisateur directement dans le processus itératif du Boosting [50]. Des méthodes ont été proposées pour le suivi d'objets dans les vidéos [26, 82], un contexte qui n'est pas totalement étranger de la recherche interactive. Dans tous les cas, on retrouve les idées issues du boosting en ligne [64], qui mettent à jour le classifieur fort en fonction d'une nouvelle annotation. On retrouve aussi l'idée des familles de classifieurs faibles dynamiques, qui évoluent en fonction du flux d'exemples.

Enfin, on trouve un très grand nombre de méthodes basées sur le Boosting et ses variantes, mais qui diffèrent par le choix et le paramétrage des classifeurs faibles utilisés.

4.2 Méthode proposée

4.2.1 Schéma général

La méthode que nous proposons repose sur un ensemble $\mathcal{H}$ de classifieurs faibles, qui évolue en fonction des itérations du bouclage de pertinence. Pour former cet ensemble $\mathcal{H}$, nous nous appuyons sur les informations visuelles $\mathbf{p} \in P_i$ présentes dans les images i annotées positivement. Sur la base de ces informations visuelles, nous fabriquons des classifieurs faibles $h_{\mathbf{p}}$:

$$\mathcal{H} = \{h_{\mathbf{p}} | \mathbf{p} \in P_i \text{ et } y_i > 0\} \tag{4.4}$$

Nous verrons dans la section suivante comment produire ces classifieurs faibles. Cependant, dans tous les cas, l'idée est de fabriquer des classifieurs faibles qui reposent sur des propriétés visuelles propres à l'image utilisée pour les fabriquer. Par exemple, si l'image utilisée est un paysage avec un ciel bleu, un classifieur faible peut être de regarder si il y a la couleur bleu dans la partie supérieure de l'image.

Ce principe est essentiel pour gérer le faible nombre d'annotations. En effet, en ne considérant que des classifieurs faibles qui reposent sur des éléments visuels présents dans les

images annotées, on s'assure que ces classifieurs faibles auront une réponse pertinente sur ces éléments annotés. Dans le cas contraire, on aurait des réponses de classifieurs faibles aléatoires, et par conséquent une sélection lors des itérations du boosting tout aussi aléatoire.

Une fois muni d'un ensemble de classifieurs faibles, on utilise l'une des variantes du boosting pour fabriquer le classifieur fort final. Pour ce faire, nous avons opté pour la méthode RankBoost, qui permet d'apprendre le classement d'un ensemble de données [20]. Dans ce cas, le classifieur fort est un opérateur qui trie la base de données, une opération plus fine que la classification binaire qui distribue les données dans deux classes. Cette méthode est bien plus adaptée à notre contexte étant donné que l'utilisateur est intéressé par le classement des documents. Elle offre en outre des résultats bien meilleurs que les méthodes de boosting plus habituelles comme AdaBoost.

4.2.2 Apprentissage actif

Nous proposons de sélectionner les images que l'utilisateur va annoter dans le but d'améliorer l'ensemble des classifieurs faibles. Pour ce faire, nous commençons par présélectionner les documents dont la classification est la plus incertaine. Puis, au sein de cette présélection, nous choisissons les images qui possèdent les classifieurs faibles les plus intéressants à ajouter dans l'ensemble $\mathcal{H}$. Ce choix est effectué en utilisant le critère de sélection de classifieur faible du boosting, noté ici $r_0(.)$:

$$a(P_i) = \underset{\mathbf{p} \in P_i}{\operatorname{argmax}}\, r_0(h_{\mathbf{p}}) \tag{4.5}$$

En procédant ainsi, nous ajoutons des classifieurs faibles qui permettent de produire un meilleur classifieur fort compte tenu des annotations courantes.

4.2.3 Les classifieurs faibles

Nous proposons deux types de classifieurs faibles. Dans les deux cas, nous considérons un découpage en 9 blocs de chaque image (cf. Fig. 4.1). Chaque bloc l d'une image i est représenté par un histogramme de propriétés visuelles (couleurs, textures, ...) noté $\mathbf{z}_{li}$.

Classifieurs faibles de type 1.

Pour ce premier type de classifieur faible, chaque image est représentée par l'ensemble des histogrammes calculés sur les blocs :

$$Z_i = \{\mathbf{z}_{li}\}_l \tag{4.6}$$

On définit un classifieur faible $h^1_{\hat{z}}$ pour chaque bloc $\hat{\mathbf{z}}$ d'une image. L'ensemble P^1_j des classifieurs faibles générés par une image j est donc défini comme suit :

$$P^1_j = \{h^1_{\hat{\mathbf{z}}} | \hat{\mathbf{z}} \in Z_j\} \tag{4.7}$$

Chaque image génère ainsi 9 classifieurs faibles.

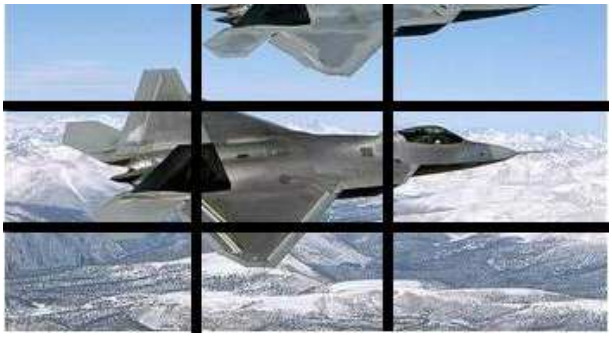


FIGURE 4.1 – Classifieurs faibles – découpage en 9 blocs.

Chaque classifieur faible $h^1_{\hat{\mathbf{z}}}$ évalue une image représentée par Z_i en fonction de la plus petite distance à l'histogramme $\hat{\mathbf{z}}$ associé au classifieur faible :

$$h^1_{\hat{\mathbf{z}}}(Z_i) = 1 - \min_{\mathbf{z}_{li} \in Z_i} d(\mathbf{z}_{li}, \hat{\mathbf{z}}) \tag{4.8}$$

avec $d(.,.)$ la distance du χ^1.

Classifieurs faibles de type 2.

Pour ce second type de classifieurs faibles, nous considérons les combinaisons C_r de blocs dans l'image. Une combinaison de blocs est, par exemple, $\{1, 4, 7\}$, autrement dit tous les blocs sur la partie gauche d'une image. Nous considérons toutes les combinaisons possibles, sauf celle qui n'inclue aucun bloc. Puis, nous décrivons chaque combinaison de blocs d'une image i à l'aide de la somme des histogrammes des blocs associés :

$$\boldsymbol{\rho}_{ri} = \sum_{l \in C_r} \mathbf{z}_{li} \tag{4.9}$$

Chaque image est représentée par l'ensemble de ces histogrammes :

$$R_i = \{\boldsymbol{\rho}_{ri}\}_r \tag{4.10}$$

On définit un classifieur faible $h^2_{\hat{\boldsymbol{\rho}},r}$ pour chaque combinaison de blocs C_r d'une image. L'ensemble P^2_j des classifieurs faibles générés par une image j est donc défini comme suit :

$$P^2_j = \{h^2_{\hat{\boldsymbol{\rho}},r} | \hat{\boldsymbol{\rho}} = \boldsymbol{\rho}_{rj} \in R_j\}_r \tag{4.11}$$

Chaque image génère ainsi 511 classifieurs faibles pour un découpage en 9 blocs.

Chaque classifieur faible $h^2_{\hat{\boldsymbol{\rho}},r}$ évalue une image représentée par R_i en fonction de la distance entre l'histogramme $\hat{\boldsymbol{\rho}}$ associé au classifieur faible et l'histogramme dans l'image i issu de la même combinaison de blocs :

$$h^2_{\hat{\boldsymbol{\rho}},r}(R_i) = 1 - d(\boldsymbol{\rho}_{ri}, \hat{\boldsymbol{\rho}}) \tag{4.12}$$

avec $d(.,.)$ la distance du χ^1.

Dictionnaire	32 mots		64 mots	
Méthode	SVM	Boosting	SVM	Boosting
bicycle	34	44	43	38
bus	40	45	48	46
car	57	68	65	68
cat	26	30	28	28
cow	32	40	47	37
dog	20	25	22	24
horse	16	19	22	19
motorbike	35	42	46	41
person	34	38	33	38
sheep	46	53	48	44
all	35	40	40	38

TABLE 4.1 – Précision Moyenne(%) sur l'ensemble de test de VOC2006. Initialisation avec 1 image, 10 bouclages, 5 annotations par bouclage.

4.3 Expériences

Nous avons mené des expériences pour comparer l'approche par boosting interactif proposé à une approche avec des SVMs. Ces expériences ont été menées sur la base VOC 2006 (cf section A.2.2) en suivant le protocole de recherche interactive présenté en section A.1.1. Dans tous les cas, les descripteurs visuels sont des couleurs $L^\star a^\star b^\star$ (lab) et des textures par ondelettes quaternioniques (qw).

Nous avons testé les méthodes avec deux dictionnaires, l'un avec 32 mots par type de descripteur visuel, et l'autre avec 64 mots. Puis nous avons utilisé l'un de ces dictionnaires pour calculer les histogrammes dans les images. Pour les SVMs, nous avons calculé un seul histogramme par image, et avons utilisé un noyau gaussien avec une distance du χ^1. Les résultats sont présentés en Figure 4.1.

Comme nous pouvons le constater, les résultats sont tantôt meilleurs pour les SVMs, tantôt meilleurs pour le boosting interactif. Il nous est ainsi difficile d'affirmer si l'une ou l'autre méthode est véritablement meilleure en terme de performances. En tout état de cause, ces résultats sont très encourageants étant donné que ce nouveau cadre d'apprentissage par boosting interactif est capable de rivaliser avec une méthode bien rodée. Ainsi, nous pensons qu' un travail plus approfondi sur les classifieurs permettra d'obtenir de bien meilleurs résultats.

Notons qu'avec des techniques de boosting classiques, non adaptées au contexte interactif, les résultats sont très mauvais, voire totalement aléatoires. Par exemple, le simple fait de travailler avec un ensemble statique de classifieurs faibles produit des classifieurs forts de très mauvaise qualité.

4.4 Conclusion et Perspectives

Nous avons présenté dans ce chapitre une approche par boosting pour effectuer des recherches interactives dans les bases d'images. Pour ce faire, nous nous appuyons sur la notion d'ensemble dynamique de classifieurs faibles, ce qui nous permet de gérer le faible nombre d'annotations. Le cadre peu restreint du boosting nous permet aussi de concevoir facilement des classifieurs assez complexes, contrairement à l'approche par noyau où cette étape est plus difficile.

Compte tenu de la très grande quantité de données qu'il faut gérer avec les vidéos, nous avons jugé préférable de commencer par concevoir et valider les algorithmes sur des bases d'images. Dans le contexte de la thèse de A. Lechervy, l'étape suivante est le passage de ces techniques interactives à des techniques collaboratives. L'objectif dans ce cas est de mutualiser les recherches d'un ensemble d'utilisateurs dans le but de former plus rapidement les classifieurs finaux. Une fois ce travail fait, nous pourrons travailler directement sur les vidéos. Ce passage devrait se faire assez facilement grâce à l'approche par boosting qui permet une conception avec peu de contraintes.

Conclusion et perspectives

Conclusion et perspectives

Recherches basées sur les mises en correspondance

L'un des thèmes abordés lors de nos travaux est celui autour des techniques basées sur les mises en correspondance de descripteurs visuels. Ces techniques ont été présentées pour résoudre la recherche interactive d'objet (parfois appelée recherche d'instance), mais sont aussi utilisées pour la recherche de copie (image et ses déformations), et dans une certaine mesure pour la recherche d'une classe d'objet – tous problèmes où la notion de mise en correspondance est pertinente.

Nous avons présenté au cours de la première partie de ce mémoire des méthodes dans ce contexte basées sur les fonctions noyaux, ce qui nous a permis d'utiliser des techniques avancées d'apprentissage, comme les SVMs ou l'apprentissage actif. Il reste encore des améliorations possibles, notamment si l'on souhaite produire des solutions qui généralisent davantage. En effet, les solutions actuelles ont un comportement plutôt binaire lors de la prise de décision : le document à classer est ou n'est pas considéré comme pertinent, il y a peu de niveaux intermédiaires entre ces deux extrêmes. Cet objectif représente un véritable défi, et nous pensons qu'il sera résolu à long terme.

Un des reproches souvent fait aux méthodes par mise en correspondance est leur coût en calcul. Ainsi, un des enjeux actuels autour de cette thématique est de trouver le juste compromis pour approcher ces techniques, tout en conservant au mieux les performances. Pour ce faire, une approche que nous étudions est la *linéarisation* des processus de mise en correspondance. Ainsi, si chaque document est représenté par un vecteur, et que le produit scalaire entre deux représentations approche bien le processus de mise en correspondance, on obtient un gain considérable en temps de calcul, même si les vecteurs sont de très grande dimension. Par exemple, on peut voir les représentations VLAD [34] comme une approximation d'un noyau sur sacs. Pour linéariser ces calculs, une piste est celle de l'approximation en séries de Taylor des fonctions noyaux sur sacs. En suivant cette piste, on déplie littéralement les calculs de mise en correspondance, ce qui conduit à des représentations vectorielles extrêmement grandes (cela peut monter à des millions de dimensions). Cependant, ces représentations sont très creuses et/ou redondantes. Un travail de post-traitement nous permettra certainement de trouver des réductions pertinentes.

Représentations dynamiques

Un autre thème abordé lors de nos travaux est celui des représentations dynamiques, et plus précisément les représentations des documents fonctions de la requête de l'utilisateur. Dans ce contexte, nous avons présenté deux approches, l'une basée sur les noyaux, et l'autre sur le boosting. Dans les deux cas, nous avons montré qu'une telle stratégie permet d'améliorer la qualité des résultats renvoyés par le système. Il existe encore du travail en ce sens, comme nous l'avons présenté dans les chapitres concernés.

Cependant, un point que nous avons peu évoqué est l'intérêt des représentations dynamiques sur le plan de la complexité en calculs. En effet, ces approches permettent de se concentrer uniquement sur les propriétés visuelles pertinentes pour la requête, ce qui signifie qu'il faut considérer beaucoup moins de propriétés visuelles pour classer les documents. Par exemple, pour le cas des dictionnaires de couleurs et textures, une trentaine de mots visuels sont généralement suffisants avec une approche dynamique, alors qu'il en faut plusieurs centaines avec un dictionnaire statique. Il existe donc un fort potentiel ici, que nous avons commencé à explorer avec les façades de Paris, dans le cadre de la thèse de Jean-Emmanuel Haugeard. Il reste beaucoup de choses à découvrir, par exemple nous avons commencé à étudier la possibilité de combiner des index rapides issus de la communauté Data Mining dans ce contexte. Nous étudions aussi cette approche pour les vidéos, dans le cadre de la thèse d'Alexis Lechervy, dans le but de produire des solutions capables de traiter la très grande quantité de données produites par les flux vidéos.

Apprentissage collaboratif

L'apprentissage collaboratif est une approche originale pour entraîner et optimiser un système intelligent. Cette approche est similaire à celle d'un apprentissage supervisé, avec la différence que l'on considère un ensemble d'utilisateurs qui vont travailler en parallèle sur un problème, puis se consulter pour mutualiser leurs trouvailles. Un bon exemple d'apprentissage collaboratif est la recherche long terme dans une base d'image, une problématique sur laquelle nous avons travaillé [PR06,CVIU08]. L'objectif dans ce cas est d'exploiter les recherches menées par un ensemble d'utilisateur pour optimiser le système, de sorte que tout nouvel utilisateur retrouve beaucoup plus rapidement une catégorie précédemment recherchée. Un autre exemple d'apprentissage collaboratif est le filtrage collaboratif, qui est utilisé par les magasins de DVDs pour conseiller leur clients.

Ce thème de recherche est très intéressant car il correspond bien à des problèmes d'applications réelles. En effet, on dispose rarement dans la réalité d'un ensemble d'apprentissage précis et sans erreur, mais plutôt d'informations imprécises et de mauvaise qualité. Le meilleur exemple est celui des titres ou des commentaires que donnent les utilisateurs sur les dépôts publics de vidéos comme youtube ou dailymotion. Ces informations sont de mauvaise qualité, mais cependant leur très grand nombre permet de le compenser. Or pour traiter des quantités considérables d'information, une approche habituelle n'est pas pertinente compte tenu des machines actuelles. Paralléliser les traitements est donc la meilleure solution, et cela est d'autant plus vrai que l'architecture des ordinateurs ne cesse d'aller dans ce sens. Plus que simplement

paralléliser, nous proposons de concevoir tout un modèle collaboratif pour exploiter au mieux la distribution des traitements, autant sur le plan calculatoire que sur le plan de la résolution du problème d'apprentissage lui-même.

Apprentissage dans le flux

Une thématique sur laquelle nous comptons travailler à long terme est celle de l'apprentissage dans le flux. Cette thématique est proche des précédentes, sauf que nous considérons une forte dynamique dans l'information supervisée. Par exemple, il est courant de considérer que les objets ou catégories recherchés dans une base sont toujours les mêmes. Dans un scénario dynamique, les objets ou catégories qui intéressent les utilisateurs peuvent changer du tout au tout. Dans ce cas, le système doit être capable de se réadapter à ces nouveaux centres d'intérêt. Le meilleur exemple est celui des fils de discussions instantanés comme tweeter. Ce type d'application produit un flux constant de micro informations produites en très grand nombre. Ces informations ne sont pas indépendantes, et il existe des thèmes d'intérêt qui émergent (les "buzz"). Ces thèmes évoluent très rapidement, d'un jour à l'autre, voire en l'espace d'une heure.

Ce thème est difficile car il réunit de nombreux problèmes. En effet, nous avons à la fois un apprentissage dynamique, des informations supervisées de mauvaise qualité, et un très grand nombre de données ! En d'autres termes, ce thème est la synthèse de toutes les problématiques que nous avons abordé ou que nous comptons aborder. Concevoir un système capable de résoudre efficacement ce problème constitue ainsi un véritable défi que nous comptons relever.

Annexes

Annexe A

Bases et Protocoles

A.1 Critères et Protocoles

A.1.1 Protocole de simulation de recherches interactives

Afin d'évaluer la qualité d'une méthode de recherche interactive, nous simulons un grand nombre de sessions de recherche. Le processus est le même qu'en situation réelle, sauf que l'utilisateur est remplacé par un robot qui annote les images en utilisant la vérité terrain.

Protocole 1 Protocole d'évaluation pour une catégorie

Paramètres :
- Catégorie C
- Ensemble d'apprentissage A, ensemble de test T,
- Nombre de bouclages par session F, Nombre d'annotations par bouclage s

Pour chaque image c de $C \cap A$ faire
- Démarrer une nouvelle session de recherche
- Initialisation :
  - Annoter c positivement
  - $I_0^\star$ = s images les plus proches de c dans A
  - Annoter positivement les images de $I_0^\star \cap C$
  - Annoter négativement les images de $I_0^\star \cap \bar{C}$
- Pour f de 1 à F faire
  - Entraîner un nouveau classifieur avec les annotations courantes
  - Trier les images de T en utilisant la fonction de classification
  - Calculer la courbe de Précision/Rappel $pr(f, c)$ sur T
  - $I_f^\star$ = s images sélectionnées dans A avec l'apprentissage actif
  - Annoter les images de $I_f^\star$

Sortie : Courbe de Précision/Rappel $PR(f) = \underset{c}{\text{moyenne}} \Big(pr(f, c) \Big)$

A.1.2 Précision Moyenne (AP) et Moyenne des Précision Moyennes (MAP)

Ce protocole produit une courbe de précision/rappel pour une catégorie C. Dans le but de produire des résultats plus synthétiques, on considère la Précision Moyenne (*Average Precision*) définie par l'intégrale de la courbe de précision/rappel. Enfin, dans le cas où l'on souhaite évaluer un système ou une méthode à l'aide d'un seul critère, on calcule la moyenne des Précisions Moyennes (*Mean Average Precision*) pour toutes les catégories de la base de données.

A.2 Bases

A.2.1 Base Objets sur Fonds Aléatoires

Dans le but de valider les algorithmes pour la recherche d'objets, nous avons conçu une petite base de 50 objets sur fonds aléatoires. Les images des objets sont extraites de la base COIL-100 de l'université de Columbia [1], et placées sur des images de paysages prises aléatoirement dans la base de l'université de Washington [2]. Nous avons conservé 12 vues de chaque objet, ce qui donne un total de 600 images.

FIGURE A.1 – Objets sur fond aléatoires

L'objectif de cette base est avant tout de disposer d'un cas d'école pour tester les méthodes. En effet, les objets sont clairement identifiés et les descripteurs visuels actuels sont suffisamment robustes pour faire l'hypothèse qu'il existe toujours des mises en correspondances visuelles. De plus, nous avons volontairement laissé des objets différents sur exactement les mêmes images de fond, dans le but de disposer de fausses mises en correspondance.

1. http ://www.cs.columbia.edu/CAVE/software/softlib/coil-100.php
2. http ://www.cs.washington.edu/research/imagedatabase/

A.2.2 Visual Object Challenge 2006 (VOC2006)

La base Visual Object Challenge 2006 (VOC2006) est une base créée par le réseau d'excellence PASCAL, dans le but de comparer les différentes techniques des chercheurs.

La base est constituée de 5304 images réparties en 3 sous-ensembles. Les deux premiers ensembles "train" (1277 images) et "val" (1341 images) servent pour l'entraînement des méthodes. Le dernier ensemble "test" (2686 images) sert pour l'évaluation des méthodes. La base est aussi munie d'une vérité terrain sous la forme de 10 catégories, dont des exemples sont présentés en Figure A.2.

D'une manière générale, les images sont des photographies centrées sur un objet, une personne ou un animal.

A.2.3 Visual Object Challenge 2007 (VOC2007)

La base Visual Object Challenge 2007 (VOC2007) est une base similaire à VOC2006, mais plus grande et plus difficile.

La base est constituée de 9963 images réparties en 3 sous-ensembles. Les deux premiers ensembles "train" (2501 images) et "val" (2510 images) servent pour l'entraînement des méthodes. Le dernier ensemble "test" (4952 images) sert pour l'évaluation des méthodes. La base est aussi munie d'une vérité terrain sous la forme de 20 catégories, dont des exemples sont présentés en Figure A.3.

Contrairement à VOC2006, les objets d'intérêt de cette base sont souvent mal centrés ou mal cadrés, et souvent présents dans l'arrière plan.

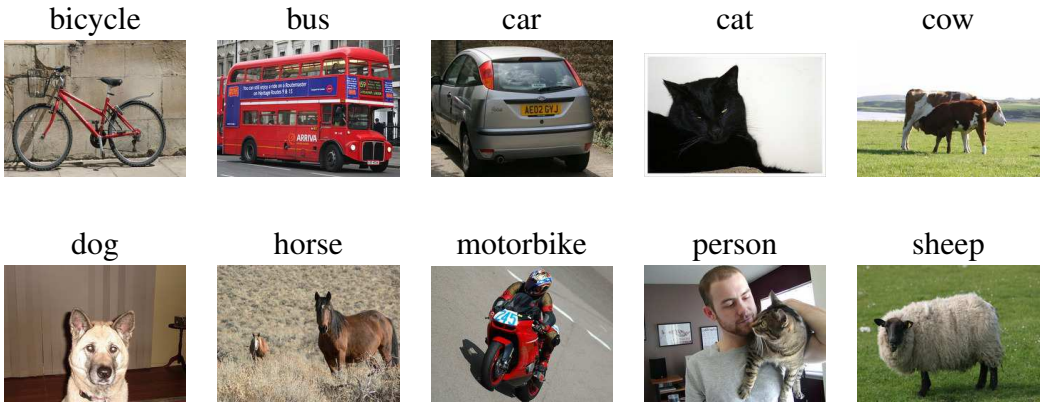


FIGURE A.2 – Images de la base VOC2006.

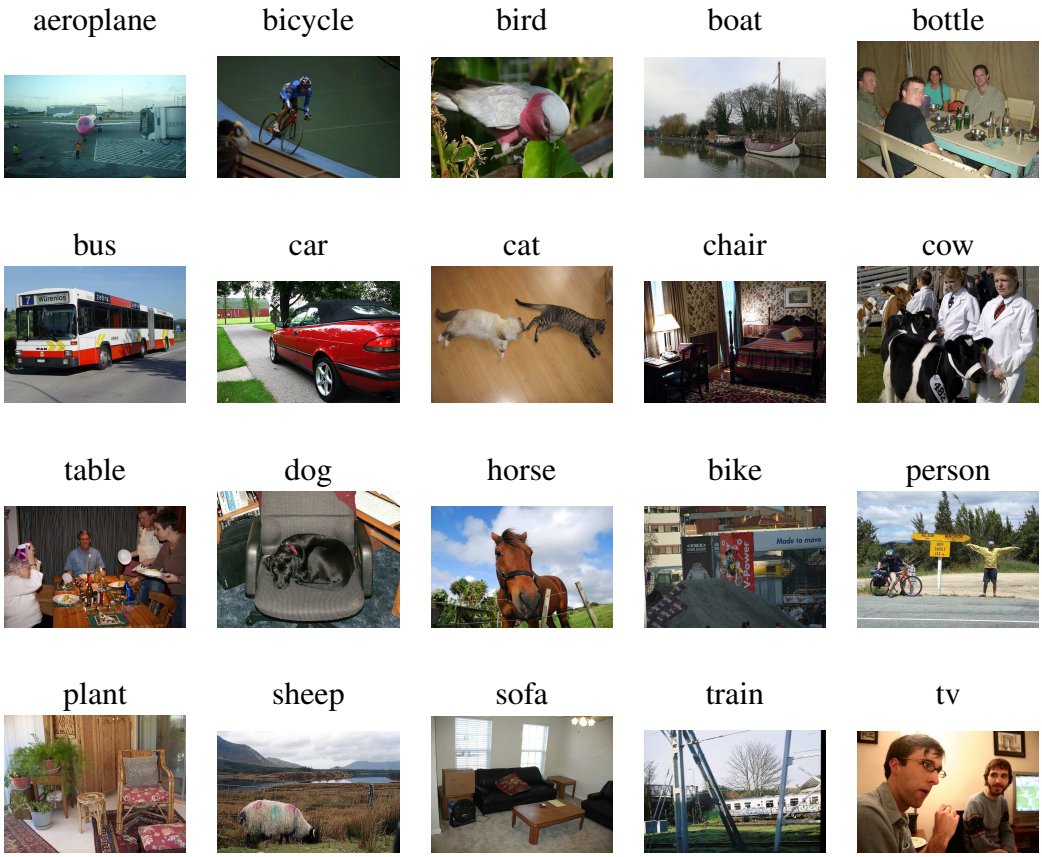


FIGURE A.3 – Images de la base VOC2007.

Annexe B

Curriculum Vitæ

Philippe-Henri GOSSELIN
né le 25/12/1979 à Chambray les Tours
Union libre

Maître de Conférences à l'ENSEA

ETIS / ENSEA - UCP - CNRS UMR 8051
6 ave du Ponceau, 95000 Cergy-Pontoise, France
téléphone : 01 30 73 62 98
fax : 01 30 73 66 27
mail : gosselin@ensea.fr

B.1 Résumé

Curriculum Vitæ	Philippe-Henri Gosselin, 31 ans, Union libre Actuellement Maître de Conférences à l'ENSEA Equipe « *Indexation Multimédia et Intégration de Données* » Cursus académique et professionnel 2002 DEA de l'Université de Cergy-Pontoise « *Traitement de l'Image et du Signal* » 2005 Thèse de Doctorat de l'Université de Cergy-Pontoise « *Méthodes d'apprentissage pour la recherche de catégorie d'images* » 2005–06 Post-Doctorat au laboratoire ETIS (UMR 8051) « *Fonctions noyaux pour la recherche de classes d'objets* » 2006–07 Post-Doctorat au laboratoire LIP6 (UMR 7606) « *Intégration des relations spatiales dans les fonctions noyaux pour la recherche d'objets* »
Enseignement	Total d'environ 926h équivalent TD, hors encadrements. ENSEA Ecole Nationale Supérieure d'Electronique et de ses Applications *Langages de programmation, structures de données, algorithmique,... (708h)* ISEA Formation Continue *Langage Java et SQL (168h)* Master Systèmes Intelligents et Communicants *Indexation Multimédia et Apprentissage statistique (50h)*
Encadrement	Thèses 2 soutenues, 3 en cours Master 5 mini-projets, 8 stages Ingénieur 13 mini-projets
Recherche	Apprentissage statistique autour des bases de données multimédia *Analyse, Clustering, Indexation, Classification, Interactif, Collaboratif, ...* Publications 8 Revues internationales avec comité de lecture *Trans. on Image Processing, Pattern Recognition, Computer Vision and Image Understanding, ...* 17 Conférences internationales avec comité de lecture *IEEE Int. Conf. on Image Processing, IEEE Int. Conf. on Pattern Recognition, ACM Int. Conf. on Image and Video Retrieval, ...* 5 Conférences nationales avec comité de lecture *Reconnaissance des Formes et Intelligence Artificielle, ...* 1 Chapitre de livre *Machine Learning Techniques for Multimedia, Springer 2008*

B.2 Cursus académique et professionnel

B.2.1 Cursus

2002	DEA de l'Université de Cergy-Pontoise « *Traitement de l'Image et du Signal* »
2005	Thèse de Doctorat de l'Université de Cergy-Pontoise « *Méthode d'apprentissage pour la recherche de catégorie d'images* »
2005-06	Post-Doctorat au laboratoire ETIS (UMR 8051) « *Fonctions noyaux pour la recherche de classes d'objets* »
2006-07	Post-Doctorat au laboratoire LIP6 (UMR 7606) « *Intégration des relations spatiales dans les fonctions noyaux pour la recherche d'objets* »
depuis 2007	Maître de Conférences à l'ENSEA

B.2.2 Parcours dans la recherche

Stage DEA « Quaternions et couleurs »

Ma première expérience de la recherche s'est effectuée lors de mon stage DEA au laboratoire IRCOM-SIC à Poitiers, sous la direction de Philippe Carré, d'avril à août 2002.

L'objet de ce stage était l'étude du formalisme des quaternions pour le traitement des couleurs dans les images. Pour rappel, l'ensemble des quaternions est une extension de l'ensemble des complexes qui permet de formaliser simplement de nombreuses transformation des espaces en trois dimensions. Ils sont beaucoup utilisés en image de synthèse pour représenter les transformations des objets (translations, rotations, etc.). Dans le cadre du stage, je les ai utilisés pour formaliser les traitements des vecteurs couleurs (RGB, HSV, etc.), comme la convolution ou la corrélation.

Thèse « Méthode d'apprentissage pour la recherche de catégorie d'images »

Président - Jean-Pierre Cocquerez (PR, Heudiasyc)
Rapporteur - Patrick Gallinari (DR, LIP6)
Rapporteur - Frédéric Jurie (CR, LEAR)
Directrice de thèse - Sylvie Philipp-Foliguet (PR, ETIS)
Co-Directeur de thèse - Mathieu Cord (MCF, ETIS)

Encadré par Matthieu Cord et Sylvie Philipp-Foliguet au laboratoire ETIS à Cergy-Pontoise, mon sujet de thèse portait sur les méthodes d'apprentissage pour la recherche d'images par le contenu. Pour parvenir à un système de recherche complet, un ensemble d'outils est nécessaire.

La première étape de tout système de recherche réside dans le calcul de signatures visuelles. Dans ce cadre, j'ai étudié différents espaces couleurs et différentes techniques d'extraction de texture basées sur les filtres de Gabor. Puis, j'ai proposé une méthode pour calculer rapidement et efficacement des dictionnaires visuels sur de grandes bases d'images. Suite à ce processus,

chaque image est représentée par des histogrammes formés par le nombre d'occurrences de mots visuels.

Suite à cette première étape, il est nécessaire de choisir une métrique pour pouvoir comparer les signatures. Pour ce faire, j'ai opté pour le formalisme des fonctions noyaux, dans le but de pouvoir utiliser les signatures avec les classifieurs issus de l'apprentissage statistique. Ce choix a permis de bénéficier immédiatement des performances de ces outils éprouvés par une théorie riche en résultats. Ce formalisme offre aussi bien d'autres avantages sur lesquels je me suis appuyé pour proposer de nouvelles techniques d'analyse et de recherche.

Une fois muni de signatures et d'une métrique, il devient possible d'effectuer la recherche proprement dite. Il existe un certain nombre de scénarios de recherche, comme la recherche de copie ou de cible. Dans ma thèse, je me suis intéressé à la recherche interactive de catégorie d'images, une catégorie étant un ensemble d'images avec une composante sémantique commune (même objet, même ambiance, même lieu, ...). Pour ce faire, j'ai étudié les différentes techniques de classification compatibles avec les fonctions noyaux, et dans certains cas j'ai rendu compatibles des méthodes existantes. Pour gérer l'aspect interactif, j'ai opté pour l'apprentissage actif, dont l'objectif est la sélection des images les plus intéressantes à faire annoter.

Enfin, j'ai aussi étudié lors de ma thèse le problème de l'apprentissage long terme, dont l'objet est l'optimisation d'un système de recherche en fonction de son utilisation. Deux méthodes ont été proposées dans ce cadre.

Publications liées aux travaux de thèse : 5 journaux [JASP06, PR06, IVC07, CVIU08, IP08], 9 conférences internationales [CVDB04, ACIVS04, ICIP04, MCBAR04, MLMM05, ICIP05, ICME06, ICIP06a, ICIP06b], 2 conférences nationales [RFIA04, CAP05], 1 chapitre de livre [MLTM08].

Post-Doctorat « Fonctions noyaux pour la recherche de classes d'objets »

Mon premier post-doctorat s'est déroulé au laboratoire ETIS, sous la direction de Matthieu Cord et Sylvie Philipp-Foliguet, de décembre 2005 à novembre 2006. L'objectif était l'étude des fonctions noyaux sur les ensembles de descripteurs visuels (sacs de descripteurs). L'idée derrière cet objectif était de reproduire les systèmes de mise en correspondance de descripteurs visuels locaux, mais en assurant la compatibilité avec le cadre des fonctions noyaux. Pour ce faire, j'ai étudié les fonctions de la littérature, mais aussi proposé des fonctions innovantes.

Post-Doctorat « Intégration des relations spatiales dans les fonctions noyaux »

Mon deuxième post-doctorat a démarré au laboratoire LIP6 en janvier 2007, sous la direction de Matthieu Cord et Sylvie Philipp-Foliguet. Il s'est ensuite poursuivi au laboratoire ETIS pour des raisons de financement jusqu'en août 2007. L'objectif était dans la continuité du précédent, et avait pour but l'intégration des relations spatiales dans les fonctions noyaux sur descripteurs visuels locaux.

Publications liées aux travaux de post-doctorat : 3 conférences internationales [CIVR07a, CIVR07b, ICIP07].

Maître de Conférences

J'ai été recruté en septembre 2007 au sein de l'équipe MIDI du laboratoire ETIS à Cergy Pontoise, équipe dirigée par Sylvie Philipp-Foliguet.

Ma première mission au sein de cette équipe a été de participer aux projets engagés autour des méthodes de recherche multimédia par Graphes (images et objets en trois dimensions). Dans ce contexte, les documents sont représentés par des Graphes Relationnels Valués. Dans le but de les utiliser dans des classifieurs, j'ai contribué à la conception de fonction noyaux sur de telles structures, ainsi qu'à leurs applications (projets ANR EROS3D et iTowns).

Parallèlement à ces travaux, je travaille sur différentes techniques autour de l'apprentissage en multimédia, dont l'apprentissage faiblement supervisé de représentation des images (par fonctions noyaux et par boosting), ainsi que l'apprentissage collaboratif pour la classification par lot et la recherche long terme dans les bases d'images.

Publications liées aux travaux de maître de conférence : 3 journaux [CVIU09, PR11, IVC11] (+3 soumis [PRL, GRS, CVIU]), 5 conférences internationales [ICPR08, ACM10, ICPR10a, ICPR10b, ICIP11], 2 conférences nationales [EGC08, RFIA10].

B.3 Activités d'enseignement et d'encadrement

B.3.1 Enseignements

Depuis la rentrée 2007, j'effectue un service complet réparti entre l'Ecole Nationale Supérieure de l'Electronique et de ses Applications (ENSEA) à Cergy-Pontoise et le Master Recherche Systèmes Intelligents et Communicants (SIC) de l'Université de Cergy-Pontoise. Les disciplines que j'ai enseignées sont l'informatique (programmation, algorithmique), le traitement des images et l'apprentissage pour le multimédia.

ENSEA. En première année, j'enseigne le langage C (module de 26h de TD). En deuxième année, j'enseigne le langage Java (module de 6h de CM et 12h de TD). J'ai aussi initié une option sur l'indexation dans les bases relationnelles et multimédia, dont j'assure la majeure partie des cours et travaux dirigés (module de 12h de CM et 16h de TP). Cette option permet d'introduire aux étudiants les techniques classiques d'indexation des données (arbres, hachage, etc.) qu'ils mettent en pratique lors des TPs pour indexer les pages de Wikipédia. Enfin, pour la troisième année, je suis responsable et principal acteur du module d'algorithmique (12h de cours, 6h de TD et 20h de TP). Ce module présente les méthodologies utilisées en informatique pour permette la résolution rapide des problèmes dans les espaces d'états finis. Les étudiants mettent pratique ces méthodes au cours de projets. Enfin, j'encadre chaque année des étudiants de troisième année pour leur mini-projet.

Formation Continue. Dans le cadre de l'ISEA, je participe à la formation continue des élèves en alternance au sein du module de troisième année (12h de CM et 24h de TD) sur la conception de logiciel client pour SQL.

Master SIC. Dans le cadre du Master recherche SIC, je participe à l'unité d'enseignement complémentaire sur l'indexation multimédia, où j'enseigne les méthodes d'apprentissage pour la recherche multimédia (10h de CM). Je participe également à l'encadrement des mini-projets.

TABLE B.1 – Résumé des enseignements (hors encadrements)

Niveau	Heures	Description
Bac+3 1ère année ENSEA	364h	Langage C *Initiation à l'informatique bas niveau via le langage C.*
Bac+3 1ère année ENSEA	72h	Structures de données *Tableaux, Listes, Arbres, Graphes et algorithmes associés.*
Bac+4 2ème année ENSEA	96h	Langage Java *Programmation orientée objet avec Java.*
Bac+4 2ème année ENSEA	100h	Option Indexation *Algorithmes et structures de données pour la recherche rapide dans les bases de données relationnelles et multimédia.*
Bac+5 3ème année ENSEA	76h	Algorithmique *Algorithmes pour la résolution de problèmes en informatique (stockage, recherche, optimisation, jeux, ...)*
Bac+5 Master SIC	50h	Apprentissage statistique pour la recherche multimédia *Méthodes et algorithmes pour la recherche dans les bases multimédia*
Formation Continue 3ème année ISEA	168h	Java et SQL *Interaction avec les bases relationnelles avec JDBC*
Total	926h	

B.3.2 Encadrement de thèses

Encadrement de thèses

2006 Justine Lebrun « *Fonctions noyaux sur Graphes pour la recherche d'objets dans les bases multimédia* »
Allocation de recherche du ministère de l'enseignement supérieur
Thèse commencée en septembre 2006 et soutenue le 16 mai 2011
Encadrement Sylvie Philipp-Foliguet (50%), Philippe-Henri Gosselin (50%)
Publications : 1 Article Journal [IVC11], 1 Conférence internationale [ICPR08], 1 Conférence nationale [EGC08]

2007 Jean-Emmanuel Haugeard « *Extraction et reconnaissance d'objets dans les façades de Paris à l'aide d'appariement de graphes* »
Financement Agence Nationale de la Recherche (Projet iTown)
Thèse commencée en octobre 2007 et soutenue le 17 décembre 2010
Encadrement Sylvie Philipp-Foliguet (50%), Philippe-Henri Gosselin (25%), Frédéric Précioso (25%)
Publications : 1 Article Journal [PRL] (Soumis), 1 Conférence internationale [ICPR10b]

2009 Alexis Lechervy « *Apprentissage interactif et collaboratif pour la détection d'évènement dans les vidéos* »
Financement Direction Générale des Armées
Thèse commencée en octobre 2009
Encadrement Philippe-Henri Gosselin (60%), Frédéric Précioso (40%)
Publications : 1 Conférence internationale [ICPR10a], 1 Conférence nationale [RFIA10]

2009 Jefersson Alex dos Santos « *Multi-Scale Classification of Remote Sensing Images* »
Cette thèse financée par le Brésil est en cotutelle entre l'Université de Cergy-Pontoise et l'Université de Campinas
Thèse commencée en mars 2009 au Brésil, en janvier 2011 en France
Encadrement Brésil : Alexandre Xavier Falcao, Ricardo da S. Torres
Encadrement France : Philippe-Henri Gosselin (50%), Sylvie Philipp-Foliguet (50%)
Publication : 1 Article Journal [GRS]

2011 Romain Negrel « *Représentations optimales pour la recherche dans les bases d'images patrimoniales* »
Financement Labex Patrima
Thèse commencée en septembre 2011
Encadrement David Picard (50%), Philippe-Henri Gosselin (50%)

B.3.3 Encadrement de stages et mini-projets

Encadrement de stages Master

2004	Sylvia Colon « *Modélisation de la rétine humaine pour la recherche d'images* »
2005	Julien Gony « *Apprentissage pour la recherche d'images à l'aide de régions floues* »
	Guillaume Logerot « *Utilisation conjointe du texte et du contenu des images pour l'indexation de pages Web* »
2009	Alexy Lechervy « *Boosting interactif pour la recherche d'images* »
2010	Khaled Mohamed « *Descripteurs globaux pour l'indexation des objets 3D* »
	Mathias Fuzier « *Classification d'Objets 3D par appariement de graphes* »
	Virginia Fernandez Mota « *Tenseurs pour la reconnaissance dans les vidéos* »
2011	Romain Negrel « *Tenseurs pour la description locale du flot optique* »

Encadrement de mini-projets Master

2003	Aurélien Hazart « *Indexation d'images à l'aide de coefficients d'ondelettes* »
2007	Marion Piccinelli « *Segmentation vidéo 2D+T par quantification* »
2008	Guillaume Sargent « *Ondelettes complexes pour l'indexation* »
2009	Virginia Fernandez Mota « *Tenseurs pour la description du flot optique* »
2010	Romain Negrel « *Tenseurs pour la description locale du flot optique* »

Encadrement de mini-projets Ingénieur

2003	Nicolas Vivien « *Interface client/serveur Java pour le traitement des images* »
	Jean-Pierre Deville « *Techniques de compressions d'images* »
	Vincent Jousse « *Quantification vectorielle pour l'indexation* »
2004	Cédric Monier « *Transformée de Fourier en assembleur IA32 SSE* »
	Alexandre Torti « *JPEG 2000* »
2005	Nicolas Sergy « *Optimisation quadratique - application aux SVM* »
2008	Hugo Moll et Haithem MOUSSAOUI « *Mini SQL en C++* »
	Felix Lecardonnel « *Intelligence artificielle pour les jeux de cartes* »
2009	Fabien Martin et Romain Busuttil « *Calculs distribués sur un réseau de machines* »
	Gil Gougnot et Khalid Bellag « *Données distribuées sur un réseau de machines* »
	Youcef Mahoub et Guillaume Rene « *Service réseau multi-clients* »
2010	Benjamin Laroche « *Mise en correspondance de graphes sur GPGPU* »
	Antoine Cottin « *Gestionnaire de stock* »

B.4 Activités de recherche

B.4.1 Rayonnement

Chairman à Socpar 2010 (International Conference on Soft Computing and Pattern Recognition).

Membre(2005-2008) du réseau d'excellence européen MUSCLE (Multimédia Understanding through Semantics, Computation and LEarning).

Relecteur pour différents journaux : ACM Transactions on Multimedia Computing, Communications and Applications, Computer Vision and Image Understanding, Information Sciences, International Journal of Information Technology & Decision Making.

Relecteur pour différentes conférences : IEEE International Conference on Image Processing, International Conference on Pattern Recognition, International Conference on Computer Analysis of Images and Patterns

B.4.2 Valorisation

Création et Administration de la plateforme RETIN. Pendant mon doctorat, j'ai conçu une architecture logicielle pour la recherche interactive dans les bases multimédia. Elle est le principal support matériel de mes travaux, mais aussi de presque tous les étudiants qui travaillent sur la recherche multimédia à ETIS. J'assure régulièrement la formation des nouveaux arrivants ainsi que l'harmonisation des différents outils implantés par chacun. Actuellement à la quatrième version, cette plateforme gère toutes les facettes de la recherche multimédia, du stockage des descripteurs visuels aux techniques de recherche.

Membre du projet ANR iTowns (2008-2011). Dans le cadre de ce projet, j'ai co-encadré la thèse de Jean-Emmanuel Haugeard, dont l'objet est la conception de fonctions noyaux sur graphes pour la recherche dans les images de façades.

Membre du projet ANR EROS 3D (2006-2009). L'objet de ce projet était la conception de méthodes pour l'indexation d'œuvres d'art numérisées en trois dimensions. J'ai participé à tous les encadrements autour de cette thématique, autant sur le plan scientifique que technique.

Collaboration avec la Réunion des Musées Nationaux. (2011-2015). L'objet de cette collaboration est de travailler sur la base de 550.000 images de la RMN. Le but est d'étudier des solutions pour aider les clients de l'agence photographique de la RMN à trouver les images qu'ils recherchent.

Collaboration avec le Centre de Recherche et de Restauration des Musées de France (2002-2008). Mise en place du système RETIN pour la classification des photographies d'œuvres d'art et des modèles d'objets en trois dimensions.

Participation à la campagne TRECVid (2009-2011). Cette campagne, renouvelée tous les ans depuis 2003, permet de comparer des centaines de méthodes de par le monde. Dans le cadre du consortium IRIM organisé par le GDR-ISIS, j'ai participé à la tâche "Semantic Indexing" de la campagne d'évaluation TRECVid. J'ai fourni des signatures visuelles aux autres membres du consortium, qui les ont utilisées pour entraîner leurs classifieurs.

B.4.3 Responsabilités Administratives

Adjoint Hygiène et Sécurité à l'Ecole Nationale de l'Electronique et de ses Application depuis 2009.

Membre de la commission de spécialistes de l'Université de Caen en 2009 et 2010.

Membre de la commission de spécialistes de l'Université de Cergy-Pontoise en 2011.

B.5 Publications

Toutes les publications sont téléchargeables à l'adresse suivante :
`http://perso-etis.ensea.fr/~gosselin`

B.5.1 Synthèse

Revues	FI⋆	Total
IEEE Trans. on Image Processing (IP)	3.908	1
Elsevier Pattern Recognition (PR)	3.402	2
Elsevier Computer Vision and Image Understanding (CVIU)	2.730	2
Elsevier Image and Vision Computing (IVC)	1.840	2
EURASIP Journal on Advances in Signal Processing (JASP)	1.136	1

Revues soumises ou en révision	FI⋆	Total
IEEE Transactions on Geoscience and Remote Sensing (GRS)	2.995	(1)
Elsevier Computer Vision and Image Understanding (CVIU)	2.730	(1)
Elsevier Pattern Recognition Letters (PRL)	1.864	(1)
IEEE J. of Sel. Topics in Applied Eatch Obs. and Remote Sensing (JSTARS)	1.140	(1)

Conférences internationales	Total
IEEE Int. Conf. on Image Processing (ICIP)	6
IEEE Int. Conf. on Pattern Recognition (ICPR)	3
ACM Int. Conf. on Image and Video Retrieval (CIVR)	2
ACM Int. Workshop on 3D Object Retrieval	1
IEEE Int. Conf. on ACIVS	1
IEEE Int. Conf. on Multimedia and Expo	1
IEEE Int. Workshop on Multimedia CBAR	1
ACM Int. Workshop on Computer Vision meets Databases	1
Int. Workshop on Machine Learning Tech. for Multimedia	1

Conférences nationales	Total
Reconnaissance des Formes et Intelligence Artificielle (RFIA)	3
Extraction et Gestion des Connaissances (EGC)	1
Conférence sur l'Apprentissage Automatique (CAP)	1

⋆FI : Facteur d'impact sur 5 années, ISI Web of Knowledge / Journal Citation Reports, 2010.

B.5.2 Revues internationales avec comité de lecture

Soumises ou en révision

[CVIU] D. Picard and P.H. Gosselin
Efficient image signatures and similarities using tensor products of descriptors. Computer Vision and Image Understanding. Soumis.

[GRS] J. Alex dos Santos, P.H. Gosselin, S. Philipp-Foliguet, R. da S. Torres and A.X. Falcao.
Multi-Scale Classification of Remote Sensing Images. IEEE Transactions on Geoscience and Remote Sensing. Soumis.

[JSTARS] J. Alex dos Santos, P.H. Gosselin, S. Philipp-Foliguet, R. da S. Torres and A.X. Falcao.
Interactive Classification of High Resolution Images. IEEE Journal of Selected Topics in Applied Eatch Observation and Remote Sensing. Soumis.

[PRL] J.E. Haugeard, P.H. Gosselin, S. Philipp-Foliguet.
Kernel on Graphs of Contours for Image Retrieval. Pattern Recognition Letters. Soumis.

Parues ou acceptées

[IVC11] J. Lebrun, P.H. Gosselin, S. Philipp-Foliguet.
Inexact Graph Matching Based on Kernels for Object Retrieval in Image Databases. Image and Vision Computing. En cours de publication, Volume 29, Pages 716-729,doi :10.1016/j.imavis.2011.07.008, 2011.

[PR11] P.H. Gosselin, F.Precioso, S. Philipp-Foliguet.
Incremental Kernel Learning for Active Image Retrieval without Global Dictionaries. In Pattern Recognition, Special issue on Semi-Supervised Learning for Visual Content Analysis and Understanding. Volume 44, Issue 10-11, Pages 2244-2254, doi :10.1016/j.patcog.2010.12.006, 2011.

[CVIU09] S. Philipp-Foliguet, J. Gony, P.H. Gosselin.
FReBIR : an image retrieval system based on fuzzy region matching. In Computer Vision and Image Understanding. Volume 113, Issue 6, Pages 693-707, ISSN : 1077-3142, 2009.

[IP08] P.H. Gosselin, M. Cord.
Active learning methods for Interactive Image Retrieval. In IEEE Transactions on Image Processing. Volume : 17, Issue : 7, Pages 1200-1211, ISSN : 1057-7149, 2008.

[CVIU08] P.H. Gosselin, M. Cord, Sylvie Philipp-Foliguet.
Combining visual dictionary, kernel-based similarity and learning strategy for image category retrieval. In Computer Vision and Image Understanding, Special Issue on Similarity Matching in Computer Vision and Multimedia. Volume 110, Issue 3, Pages 403-417, ISSN :1077-3142, 2008.

[IVC07] M. Cord, P.H. Gosselin, S. Philipp-Foliguet.
Stochastic exploration and active learning for image retrieval. In Image and Vision Computing. N25, p. 14-23, 2007

[PR06] P.H. Gosselin, M. Cord.
Feature based approach to semi-supervised similarity learning. In Pattern Recognition, Special issue on Similarity-Based Pattern Recognition. N39, p. 1839-1851, 2006.

[JASP06] M. Cord, J. Fournier, P.H. Gosselin and S. Philipp-Foliguet.
Interactive Exploration for Image Retrieval. In EURASIP Journal on Applied Signal Processing, Special issue on Advances in Intelligent Vision Systems : Methods and Applications. N14, p. 2173-2186, 2006.

B.5.3 Conférences internationales avec publication des actes

[ICIP11] D. Picard and P.H. Gosselin.
Improving image similarity with vectors of locally aggregated tensors. In IEEE International Conference on Image Processing, Brussels, Belgium, September 2011.

[ICPR10a] A. Lechervy, P.H. Gosselin and F. Precioso.
Active Boosting for interactive object retrieval. In International Conference on Pattern Recognition, Istanbul, Turkey, August 2010.

[ICPR10b] J.E. Haugeard, S. Philipp-Foliguet, P.H. Gosselin.
Kernel on Graphs based on Dictionary of Paths for Image Retrieval. In International Conference on Pattern Recognition, Istanbul, Turkey, August 2010.

[ACM10] Sylvie Philipp-Foliguet, Michel Jordan, Matthias Fuzier, Philippe Gosselin.
Indexing of 3D Models Based on Graph of Surfacic Regions. In International Workshop on 3D Object Retrieval, ACM Multimedia 2010, Florence, Italy, October 2010.

[ICPR08] J. Lebrun, S. Philipp-Foliguet, P.-H. Gosselin.
Image retrieval with graph kernel on regions. In International Conference on Pattern Recognition, Tampa, Florida, USA, December 2008.

[ICIP07] P.H. Gosselin, M. Cord and S. Philipp-Foliguet.
Kernel on Bags of Fuzzy Regions for fast object retrieval. In IEEE International Conference on Image Processing, San Antonio, Texas, USA, September 2007.

[CIVR07a] P.H. Gosselin, M. Cord and S. Philipp-Foliguet.
Kernel on Bags for multi-object database retrieval. In ACM International Conference on Image and Video Retrieval, Amsterdam, The Netherlands, July 2007.

[CIVR07b] J. Gony, M. Cord, S. Philipp-Foliguet, P.H. Gosselin and M. Jordan.
RETIN : a smart interactive digital media retrieval system. In ACM International Conference on Image and Video Retrieval, Amsterdam, The Netherlands, July 2007.

[ICIP06a] P.H. Gosselin and M. Cord.
Precision-Oriented Active Selection for Interactive Image Retrieval. In IEEE International Conference on Image Processing, Atlanta, GA, USA, October 2006.

[ICIP06b] M. Cord and P.H. Gosselin.
Image Retrieval using Long-Term Semantic Learning. In IEEE International Conference on Image Processing, Atlanta, GA, USA, October 2006.

[ICME06] S. Philipp-Foliguet, G. Logerot, P. Constant, P.H. Gosselin and C. Lahanier.
Multimedia Indexing and Fast Retrieval Based on a Vote System. In IEEE International Conference on Multimedia and Expo, Toronto, Ontario, Canada, July 2006.

[ICIP05] P.H. Gosselin and M. Cord.
Semantic kernel learning for interactive image retrieval. In IEEE International Conference on Image Processing, Genova, Italy, September 2005.

[MLMM05] P.H. Gosselin and M. Cord.
Active learning techniques for user interactive systems : application to image retrieval. In International Workshop on Machine Learning techniques for processing MultiMedia content, Bonn, Germany, August 2005.

[MCBAR04] P.H. Gosselin and M. Cord.
Semantic Kernel Updating for Content-Based Image Retrieval. In IEEE International Workshop on Multimedia Content-based Analysis and Retrieval, Miami, Florida, USA, December 2004.

[ICIP04] P.H. Gosselin and M. Cord.
RETIN AL : An Active Learning Strategy for Image Category Retrieval. In IEEE International Conference on Image Processing, Singapore, October 2004.

[ACIVS04] P.H. Gosselin and M. Najjar and M. Cord and C. Ambroise.
Discriminative Classification vs Modeling Methods in CBIR. In IEEE International Conference on Advanced Concepts for Intelligent Vision Systems, Brussel, Belgium, September 2004.

[CVDB04] P.H. Gosselin and M. Cord.
A Comparison of Active Classification Methods for Content-Based Image Retrieval. In International Workshop on Computer Vision meets Databases, ACM Sigmod, Paris, France, June 2004.

B.5.4 Conférences nationales avec publication des actes

[RFIA12] R. Negrel, V. Fernandes, P.H. Gosselin, M. Vieira and F. Precioso.
Indexation des Bases Vidéos à l'aide d'une Modélisation du Flot Optique par Bases de Polynômes. In Proceedings of Reconnaissance des Formes et Intelligence Artificielle, Lyon, France, January 2012.

[RFIA10] A. Lechervy and P.H. Gosselin and F. Precioso.
Boosting actif pour la recherche interactive d'images. In Proceedings of Reconnaissance des Formes et Intelligence Artificielle, Caen, France, January 2010.

[EGC08] P.H. Gosselin, J. Lebrun and S. Philipp-Foliguet.
Recherche d'images par noyaux sur graphes de régions. In Extraction et Gestion des Connaissances (EGC), INRIA Sophia Antipolis, France, January 2008.

[CAP05] P.H. Gosselin and M. Cord.
Méthodes d'apprentissage sémantiques : application à la recherche d'images. In Conférence sur l'Apprentissage Automatique, Nice, France, June 2005.

[RFIA04] P.H. Gosselin and M. Najjar and M. Cord and C. Ambroise and S. Philipp-Foliguet.
Méthodes d'Apprentissage pour la Recherche d'Images par le Contenu. In Proceedings of Reconnaissance des Formes et Intelligence Artificielle, Toulouse, France, January 2004.

B.5.5 Campagne d'évaluation

[IRIM10] IRIM Consorsium.
IRIM at TRECVID 2010 : Semantic Indexing and Instance Search. In TRECVID'10 workshop, National Institute of Standards and Technology ed. Gaithersburg, MD, USA, 2010.

[IRIM09] IRIM Consorsium.
IRIM at TRECVID 2009 : High Level Feature Extraction. In TRECVID'09 workshop, National Institute of Standards and Technology ed. Gaithersburg, MD, USA, 2009.

B.5.6 Thèse

[THESE] P.H. Gosselin
Méhodes d'apprentissage pour la recherche de catégories dans des bases d'images.
Université de Cergy-Pontoise, France, Décembre 2005.

B.5.7 Chapitre de livre

[MLTM08] M. Cord and P.H. Gosselin
Machine Learning Techniques for Multimedia.
Chapter 5 "Online content-based image retrieval using active learning", Springer, ISBN 978-3-540-75170-0, 2008.

B.6 Séminaires

26/04/2011 **Improving image similarity with vectors of locally aggregated tensors.**
Journée GDR IRIM, Paris.

07/01/2011 **Apprentissage interactif pour la recherche dans les bases d'images.**
TexMex, IRISA CNRS, Rennes.

07/02/2007 **Kernels on Bags for Object Retrieval in Image Databases.**
Centre for Computational Statistics and Machine Learning, University College London.

05/12/2006 **Statistical learning methods for image category search and object class recognition.**
Machine Intelligence Laboratory, University of Cambridge.

15/09/2006 **Méthodes d'apprentissage faiblement supervisées pour la recherche d'images par le contenu.**
Journée GDR ISIS, Paris.

Bibliographie

[1] S. Aksoy, R.M. Haralick, F.A. Cheikh, and M. Gabbouj. A weighted distance approach to relevance feedback. In *IAPR International Conference on Pattern Recognition*, volume IV, pages 812–815, Barcelona, Spain, September, 3-8 2000.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Surf : Speeded up robust features. *Computer Vision and Image Understanding*, 110(3) :346–359, 2008.

[3] Karsten M. Borgwardt, Cheng Soon Ong, Stefan Schönauer, S. V. N. Vishwanathan, Alexander J. Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21(1) :47–56, 2005.

[4] Christian Böhm, Stefan Berchtold, and Daniel A. Keim. Searching in high-dimensional spaces : Index structures for improving the performance of multimedia databases. *ACM Computing Surveys*, 33(3) :322–373, 2001.

[5] W. L. Chan, H. Choi, and R. Baraniuk. Quaternion wavelets for image analysis and processing. In *International Conference on Image Processing*, volume 5, pages 3057–3060, October 2004.

[6] O. Chapelle, P. Haffner, and V. Vapnik. Svms for histogram based image classification. *IEEE Transactions on Neural Networks*, 10 :1055–1064, 1999.

[7] B. Collins, J. Deng, K. Li, and L. Fei-Fei. Towards scalable dataset construction : An active learning approach. In *European Conference on Computer Vision*, pages 86–98, 2008.

[8] I.J. Cox, M.L. Miller, T.P. Minka, T.V. Papathomas, and P.N. Yianilos. The bayesian image retrieval system, PicHunter : Theory, implementation and psychophysical experiments. *IEEE Transactions on Image Processing*, 9(1) :20–37, 2000.

[9] K. Crammer, J. Keshet, and Y. Singer. Kernel design using boosting. In *Advances in Neural Information Processing Systems*, pages 537–544. MIT Press, 2003.

[10] N. Cristianini, J. Shawe-Taylor, A. Elisseff, and J. Kandola. On kernel target alignement. In *Neural Information Processing Systems*, pages 367–373, Vancouver, Canada, December 2001.

[11] M. Datar, N. Immorlica, P. Indyk, and V.S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *ACM Symposium on Computational Geometry*, pages 253–262, 2004.

[12] Lili Diao, Keyun Hu, Yuchang Lu, and Chunyi Shi. A method to boost support vector machines. In *PAKDD '02 : Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, pages 463–468, London, UK, 2002. Springer-Verlag.

[13] N. Doulamis and A. Doulamis. A recursive optimal relevance feedback scheme for cbir. In *International Conference on Image Processing*, pages 741–744, Thessaloniki, Greece, October 2001.

[14] Martin Druon. *Modélisation du mouvement par polynômes orthogonaux : application à l'étude d'écoulements fluides*. PhD thesis, Université de Poitiers, 2009.

[15] R.O. Duda and P.E. Hart. *Pattern Classfication and Scene Analisys*. John Wiley and Sons, 1973.

[16] F.-X. Dupé and Luc Brun. Classification de formes avec un noyau sur graphes flexibles et robuste au bruit. In *Reconnaissance des Formes et Intelligence Artificielle*, Caen, France, January 2010.

[17] M. Ferecatu, N. Boujemaa, and M. Crucianu. Hybrid visual and conceptual image representation within active relevance feedback context. In *7th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR'05)*, Singapore, November 2005.

[18] J. Fournier, M. Cord, and S. Philipp-Foliguet. Retin : A content-based image indexing and retrieval system. *Pattern Analysis and Applications Journal, Special issue on image indexation*, 4(2/3) :153–173, 2001.

[19] Y. Freund and R. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, Bari, Italy, 1996.

[20] Y. Freund, R. Yyer, R.E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal on Machine Learning Research*, 4 :933–969, November 2003.

[21] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression : a statistical view of boosting. *The Annals of Statistics*, 2 :337–374, April 2000.

[22] T. Gärtner, P.A. Flach, and S. Wrobel. On graph kernels : Hardness results and efficient alternatives. In *Proceedings of the 16th Annual Conference on Computational Learning Theory and the 7th Kernel Workshop*, pages 129–143, 2003.

[23] Thomas Gärtner. A survey of kernels for structured data. *SIGKDD Explorations*, 5(1) :49–58, 2003.

[24] P.H. Gosselin and M. Cord. Precision-oriented active selection for interactive image retrieval. In *International Conference on Image Processing*, pages 3197–3200, Atlanta, GA, USA, October 2006.

[25] P.H. Gosselin, Micheline Najjar, M. Cord, and Christophe Ambroise. Discriminative classification *vs* modeling methods in CBIR. In *IEEE Advanced Concepts for Intelligent Vision Systems*, Brussel, Belgium, September 2004.

[26] Helmut Grabner and Horst Bischof. On-line boosting and vision. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 260–267, Washington, DC, USA, 2006. IEEE Computer Society.

[27] K. Grauman and T. Darell. The pyramid match kernel : Discriminative classification with sets of image features. In *IEEE International Conference on Computer Vision*, volume 2, pages 1458–1465, Beijing, China, October 2005.

[28] B. Haasdonk and E. Pekalska. Indefinite kernel fisher discriminant. In *IAPR International Conference on Pattern Recognition*, pages 1–4, Tampa, USA, Dec. 2008.

[29] Bernard Haasdonk. Feature space interpretation of svms with indefinite kernels. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(4) :482–492, 2005.

[30] David Haussler. Convolution kernels on discrete structures. Technical report, Institute of California at Santa Cruz, 1999.

[31] T. Hertz, A. Bar Hillel, and D. Weinshall. Learning a kernel function for classification with small training samples. In *International Conference on Machine Learning*, pages 401–408, 2006.

[32] Berthold K. P. Horn. Extended gaussian images. *Proceedings of the IEEE*, 72(2) :1671–1686, 1984.

[33] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *European Conference on Computer Vision*, pages 304–317. Springer, 2008.

[34] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 3304–3311, June 2010.

[35] T. Kin K. Tsuda and K. Asai. Marginalized kernels for biological sequences. *Bioinformatics*, 1 :268–275, 2002.

[36] S. B. Kang and K. Ikeuchi. The complex egi : A new representation for 3-d pose determination. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(7) :249–258, July 1993.

[37] H. Kashima and Y. Tsuboi. Marginalized kernels between labeled graphs. In *International Conference on Machine Learning*, pages 321–328, Washington, DC USA, August 2003.

[38] H. Kashima and Y. Tsuboi. Kernel-based discriminative learning algorithms for labeling sequences, trees and graphs. In *International Conference on Machine Learning*, page 58, Banff, Alberta, Canada, 2004.

[39] M. Kass, A. Witkin, and D. Terzopoulos. Snakes : Active contour models. *International Journal of Computer Vision*, 1 :321–331, 1988.

[40] M. Kawanabe, S. Nakajima, and A. Binder. A procedure of adaptive kernel combination with kernel-target alignment for object classification. In *ACM International Conference on Image and Video Retrieval*, 2009.

[41] N. G. Kingsbury. Complex wavelets for shift invariant analysis and filtering of signals. *Journal of Applied and Computational Harmonic Analysis*, 10(3) :234–253, May 2001.

[42] R. Kondor and T. Jebara. A kernel between sets of vectors. In *International Conference on Machine Learning*, pages 361–368, 2003.

[43] G. R. G. Lanckriet, N. Cristianini, N. Bartlett, L. El Ghaoui, and M. I. Jordan. Learning the kernel matrix with semi-definite programming. *International Journal on Machine Learning Research*, 5 :27–72, 2004.

[44] I. Laptev. *Local Spatio-Temporal Image Features for Motion Interpretation*. PhD thesis, KTH Numerical Analysis and Computer Science, Stockholm, 2004.

[45] S. Le and R. Nevatia. Extraction and integration of window in a 3D building model from ground view image. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 113–120, 2004.

[46] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems*, volume 19, pages 801–808, 2007.

[47] H. Lejsek, F.H. Ásmundsson, B. Jónsson, and L. Amsaleg. NV-Tree : An efficient disk-based index for approximate search in very large high-dimensional collections. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(5) :869–883, 2009.

[48] X. Li, L. Wang, and E. Sung. Improving adaboost for classfication on small training sample sets with active learning. In *Asian Conference on Computer Vision*, Jeju, Korea, January 2004.

[49] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2(60) :91–110, 2004.

[50] Y. Lu, Q. Tian, and T.S. Huang. Interactive boosting for image classification. In *International Conference on Multiple Classifier Systems*, pages 315–324, 2007.

[51] S. Lyu. Mercer kernels for object recognition with local features. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 223–229, San Diego, CA, 2005.

[52] P. Mahé and J.-P. Vert. Graph kernels based on tree patterns for molecules. *Machine Learning*, 75(1) :3–35, 2009.

[53] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *International Conference on Machine Learning*, pages 689–696, 2009.

[54] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *Advances in Neural Information Processing Systems*, volume 21, pages 1033–1040, 2008.

[55] B. S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. on Pattern Analysis and Machine Intelligence, Special Issue on Digital Libraries*, 18(8) :837–42, August 1996.

[56] L. Mason, J. Baxter, P. Bartlett, and M. Frean. Functional gradient techniques for combining hypotheses. *Advances in Large Margin Classifiers*, 1 :109–120, 2000.

[57] J. Matas, O. Chum, M. Urban, and T. Pajdla. Rosbut wide baseline stereo from maximally stable external regions. In *British Machine Vision Conference*, pages 384–393, 2002.

[58] Ajmal S. Mian, Mohammed Bennamoun, and Robyn Owens. Keypoint detection and local feature matching for textured 3d face recognition. *International Journal of Computer Vision*, 79(1) :1–12, August 2008.

[59] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *European Conference on Computer Vision*, pages 128–142, London, UK, 2002. Springer-Verlag.

[60] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1–2) :43–72, November 2005.

[61] F. Moosmann, E. Nowak, and F. Jurie. Randomized clustering forests for image classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(9) :1632–1646, september 2008.

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

[63] C.S. Ong, X. Mary, S. Canu, and A.J. Smola. Learning with non-positive kernels. In *International Conference on Machine Learning*, pages 639–646, 2004.

[64] Nikunj C. Oza. *Online Ensemble Learning*. PhD thesis, The University of California, Berkeley, CA, Sep 2001.

[65] G. Patanè and M. Russo. The enhanced LBG algorithm. *Neural Networks*, 14(9) :1219–1237, November 2001.

[66] Elzbieta Pekalska and Bernard Haasdonk. Kernel discriminant analysis for positive definite and indefinite kernels. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(6) :1017–1032, 2009.

[67] F. Perronnin and C. R. Dance. Fisher kernels on visual vocabularies for image categorization. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.

[68] S. Philipp-Foliguet and J. Gony. FReBIR : Fuzzy regions-based image retrieval. In *Information Processing and Management of Uncertainty (IPMU)*, pages 693–707, Paris, France, July 2006.

[69] Sylvie Philipp-Foliguet, Michel Jordan, Laurent Najman, and Jean Cousty. Artwork 3D model database indexing and classification. *Pattern Recognition*, 44(3) :588–597, 2011.

[70] S. Poullot, O. Buisson, and M. Crucianu. Scaling content-based video copy detection to very large databases. *Multimedia Tools and Applications*, 47 :279–306, 2009.

[71] J. Rodriguez and J. Maudes. Boosting recombined weak classifiers. *Pattern Recognition Letters*, 29(8) :1049–1059, 2007.

[72] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *International Conference on Machine Learning*, pages 441–448, 2001.

[73] Y. Rui, T. Huang, S. Mehrotra, and M. Ortega. A relevance feedback architecture for content-based multimedia information retrieval systems. In *IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 92–89, 1997.

[74] R.E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 3(37) :297–336, 1999.

[75] B Schölkopf and A Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[76] J. Shawe-Taylor and N. Cristianini. *Kernel methods for Pattern Analysis*. Cambridge University Press, ISBN 0-521-81397-2, 2004.

[77] N. Shervashidze and K. M. Borgwardt. Fast subtree kernel on graphs. In *Advances in Neural Information Processing Systems*, pages 1660–1668, 2009.

[78] N. Shervashidze, S. V. N. Vishwanathan, T.H. Petri, K. Mehlhorn, and K. M. Borgwardt. Efficient graphlet kernels for large graph comparison. In *AISTATS*, volume 5, pages 488–495, 2009.

[79] J.R. Smith and S.F. Chang. VisualSEEK : a fully automated content-based image query system. In *ACM Multimedia*, pages 87–98, Boston, USA, November 1996.

[80] A.J. Smola, P. Barlett, B. Scholkopf, and C. Schuurmans. *Advances in Large Margin Classifiers*. MIT Press, 2000.

[81] Frédéric Suard, Vincent Guigue, Alain Rakotomamonjy, and Abdelaziz Bensrhair. Pedestrian detection using stereo-vision and graph kernels. In *Intelligent Vehicles Symposium*, pages 267–272, Las Vegas, Nevada, June 2005.

[82] Nguyen Thuy, B. D. Nguyen, and Bischof Horst. Efficient boosting-based active learning for specific object detection problems. In *International Conference on Computer Vision, Image and Signal Processing*, 2008.

[83] S. Tong and D. Koller. Support vector machine active learning with application to text classification. *International Journal on Machine Learning Research*, 2 :45–66, November 2001.

[84] N. Vasconcelos. *Bayesian models for visual information retrieval*. PhD thesis, Massachusetts Institute of Technology, 2000.

[85] A. Vezhnevets and V. Vezhnevets. Modest adaboost – teaching adaboost to generalize better. *Graphicon*, 5 :987–997, September 2005.

[86] L. Vincent and P. Soille. Watersheds in digital spaces : an efficient algorithm based on immersion simulations. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(6) :583–598, 1991.

[87] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 511–518, 2001.

[88] S. V. N. Vishwanathan, Karsten M. Borgwardt, Imre Risi Kondor, and Nicol N. Schraudolph. Graph kernels. *CoRR*, abs/0807.0093, 2008.

[89] S. V. N. Vishwanathan and Alexander J. Smola. Fast kernels for string and tree matching. In *Advances in Neural Information Processing Systems*, pages 569–576. MIT Press, 2003.

[90] C. Wallraven, B. Caputo, and A.B.A. Graf. Recognition with local features : the kernel recipe. In *IEEE International Conference on Computer Vision*, volume 2, pages 257–264, 2003.

[91] Jingjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *IAPR International Conference on Pattern Recognition*, pages 3360–3367, 2010.

[92] L. Wolf and I. Martin. Robust boosting for learning from few examples. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages I : 359–364, 2005.