



ECOLE DOCTORALE SCIENCES ET INGÉNIERIE
de l'Université de Cergy-Pontoise

Manuscrit de thèse pour l'obtention du titre de docteur en Sciences et Technologies
de l'Information et de la Communication

**REPRÉSENTATIONS OPTIMALES POUR LA RECHERCHE
DANS LES BASES D'IMAGES PATRIMONIALES**

par

ROMAIN NEGREL

ETIS - ENSEA / Université de Cergy-Pontoise / CNRS UMR 8051
6 avenue du Ponceau, 95014 Cergy-Pontoise, France

Soutenue le 03 Décembre 2014

Devant un jury présidé par M. Bernard Merialdo et composé de

M. Florent Perronnin	Directeur de recherche au Centre de Recherche Xerox Europe	Rapporteur
M. Georges Quénot	Directeur de Recherche au CNRS Responsable du groupe MRIM	Rapporteur
M. Bernard Merialdo	Professeur à l'Institut Eurecom Département Communications Multimedia	Examineur
M. Frédéric Jurie	Professeur à l'Université de Caen Département Image	Examineur
M. Philippe-Henri Gosselin	Professeur à l'ENSEA ETIS – MIDI	Directeur de thèse
M. David Picard	Maître de conférences à l'ENSEA ETIS – MIDI	Co-encadrant de thèse

Remerciements

Je voudrais tout d'abord remercier MM. Florent Perronnin, Georges Quénot, Bernard Merialdo et Frédéric Jurie d'avoir accepté de participer à mon jury de thèse, et pour leurs remarques pertinentes et leurs questions très intéressantes lors de la soutenance.

Je tiens également à exprimer de vifs et chaleureux remerciements envers Philippe-Henri Gosselin et David Picard, mes deux encadrants de thèse. Je les remercie pour la qualité de leur encadrement, leur aide, leur soutien, leur disponibilité et la confiance qu'ils m'ont accordés.

Je remercie Inbar Fijalkow, directrice d'ETIS lors de mon arrivée et Mathias Quoy, actuel directeur, pour leur accueil au sein du laboratoire et pour leur soutien dans mes démarches administratives. Je remercie également Dan Vodislav directeur de l'équipe MIDI pour son soutien.

Je remercie le LabEx PATRIMA de la Fondation des Sciences du Patrimoine qui a financé cette thèse sans quoi ces travaux n'auraient pas été possibles et qui m'a permis de rencontrer des doctorants, des chercheurs et de nombreux acteurs du patrimoine.

Je souhaite aussi remercier Mohamed El Amine Benkhelifa et Fakhreddine Ghaffari pour leur aide et leur disponibilité surtout à l'occasion de mes enseignements à l'IUT de Cergy-Pontoise - site de Neuville-sur-Oise.

Je remercie l'ensemble des membres de l'équipe MIDI – Olivier Kihl, Michel Jordan, Son Vu, Hedi Tabia, Jérôme Fellus – pour leur aide et tout ce que nous avons partagé.

Un grand merci à Annick Bertinotti, Anthony Carqueijeiro et à toutes les personnes des services administratifs d'ETIS, de l'ENSEA et de l'UCP pour leur aide dans toutes les démarches administratives, pour leur efficacité permanente et leur disponibilité.

Je tiens tout particulièrement à remercier l'ensemble des membres du laboratoire ETIS et plus particulièrement les doctorants sans lesquels la vie au laboratoire aurait été différente. J'ai été accueilli par une équipe de doctorants très soudée formée de Laurent R., Ludo, JC, Leila, Mathilde, Erbao, Alan, Fouad, Alexis, Gaël, Romain T., Rémi, Liang et tant d'autres. Je remercie Anne Savard, Laurent Fiack et également Antoine Tauvel avec qui j'ai fondé l'association Nova-Robotics et avec qui j'ai conçu deux robots qui se sont brillamment illustrés à la coupe de France de robotique. Un grand merci également à Yuhui, Jérôme, Agathe, Raouia, Frédéric D., David P.-Q. et tant d'autres qui ont rendu ces années inoubliables.

Je remercie David Picard, Philippe-Henri Gosselin, Elena-Veronica Belmega, Anne Savard et Laurent Fiack pour l'aide qu'ils m'ont apporté lors de la rédaction de mon mémoire de thèse.

Je remercie ma famille qui m'a soutenu toutes ces années sans jamais douter de moi et qui a toujours été une source d'encouragement et de réconfort.

Je tiens à remercier l'ensemble de mes amis avec qui j'ai partagé tant de moments inoubliables, les parties de jeux de rôle interminables, les soirées discussions prise de tête sans intérêt, les sorties aux bars, aux cinés, et de nombreuses autres aventures. Pour tout cela, je vous dis un grand merci.

Enfin, je souhaite dédier cette thèse à Veronica, la personne la plus importante de ma vie. Tu m'as encouragé, conseillé, soutenu pendant toutes ces années, pour cela je t'exprime toute ma gratitude et mon amour.

Résumé

Depuis plusieurs décennies, le développement des technologies de numérisation et de stockage ont permis la mise en œuvre de nombreux projets de numérisation du patrimoine culturel. L’approvisionnement massif et continu de ces bases de données numériques du patrimoine culturel entraîne de nombreux problèmes d’indexation. En effet, il n’est plus possible d’effectuer une indexation manuelle de toutes les données. Pour indexer et rendre accessible facilement les données, des méthodes d’indexation automatique et d’aide à l’indexation se sont développées depuis plusieurs années. Cependant, les méthodes d’indexation automatique pour les documents non-textuels (image, vidéo, son, modèle 3D, ...) sont encore complexes à mettre en œuvre pour de grands volumes de données.

Dans cette thèse, nous nous intéressons en particulier à l’indexation automatique d’images. Pour effectuer des tâches d’indexation automatique ou d’aide à l’indexation, il est nécessaire de construire une méthode permettant d’évaluer la similarité entre deux images. Nos travaux sont basés sur les méthodes à signatures d’image ; ces méthodes consistent à résumer le contenu visuel de chaque image dans une signature (vecteur unique), puis d’utiliser ces signatures pour calculer la similarité entre deux images. Pour extraire les signatures, nous utilisons la chaîne d’extraction suivante : en premier, nous extrayons de l’image un grand nombre de descripteurs locaux ; puis nous résumons l’ensemble de ces descripteurs dans une signature de grande dimension ; enfin nous réduisons fortement la dimension de la signature. Les signatures de l’état de l’art basées sur cette chaîne d’extraction permettent d’obtenir de très bonnes performances en indexation automatique et en aide à l’indexation. Cependant, les méthodes de l’état de l’art ont généralement de forts coûts mémoire et calculatoires qui rendent impossible leur mise en œuvre sur des grands volumes de données.

Dans cette thèse, notre objectif est double : d’une part nous voulons améliorer les signatures d’images pour obtenir de très bonnes performances dans les problèmes d’indexation automatique ; d’autre part, nous voulons réduire les coûts de la chaîne de traitement, pour permettre le passage à l’échelle. Nous proposons des améliorations d’une signature d’image de l’état de l’art nommée VLAT (*Vectors of Locally Aggregated Tensors*). Ces améliorations permettent de rendre la signature plus discriminante tout en réduisant sa dimension.

Pour réduire la dimension des signatures, nous effectuons une projection linéaire de la signature dans un espace de petite dimension. Nous proposons deux méthodes pour obtenir des projecteurs de réduction de dimension tout en conservant les performances des signatures d’origine. Notre première méthode consiste à calculer les projecteurs qui permettent d’approximer le mieux possible les scores de similarités entre les signatures d’origine. La deuxième méthode est basée sur le problème de recherche de quasi-copies ; nous calculons les projecteurs qui permettent de respecter un ensemble de contraintes sur le rang des images dans la recherche par rapport à l’image requête. L’étape la plus coûteuse de la chaîne d’extraction est la réduction de dimension de la signature à cause de la grande dimension des projecteurs.

Pour les réduire, nous proposons d’utiliser des projecteurs creux en introduisant une contrainte de parcimonie dans nos méthodes de calcul des projecteurs. Comme il est généralement complexe de résoudre un problème d’optimisation avec une contrainte de parcimonie stricte, nous proposons pour chacun des problèmes une méthode pour obtenir une approximation des projecteurs creux recherchés.

L’ensemble de ces travaux font l’objet d’expériences montrant l’intérêt pratique des méthodes proposées par comparaison avec les méthodes de l’état de l’art.

Abstract

In the last decades, the development of scanning and storing technologies resulted in the development of many projects of cultural heritage digitization. The massive and continuous flow of numerical data in cultural heritage databases causes many problems for indexing. Indeed, it is no longer possible to perform a manual indexing of all data. To index and ease the access to data, many methods of automatic and semi-automatic indexing have been proposed in the last years. The current available methods for automatic indexing of non-textual documents (images, video, sound, 3D model, ...) are still too complex to implement for large volumes of data.

In this thesis, we focus on the automatic indexing of images. To perform automatic or semi-automatic indexing, it is necessary to build an automatic method for evaluating the similarity between two images. Our work is based on image signature methods; these methods involve summarising the visual content of each image in a signature (single vector), and then using these signatures to compute the similarity between two images. To extract the signatures, we use the following pipeline : First, we extract a large number of local descriptors of the image; Then we summarize all these descriptors in a large signature; Finally, we strongly reduce the dimensionality of the resulting signature. The state of the art signatures based on this pipeline provide very good performance in automatic indexing. However, these methods generally incur high storage and computational costs that make their implementation impossible on large volumes of data.

In this thesis, our goal is twofold : First, we wish to improve the image signatures to achieve very good performance in automatic indexing problems; Second, we want to reduce the cost of the processing chain to enable scalability. We propose to improve an image signature of the state of the art named VLAT (Vectors of Locally Aggregated Tensors). Our improvements increase the discriminative power of the signature.

To reduce the size of the signatures, we perform linear projections of the signatures in a lower dimensional space. We propose two methods to compute the projectors while maintaining the performance of the original signatures. Our first approach is to compute the projectors that best approximate the similarities between the original signatures. The second method is based on the retrieval of quasi-copies; We compute the projectors that meet the constraints on the rank of retrieved images with respect to the query image. The most expensive step of the extraction pipeline is the dimensionality reduction step; these costs are due to the large dimensionality of the projectors.

To reduce these costs, we propose to use sparse projectors by introducing a sparsity constraint in our methods. Since it is generally complex to solve an optimization problem with a strict sparsity constraint, we propose for each problem a method for approximating sparse projectors.

This thesis work is the subject of experiments showing the practical value of the proposed methods in comparison with existing methods

Table des matières

I	Introduction	3
I.1	Outils d'aide à l'indexation	4
I.2	La recherche par le contenu	4
I.2.1	Les critères d'évaluation	5
I.3	Objectifs de la thèse	6
I.4	Contribution	7
I.4.1	Signatures	7
I.4.2	Réduction de dimension	7
I	Les signatures d'images	11
II	État de l'art	15
II.1	Descripteurs d'images	15
II.1.1	Descripteurs globaux	15
II.1.2	Descripteurs locaux	16
II.2	Similarité sur sac	20
II.2.1	Similarité par vote	20
II.2.2	Les noyaux sur sac	21
II.2.3	Signatures et Similarité	22
II.2.4	Pré-traitements et Post-traitements	28
III	Méthode proposée	31
III.1	Vision déviation de modèle de la signature VLAT	31
III.2	Optimisation du modèle	32
III.2.1	ACP dans les parties du modèle	32
III.2.2	Réduction de dimension	34
III.3	Expériences	34
III.3.1	Étude des coûts	35
III.3.2	Recherche par similarité	36
III.3.3	Classification automatique	39
III.4	Conclusions	41
II	Réduction de dimension	43
IV	Réduction de dimension non-supervisée	45
IV.1	État de l'art	45
IV.1.1	Réduction de dimension	46
IV.1.2	Codage de signature	49
IV.2	Méthodes proposées	50
IV.2.1	Projecteurs denses	50

IV.2.2	Projecteurs creux	54
IV.2.3	Correction d'erreurs de similarité	55
IV.3	Expériences	56
IV.3.1	Recherche par similarité	57
IV.3.2	Classification automatique	63
IV.4	Conclusions	64
V	Réduction de dimension supervisée	67
V.1	État de l'art	67
V.1.1	Métriques	68
V.1.2	Apprentissage de similarité	70
V.2	Méthode proposée	72
V.2.1	Problème d'optimisation pour la reconstruction du rang	73
V.2.2	Résolution par descente de gradient stochastique	74
V.2.3	Introduction d'une contrainte de parcimonie	76
V.3	Expériences	77
V.3.1	Recherche par similarité	77
V.4	Conclusions	81
VI	Conclusions générales	83
VI.1	Principales contributions	83
VI.1.1	Amélioration des performances de la signature	83
VI.1.2	Réduction de la dimension des signatures	83
VI.1.3	Réduction des coûts de redimensionnement	84
VI.2	Perspectives	84
VI.2.1	Perspectives à court terme	84
VI.2.2	Perspectives à moyen terme	85
VI.2.3	Perspectives à long terme	85
	Appendices	87
A	Les bases d'images	89
A.1	Bases d'images non annotées	89
A.1.1	Flickr60k	89
A.1.2	Flickr1M	89
A.2	Bases de classification	90
A.2.1	Base VOC2007	90
A.3	Bases de copies et quasi-copies d'images	91
A.3.1	Base Holidays	91
A.3.2	Base Oxford	91
A.3.3	Base Paris	92
B	Les mesures de performance	95
C	Matrices et vecteurs creux	97
C.1	Stockage des vecteurs creux	97
C.2	Opérations avec des vecteurs creux	98

Table des figures

II.1	Exemple d'extraction dense de descripteurs locaux (grille de 3 pixels sur 4 échelles de l'image).	17
II.2	Chaîne d'extraction des descripteurs HOG.	19
II.3	Effet de la normalisation en loi de puissance sur la dynamique des valeurs d'une signature.	29
III.1	Coût de stockage et coût calculatoire en fonction du nombre de dimensions moyennes conservées par partie (signature VLAT avec un modèle optimisé de $K = 64$ partie pour un sac de $N = 1000$ descripteurs locaux de dimension $D = 128$).	35
III.2	Performances (mAP) de la signature VLAT optimisé sur Holidays en fonction sa dimension W	38
III.3	Performances (mAP) de la signature VLAT optimisée sur VOC2007 en fonction sa dimension W	41
IV.1	Exemple de projection linéaire de vecteurs de dimension 3 dans un espace de dimension 2.	46
IV.2	Exemple d'axes principaux obtenus par l'ACP	47
IV.3	Exemple de projection non-linéaire	48
IV.4	Étude l'influence du nombre de parties D dans le modèle de la signature VLAT sur les performances en recherche par similarité en fonction du nombre d'images dans la base.	59
IV.5	Comparaison de notre signature avec les signatures de l'état de l'art en fonction du nombre d'images dans la base ([†] extrait [Jégou et al., 2012]).	59
IV.6	Évolution du mAP en recherche par similarité avec la base Holidays en fonction du cout calculatoire de l'étape de réduction de dimension pour différent taux de parcimonie (<i>i.e.</i> , $\tau \in \{99.9\%, 99\%, 90\%, 0\%\}$).	62
IV.7	Performance en mAP de la signature VLAT réduite en fonction de la dimensions de la signature N	64
V.1	Évolution de la fonction objectif et du mAP sur l'ensemble d'entraînement sans la contrainte de parcimonie.	80
V.2	Évolution de la fonction objectif et du mAP sur l'ensemble d'entraînement avec la contrainte de parcimonie et plusieurs taux de parcimonie.	81
V.3	Évolution du mAP sur l'ensemble d'entraînement et l'ensemble de teste avec la contrainte de parcimonie.	82
A.1	Exemple d'image de la base VOC 2007	90
A.2	Exemple d'image de la base Holidays	91
A.3	Exemple d'image de la base Oxford	92
A.4	Exemple d'image de la base Paris	92

Liste des tableaux

III.1	Performances en recherche par similarité des signatures de l'état de l'art sur la base Holidays	37
III.2	Performances en classification des signatures de l'état de l'art sur la base VOC2007	40
III.3	Comparaisons aux résultats de l'état de l'art en classification d'image sur VOC 2007 extraient de [Chatfield et al., 2011].	42
IV.1	Étude de l'influence des paramètres en recherche par similarité sur la base Holidays avec $D = 64$ (mAP).	58
IV.2	Performances en recherche par similarité de notre signature réduite par projection dense et des signatures de l'état de l'art la base Oxford et Oxford avec 100k distracteurs (mAP).	60
IV.3	Performances en recherche par similarité de signature VLAT et des signatures de l'état de l'art la base Holidays et Oxford (mAP).	61
IV.4	Évaluation de la robustesse des signatures en recherche par similarité par ajout de 100k distracteurs et comparaison avec l'état de l'art dans les base Holidays et Oxford (mAP).	63
V.1	Performances des signatures VLAD et FV non réduites et réduites avec la méthode proposé dans la section IV.2	79
V.2	Performances des signatures réduites sur l'ensemble de teste avec différent taux de parcimonie avec les projecteurs non-supervisés (Section IV.2.2.a) (mAP en %).	79
V.3	Performances des signatures réduites sur l'ensemble de teste avec différent taux de parcimonie (mAP en %).	80
B.1	Exemple de calcul du rappel et de la précision avec $ \mathcal{P}_q = 10$	96
B.2	Meilleurs performance obtenu sur les bases de l'état de l'art [Sharif Razavian et al., 2014].	96

Introduction

Depuis les peintures préhistoriques jusqu'à aujourd'hui, l'Homme n'a pas cessé d'inventer de nouvelles méthodes pour s'exprimer et communiquer. Pour diffuser et conserver ces informations, de nombreuses méthodes de stockage et d'indexation ont été inventées. Il y a encore quelques décennies le stockage de l'information sous forme physique était prédominant. Cependant, l'apparition des technologies numériques et leurs rapides évolutions ont bouleversé les méthodes de stockage des données mais également les méthodes de partage des données. De nos jours, la majorité des personnes ont accès à des technologies numériques leur permettant de stocker à tout moment du texte, des images, des vidéos, du son. Elles ont également accès au réseau Internet leur permettant de partager ces informations. Cela a généré une véritable explosion des quantités de données stockées. Depuis sa création en 2004, le site Flickr est passé de quelques milliers d'images à plus de 50 millions d'images ajoutées par mois¹. Plus de 350 millions de photos sont ajoutées chaque jour sur le réseau social Facebook². Sur youtube 100 heures de vidéo sont ajoutées chaque minute³. À ces bases de données s'ajout les campagnes de numérisation du patrimoine culturel. Le projet *Google Books*, initié par l'entreprise Google en 2004 a dépassé les 20 millions de livres numérisés en 2012⁴. Le projet de bibliothèque numérique Gallica de la Bibliothèque nationale de France contient à ce jour plus de 3 millions de documents⁵ (livres, manuscrits, partitions, images, fascicules de presse et revues, *etc*). La Réunion des Musées Nationaux (RMN) réalise des campagnes de prises de vues à l'intérieur des musées nationaux depuis les années 60 et réalise 20000 nouvelles prises de vues chaque année⁶. Le musée Rodin s'est également lancé dans une campagne de numérisation des 25 000 documents concernant l'œuvre d'Auguste Rodin ainsi que celles de Camille Claudel.

Si l'indexation manuelle des documents était encore envisageable quand les données étaient stockées sous forme matérielles, l'indexation totalement manuelle des bases de documents numériques est aujourd'hui difficilement envisageable. En effet, indexer des bases

1. <http://www.flickr.com/photos/franckmichel/6855169886/>

2. <http://www.businessinsider.com/facebook-350-million-photos-each-day-2013-9>

3. <https://www.youtube.com/yt/press/statistics.html>

4. <https://chronicle.com/article/Google-Begins-to-Scale-Back/131109/>

5. <http://gallica.bnf.fr/>

6. <http://www.photo.rmn.fr/>

de documents qui ont une croissance si rapide nécessiterait un très grand nombre de documentalistes. Il est donc complexe de garantir l'homogénéité de la sémantique utilisée pour indexer les documents. L'inhomogénéité de la sémantique rend alors difficile de répondre à des questions simples comme : "Existe-t-il des doublons dans la base ?" ou encore "Peut-on retrouver tout les documents qui ont des similitudes avec un document spécifique ?". Pour indexer ces bases de documents, il est indispensable d'automatiser une partie du travail effectué par les documentalistes.

I.1 Outils d'aide à l'indexation

Pour l'indexation de très grands volumes de documents, il est donc nécessaire de construire des outils capables d'aider les documentalistes. À partir d'une analyse automatique des documents, ces outils doivent être capables de fournir des informations pertinentes aux documentalistes sur ces documents. Par exemple, effectuer de la :

- **recherche de doublons** qui consiste à retrouver les documents identiques ou contenant des parties identiques. Par exemple des textes contenant des paragraphes identiques ou des images qui ont subi des transformations géométriques simples (découpe, rotation, *etc*) ;
- **recherche de quasi-copie** qui consiste à retrouver les documents qui ont une forte similarité. Par exemple des photographies du même objet prises dans des environnements différents ;
- **recherche interactive** qui consiste en un dialogue entre le documentaliste et le système dans le but de retrouver tous les documents appartenant à une classe sémantique. Le documentaliste commence par fournir des exemples de document qui l'intéresse. Le système utilise alors ces exemples pour déduire l'ensemble des documents de la base qui pourraient intéresser le documentaliste. Le documentaliste corrige alors le système en lui indiquant quels sont les documents qui l'intéresse et qui ne l'intéresse pas. Le système utilise ces nouvelles informations pour déduire un nouvel ensemble de documents. Ces opérations sont itérées jusqu'à ce que le système ait retrouvé tous les documents de la classe sémantique recherchés ;
- **suggestion d'annotations** qui consiste à proposer les annotations les plus probables pour un document non annoté à partir des autres documents annotés de la base.

La mise en œuvre de ces outils soulève un problème fondamental : comment faire le lien entre les informations numériques d'un document et sa sémantique ? En effet, si nous prenons le cas des images, il faut faire le lien entre le codage des couleurs de chaque pixel de l'image et l'ensemble des objets, des scènes ou toute autre signification que l'on peut y associer. Ce problème est connu sous le nom de fossé numérique/sémantique.

I.2 La recherche par le contenu

La recherche par le contenu réunit l'ensemble des approches qui analysent le contenu des documents pour en déduire des relations entre ces documents ou des informations sémantiques. Ces approches sont généralement opposées aux approches textuelles pour lesquelles chaque document est manuellement annoté avec une liste de mots-clefs qui décrivent au mieux leur contenu. Dans les approches textuelles les relations entre les documents sont alors directement obtenues dans l'espace sémantique. Les documentalistes

peuvent effectuer des recherches directement dans l'espace sémantique avec des mots-clefs. En recherche de documents par le contenu, la sémantique des documents n'est pas explicitée. Pour effectuer une recherche, le documentaliste doit fournir un ensemble de documents qui servent à définir le concept recherché.

En général, les approches de recherche par le contenu se déroulent en 2 étapes : une étape d'analyse des documents et une étape de recherche. L'étape d'analyse des documents consiste à extraire de chaque document les informations nécessaires pour effectuer les recherches par le contenu. Chaque document est analysé une seule fois à son ajout dans la base et les données extraites sont stockées avec le document. L'étape de recherche est effectuée à chaque fois que le documentaliste fait une requête au système. Cette étape utilise les données extraites des documents pour répondre à la requête (par exemple, "Y-a-t'il des documents similaires au document requête dans la base?" ou encore "Quelles sont les annotations les plus pertinentes pour ce document?"). Pour effectuer les recherches, le système évalue des similarités entre les documents pour en déduire des relations. Le point clef de ces méthodes consiste alors à déterminer quelles sont les données à extraire des documents (par exemple les éléments visuels saillants pour des images) et comment calculer des similarités entre les données extraites.

Dans cette thèse, nous nous intéressons particulièrement aux bases de documents issues de la numérisation du patrimoine (Gallica, RMN, musée Rodin, *etc.*). Ces bases ont plusieurs spécificités qui les rendent particulièrement intéressantes. Elles sont en partie issues de la numérisation de bases matérielles déjà annotées et leur contenu est alors en partie annoté. De plus, elles sont utilisées par des entreprises et des universitaires, l'indexation est alors très précise, souvent basée sur l'utilisation d'un thésaurus.

Nous nous intéressons à la recherche d'images par le contenu, et plus particulièrement aux méthodes basées sur les signatures d'images. Ces méthodes consistent à analyser chaque image pour en extraire une signature prenant la forme d'un unique vecteur décrivant son contenu. Pour extraire les signatures, nous considérons la chaîne de traitement suivante :

- **L'extraction de descripteurs locaux** qui consiste à extraire de l'image un ensemble de descripteurs, chaque descripteur permet de décrire une petite région de l'image.
- **L'agrégation en une signature** qui consiste à agréger l'ensemble des descripteurs locaux de l'image en une unique signature de grande dimension.
- **La réduction de la dimensionnalité de la signature** qui consiste à projeter les signatures dans un espace de petite dimension.

Pour évaluer la similarité entre deux images, nous utilisons alors une fonction de similarité entre les signatures extraites des deux images.

I.2.1 Les critères d'évaluation

Pour évaluer une méthode de recherche par le contenu, plusieurs points sont à prendre en compte.

I.2.1.a Les performances

Les performances d'un système de recherche par le contenu sont évaluées en fonction du nombre d'erreurs qu'il fait. Par exemple, la performance d'un système de recherche de doublons est évaluée en comptant le nombre faux positifs et de faux négatifs retournés

par le système. Ou encore, pour évaluer les performances d'un système de suggestion d'annotations, nous comptons le nombre d'annotations incorrectes proposées.

Le but est alors de créer des systèmes de recherche par le contenu qui font le moins d'erreurs possible.

I.2.1.b Les coûts d'extraction

Les coûts d'extraction sont les coûts générés par l'étape d'analyse des documents pour en extraire les données utilisées pour la recherche par le contenu. Deux types de coûts sont à prendre en compte :

- **Le coût calculatoire** qui correspond au nombre d'opérations que le système doit effectuer pour obtenir les données utilisées pour la recherche par le contenu à partir du document. Le coût calculatoire est directement proportionnel au temps nécessaire à l'extraction.
- **Le coût de stockage** qui correspond au volume de stockage occupé par les paramètres de la méthode d'extraction. Par cela, nous entendons toutes les données qu'il est nécessaire de fournir à la méthode avant de commencer l'extraction des données.

Pour traiter de très grandes quantités de documents il est indispensable de réduire au minimum ces coûts. En effet, plus le coût calculatoire est faible plus le système peut analyser un grand nombre de documents dans le même laps de temps. De plus, plus le coût de stockage est faible plus il est facile de distribuer l'analyse des documents sur un grand nombre de machines.

I.2.1.c Les coûts de recherche

Les coûts de recherche sont les coûts générés par la recherche dans la base de documents (par exemple, le coût pour parcourir toutes les données pour trouver les doublons, ou encore le coût pour évaluer tous les classificateurs pour de la suggestion d'annotations). Comme pour l'extraction, il y a deux types de coûts à prendre en compte :

- **Le coût calculatoire** qui correspond au nombre d'opérations que le système doit faire pour effectuer la recherche.
- **Le coût de stockage** qui correspond au volume de stockage occupé par les données nécessaires pour effectuer la recherche.

Ces coûts sont généralement liés avec la dimension des données extraites des documents. Pour indexer un grand nombre de documents, il est indispensable de réduire au minimum ces coûts en minimisant la dimension des données extraites des documents.

I.3 Objectifs de la thèse

Les signatures d'images obtenues à partir de l'agrégation de descripteurs locaux permettent d'obtenir de bonnes performances dans les problèmes d'indexation automatique par le contenu. Cependant, leur passage à l'échelle pour leur utilisation sur de très grands volumes d'images est complexe à mettre en œuvre. En effet dans la chaîne de traitement de l'extraction des signatures d'images, l'étape de réduction de dimension des signatures a de forts coûts mémoires et calculatoires qui deviennent prohibitif pour les bases d'images de grande dimension.

L'objectif de la thèse est double, d'une part nous voulons améliorer les signatures d'images pour obtenir de très bonnes performances dans les problèmes d'indexation automatique. D'autre part, nous voulons réduire les coûts de la chaîne de traitement, pour permettre le passage à l'échelle.

I.4 Contribution

Dans cette section, nous présentons les contributions apportées par les travaux de cette thèse.

I.4.1 Signatures

De nombreuses approches ont été proposées pour agréger les descripteurs locaux en une signature. L'approche par déviation de modèle consiste à mesurer la déviation entre un modèle statistique des descripteurs locaux et les descripteurs locaux des images. Cette approche permet d'obtenir de bonnes performances en recherche d'images par similarité. Notre première contribution est une amélioration d'une signature de l'état de l'art basée sur les approches à déviation de modèle. Nous proposons d'améliorer le modèle utilisé par cette signature pour réduire la dimension de la signature obtenue. De plus, ce nouveau modèle permet d'obtenir des propriétés statistiques intéressantes qui facilitent la réduction de la dimension de la signature. Nous présentons en détail cette contribution dans la première partie de cette thèse.

Ces travaux ont fait l'objet des publications suivantes : [Negrel et al., 2012b]
[Negrel et al., 2013]
[Negrel et al., 2014b]

I.4.2 Réduction de dimension

Pour réduire la dimension des signatures une approche couramment utilisée est la projection linéaire de la signature dans un espace de petite dimension. Nous proposons deux méthodes pour obtenir des projecteurs linéaires.

I.4.2.a Méthode non supervisée

Notre deuxième contribution est une méthode basée sur un apprentissage non supervisé pour obtenir les projecteurs. Cette méthode est basée sur la conservation de la similarité obtenue entre les signatures non réduites. Nous proposons également une amélioration qui vise à réduire drastiquement les coûts de stockage et calculatoire entraînés par la grande dimension de la matrice de projection. Pour cela, nous proposons une méthode pour obtenir des projecteurs creux et une méthode pour corriger les erreurs induites par l'approximation. Nous présentons en détail ces contributions dans la deuxième partie de cette thèse.

Ces travaux ont fait l'objet des publications suivantes : [Negrel et al., 2012a]
[Negrel et al., 2012b]
[Negrel et al., 2013]
[Negrel et al., 2014a]

I.4.2.b Méthode supervisée

Notre troisième contribution est une méthode supervisée pour obtenir les projecteurs. Cette méthode est basée sur la reconstruction du rang des images dans le tri des images pour la recherche de quasi-copie. De plus, nous proposons d'introduire dans notre méthode un critère de parcimonie pour obtenir des projecteurs creux. Nous présentons en détail ces contributions dans la troisième partie de cette thèse.

Ces travaux ont fait l'objet de la publication suivante : [Negrel et al., 2014c]

— Publications —

Revue internationale avec comité de lecture

- [Negrel et al., 2013] Negrel, R., Picard, D., and Gosselin, P.H. *Web-scale image retrieval using compact tensor aggregation of visual descriptors*. In MultiMedia, IEEE (Vol.20, Issue :3), pages 24-33, July-Sept. 2013.

Conférences internationales avec actes et comité de lecture

- [Negrel et al., 2012a] Negrel, R., Picard, D., and Gosselin, P.H. *Compact tensor based image representation for similarity search*. In Image Processing (ICIP), 2012 19th IEEE International Conference on, Orlando, FL, pages 2425-2428, Sept. 30 2012.
- [Negrel et al., 2012b] Negrel, R., Picard, D., Gosselin, P.H. *Using spatial pyramids with compacted vlat for image categorization*. Proceedings of the 21st International Conference on Pattern Recognition (ICPR), 2012, Tsukuba, Japan, pages 2460-2463, November 11-15, 2012.
- [Negrel et al., 2014a] Negrel, R., Picard, D., and Gosselin, P.H. *Dimensionality reduction of visual features using sparse projectors for content-based image retrieval*. In Image Processing (ICIP), 2014 21th IEEE International Conference on, Paris, FR, Octob. 27 2014.
- [Negrel et al., 2014b] Negrel, R., Picard, D., and Gosselin, P.H. *Evaluation of second-order visual features for land-use classification*. In Content-Based Multimedia Indexing (CBMI), 2014 12th International Workshop on, pages 1-5. IEEE.
- [Negrel et al., 2014c] Negrel, R., Picard, D., Gosselin, P.H. *Efficient dimension reduction of global signature with sparse projectors for image near duplicate retrieval*. Proceedings of the 22st International Conference on Pattern Recognition (ICPR), 2014, Stockholm, Sweden, ,August 24-28, 2014.

Première partie

Les signatures d'images

Comme nous l'avons présenté en introduction, pour effectuer des tâches d'indexation automatique (recherche de doublons, annotation automatique, ...), il est indispensable de pouvoir calculer une similarité entre deux images. Dans le premier chapitre cette partie, nous présentons une vue d'ensemble des méthodes pour calculer automatiquement une similarité entre deux images. Nous nous intéresserons plus particulièrement aux méthodes basées sur les signatures d'image. Puis dans le chapitre suivant, nous présentons notre méthode pour obtenir des signatures d'image. Notre signature est basée sur la mesure de la déviation entre l'image et un modèle générique d'image. Nous proposons pour cela un nouveau modèle optimisé, permettant de réduire la dimension des signatures obtenues tout en améliorant le pouvoir de discrimination de la signature. Enfin, nous présentons un ensemble d'expériences en recherche par similarité sur la base d'images Holidays, ainsi qu'un ensemble d'expériences en classification automatique sur la base d'image VOC2007.

État de l'art

Chapitre II

Dans ce chapitre, nous faisons l'état de l'art des méthodes pour : l'extraction de caractéristique visuelle, leurs mise en forme et le calcul de similarité basé sur ces caractéristiques visuelles.

II.1 Descripteurs d'images

Une des étapes essentielles en analyse d'images par le contenu est l'extraction d'une description de bas niveau de l'image. Cette description est une représentation numérique (analyse statistique, analyse quantitative, ...) des caractéristiques visuelles de l'image, généralement sous la forme d'un vecteur ou d'un ensemble de vecteurs.

Cette description doit avoir deux caractéristiques importantes :

- elle doit conserver suffisamment d'information pour être discriminante, c'est à dire différencier deux images avec un contenu visuel différent ;
- elle doit est le plus invariante possible (aux bruits ; aux variations d'échelle ; aux variations de contraste ; aux déformations ; *etc*) pour pouvoir généraliser le contenu visuel d'une image à une autre image qui n'est pas identique.

Nous présentons dans la suite deux familles de descripteurs : les descripteurs globaux et les descripteurs locaux.

II.1.1 Descripteurs globaux

Les descripteurs globaux permettent de résumer l'ensemble des informations visuelles de l'image en un unique descripteur (vecteur). Les descripteurs globaux sont construits en utilisant une ou plusieurs primitives visuelles¹ d'une image et en les résumant sous la forme d'un vecteur.

La couleur fait partie des premières primitives visuelles utilisées. Le plus simple des descripteurs globaux basés sur la couleur consiste à construire l'histogramme des couleurs

1. la couleur, la luminosité, le gradient de la luminosité, la réponse à un banc de filtres, les formes, les texture, *etc*.

à partir des valeurs des pixels dans l'espace RGB [Han and Ma, 2002, Chakravarti and Meng, 2009]. Pour rendre invariants les descripteurs basés sur la couleur aux variations de luminosité, d'ombre et de capteur, de nombreuses améliorations ont été proposées. L'utilisation de méthode de normalisation [Van De Sande et al., 2010] ainsi que l'utilisation d'espace colorimétrique adapté (HSV [Van De Weijer et al., 2006], L*a*b [Hays and Efros, 2008] ou encore LUV [Li and Wang, 2003]) permet d'améliorer la qualité de l'information couleur conservée.

De nombreux descripteurs globaux sont également basés sur la texture. Tamura et al. font partie des premiers à en proposer un, sur l'imitation du système de perception visuelle humain [Tamura et al., 1978]. Des approches statistiques basées sur la matrice de cooccurrence de l'intensité lumineuse [Tamura et al., 1978], ou encore sur le modèle MSAR (*Multiresolution Simultaneous Autoregressive Models*) [Mao and Jain, 1992] ont également été proposés. Une autre approche populaire est d'utiliser la réponse de l'intensité lumineuse des pixels à un banc de filtres orientés (par exemple les filtres de Gabor) [Martin et al., 2001, Liu et al., 2010].

Des descripteurs basés sur la distribution des contours ont également été proposés [Won et al., 2002, Wang et al., 2006].

II.1.2 Descripteurs locaux

Les descripteurs locaux sont de nos jours les caractéristiques visuelles les plus couramment utilisées en analyse d'image par le contenu. Ils permettent de décrire l'ensemble des informations visuelles d'une région de l'image en un descripteur (vecteur). La région d'extraction est appelée région d'intérêt et elle est généralement centrée sur un point d'intérêt. Les régions et points d'intérêt peuvent être obtenus par deux approches : avec un détecteur de points/régions d'intérêt ; ou par une grille dense.

Les descripteurs locaux ont une très bonne capacité de discrimination pour déterminer si deux régions sont similaires ou dissimilaires. Cela les rends particulièrement utiles en vision par ordinateur, notamment pour de la mise en correspondance de points d'intérêts. Ils sont utilisés en : odométrie visuelle [Nistér et al., 2004] ; reconstruction 3D [Mouragnon et al., 2006] ; détection d'objets [Lowe, 1999] ; recherche d'images par le contenu [Perronnin et al., 2010b, Jégou et al., 2008, Picard and Gosselin, 2011] ; etc.

Dans la suite, nous présentons une introduction sur les méthodes de détections de régions d'intérêts, puis nous présenterons un formalisme pour décrire les étapes d'extraction d'un descripteur local ainsi que plusieurs exemples de descripteurs locaux couramment utilisés en indexation multimédia.

II.1.2.a Régions d'intérêts

Les détecteurs de points/régions d'intérêt permettent de détecter automatiquement des régions de l'image qui sont informatives. La notion de point/région d'intérêt est apparue en premier lieu dans deux domaines de la vision par ordinateur : la vision en robotique [Moravec, 1980] et l'étude de la vision humaine [Koch and Ullman, 1987, Itti et al., 1998]. En robotique, un point/région d'intérêt est vu comme une région de forte variation de l'image. Ces variations doivent être de forte amplitude et doivent avoir des caractéristiques structurelles précises. En étude de la vision humaine, et plus particulièrement en étude de l'attention visuelle, un point/région d'intérêt est vu comme une région

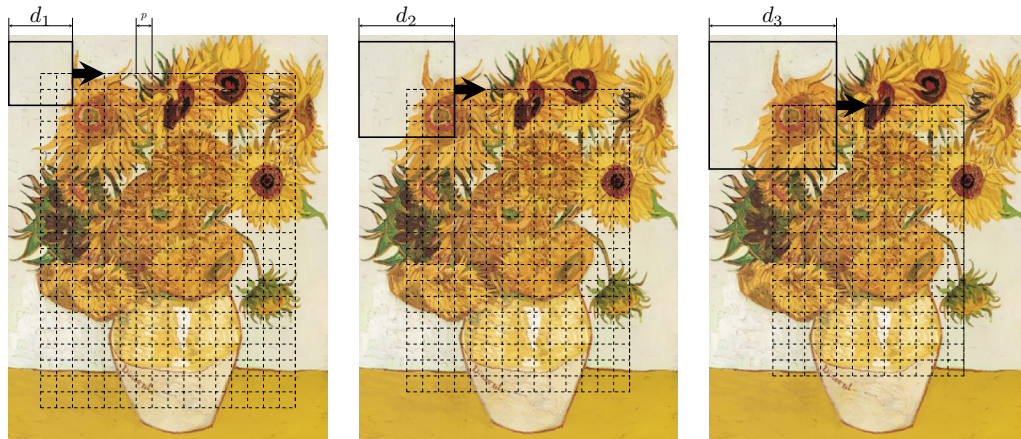


FIGURE II.1 – Exemple d'extraction dense de descripteurs locaux (grille de 3 pixels sur 4 échelles de l'image).

de forte énergie dans l'image.

En analyse d'image par le contenu, les détecteurs de points d'intérêt les plus couramment utilisés sont basés sur des approches variationnelles. Le détecteur de Harris *et al.* [Harris and Stephens, 1988] et le détecteur de Shi-Tonasi *et al.* [Shi and Tomasi, 1994] sont des extensions du détecteur de Moravec *et al.* [Moravec, 1980]. Ils utilisent les valeurs propres du tenseur de structure local basé sur le gradient de l'image pour détecter des régions de forte variation caractéristique. Kitchen *et al.* ont proposé un détecteur basé sur la courbure des contours dans l'image [Kitchen and Rosenfeld, 1982]. De nombreux détecteurs sont basés sur un filtre convolutif de Hessien de gaussiennes (LoG) [Lindeberg, 1998, Mikolajczyk and Schmid, 2004], des méthodes par différence de gaussiennes (DoG) [Lowe, 2004] et des méthodes par déterminant du Hessien (DoH) [Lindeberg, 1998]. Mikolajczyk *et al.* ont publié une évaluation complète de nombreux détecteurs [Mikolajczyk and Schmid, 2004, Mikolajczyk et al., 2005].

II.1.2.b Extraction dense

L'extraction dense de descripteurs locaux consiste à calculer des descripteurs locaux dans toutes les régions de l'image. La méthode la plus courante consiste à calculer un descripteur tous les P pixels et sur plusieurs échelles (Figure II.1).

Cette méthode est très utilisée pour le calcul de similarité entre images [Jégou et al., 2010, Delhumeau et al., 2013, Negrel et al., 2013], car elle permet d'extraire de plus grande quantité d'information de l'image. Cependant, le coût de stockage de l'ensemble des descripteurs extraits de l'image est prohibitif. Par exemple, pour une image de 256 pixels si nous calculons un descripteur de 128 dimensions tous les 3 pixels sur 4 échelles, nous calculons alors environ 30000 descripteurs. Le coût de stockage de l'ensemble de descripteurs pour une image est de 14Mo, soit 14To de descripteurs pour une base d'un million d'images.

Il est donc indispensable d'utiliser des méthodes qui ne nécessitent pas le stockage des descripteurs locaux.

II.1.2.c Formalisme d'extraction des descripteurs locaux

Les descripteurs locaux sont généralement basés sur une primitive visuelle de l'image, ces primitives subissent plusieurs traitements, et elles sont finalement agrégées en un unique vecteur. Kihl *et al.* ont proposé un formalisme pour unifier la majorité des descripteurs locaux [Kihl et al., 2013]. Ce formalisme définit les étapes nécessaires pour obtenir un descripteur local à partir d'une région de l'image. L'extraction des descripteurs est découpée en trois étapes :

- **L'extraction d'une primitive visuelle** consistant à extraire une information bas niveau (par exemple : le gradient, la réponse à des ondelettes de Haar 2D, la couleur, ...);
- **Le codage de la primitive visuelle** consistant à effectuer une injection non-linéaire de la primitive visuelle dans un espace de grande dimension. Le but de cette étape est d'améliorer la représentation en rapprochant les primitives visuelles similaires entre elles et en éloignant les primitives dissimilaires ;
- **L'agrégation des codes** consistant à rassembler toutes les primitives codées dans un unique code. L'objectif de l'agrégation des codes est de rendre robuste la mise en correspondance des descripteurs locaux aux petites déformations.

Ce formalisme de création de descripteurs locaux permet d'obtenir des descripteurs qui sont fortement discriminants tout en tolérant certaines déformations. C'est pour cela qu'ils sont couramment utilisés pour la mise en correspondance de points d'intérêt et pour calculer une similarité visuelle.

II.1.2.d Le descripteur SIFT

Le descripteur SIFT (*Scale-invariant feature transform*) est un descripteur basé sur le gradient de l'intensité de l'image. Il a été proposé par Lowe *et al.* pour la mise en correspondance de points d'intérêt [Lowe, 1999, Lowe, 2004].

Les auteurs proposent un détecteur de point d'intérêt robuste basé sur des différences de gaussienne (DoG) multi-échelle. Ce détecteur permet de déterminer l'échelle et l'orientation principale du point d'intérêt.

Le formalisme proposé par Kihl *et al.*, permet de décrire les étapes de construction du descripteur SIFT : La primitive visuelle utilisée est le gradient (représenté par son orientation et son intensité), il est obtenu par la convolution de l'image avec un masque dérivateur. À chaque orientation est appliquée une rotation égale à l'orientation du point d'intérêt. Cela permet, de rendre invariant par rotation le descripteur. Le codage est effectué par la quantification de l'orientation du gradient avec une affectation pondérée aux deux classes les plus proches. La quantification est effectuée sur 8 orientations réparties uniformément. Pour conserver l'information spatiale de la distribution du gradient dans la région d'intérêt, la région à décrire est divisée en une grille régulière de 4×4 cellules. L'agrégation des codes du gradient est effectuée dans chaque cellule en calculant la somme pondérée des codes dans une cellule. La pondération est obtenue en fonction de deux paramètres : l'intensité du gradient et la distance du pixel au centre du descripteur. Le descripteur alors obtenu est un vecteur de 128 dimensions.

De nombreuses extensions du descripteur SIFT ont été proposées : Le descripteur GLOH (*Gradient Location and Orientation Histogram*) est un descripteur SIFT pour lequel les cellules sont une grille log-polaire [Mikolajczyk and Schmid, 2005]. Le descripteur PCA-SIFT permet de réduire la dimension du descripteur en effectuant une analyse en

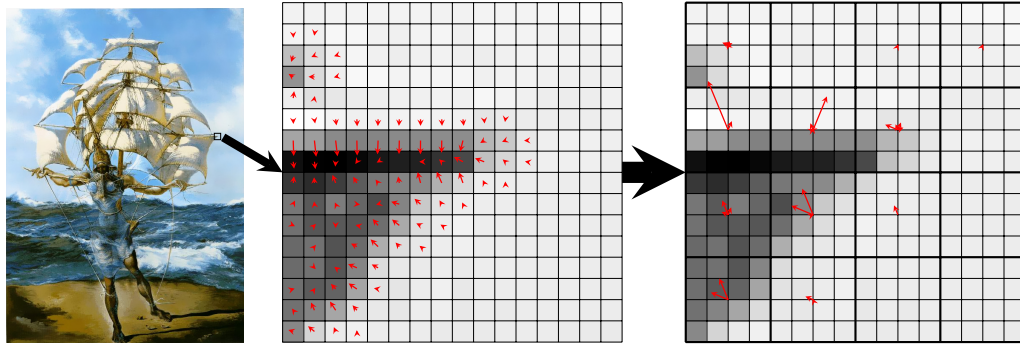


FIGURE II.2 – Chaîne d'extraction des descripteurs HOG.

composantes principales (ACP) du descripteur SIFT [Ke and Sukthankar, 2004]. Le descripteur RootSIFT est un descripteur SIFT dont les composantes ont été normalisées en puissance [Jain et al., 2012, Arandjelovic and Zisserman, 2012].

Une approximation du descripteur SIFT a également été proposée dans le but de fortement réduire le coût calculatoire de l'extraction [Bay et al., 2006]. Le descripteur SURF estime l'orientation dominante au voisinage de points d'intérêt et uniquement deux orientations (horizontale et verticale) sont utilisées. Dans chaque cellule, la somme de l'intensité du gradient et la somme de la valeur absolue du gradient sont effectuées pour les deux orientations. Le descripteur est alors de 64 dimensions.

II.1.2.e Histogramme de gradients orientés

Le descripteur HOG (*Histogram of Oriented Gradients*) est un descripteur basé sur le gradient de l'intensité. Il a été proposé dans [Dalal and Triggs, 2005] pour la détection d'objet dans les images.

Le descripteur HOG est très similaire au descripteur SIFT, il utilise le gradient de l'intensité comme primitive visuelle et la même structure en cellules. Nous pouvons utiliser le formalisme proposé par Kihl *et al.* pour décrire les étapes de construction du descripteur HOG (Figure II.2) : La primitive visuelle utilisée est le gradient (il est représenté par son orientation et son intensité) obtenu généralement par un filtrage convolutif. Le codage est effectué par la quantification de l'orientation du gradient avec une affectation pondérée aux deux classes les plus proches. La quantification est généralement effectuée sur 8 orientations réparties uniformément. L'agrégation des codages du gradient est effectuée par une simple somme des codes dans une cellule. Dalal *et al.* proposent d'utiliser un découpage de 16×16 cellules [Dalal and Triggs, 2005]. Cependant, il est courant d'utiliser un découpage similaire à celui utilisé pour le descripteur SIFT. L'échelle d'extraction du descripteur HOG est définie par la largeur S des cellules. La dimension d'un descripteur HOG est alors de $D = 128$ dimensions.

De nombreuses extensions du descripteur HOG ont été proposées : Zhu *et al.* proposent de calculer les histogrammes dans des cellules de taille variable [Zhu et al., 2006]. Wang *et al.* proposent d'utiliser une interpolation tri-linéaire du gradient et de combiner l'histogramme de gradient avec l'histogramme des LBP (*Local Binary Pattern* [Ahonen et al., 2006]) dans chaque cellule [Wang et al., 2009].

II.1.2.f Descripteurs *X-Color*

Perronnin *et al.* ont proposé un descripteur local couleur pour la classification et la recherche d'images par similarité [Perronnin et al., 2010b]. Ce descripteur utilise directement le codage RGB (Rouge, Vert, Bleu) des couleurs comme primitive visuelle. Dans ce descripteur, la primitive visuelle n'est pas codée. Comme pour le descripteur HOG, l'agrégation est effectuée dans une grille de cellules. Mais contrairement au descripteur HOG, Perronnin *et al.* proposent de calculer pour chaque cellule la moyenne et la variance de chaque composante couleur. La dimension du descripteur avec une grille de 4×4 cellules est alors de $D = 96$ dimensions.

II.2 Similarité sur sac

Dans cette section, nous présentons l'état de l'art sur les méthodes qui permettent de calculer des similarités entre deux sacs de descripteurs. Un sac est un ensemble non-ordonné de descripteurs, il regroupe l'ensemble des descripteurs d'une image. Nous notons par $B_i = \{\mathbf{b}_{ri}\}_r \subset \mathcal{B}$ le sac des descripteurs de la $i^{\text{ème}}$ image avec $\mathbf{b}_{ri} \in \mathbb{R}^D$ le $j^{\text{ème}}$ descripteur de la $i^{\text{ème}}$ image et $\mathcal{B}$ l'ensemble des descripteurs de toutes les images d'un corpus donné :

$$\mathcal{B} = \bigcup_i B_i. \quad (\text{II.1})$$

L'utilisation de sacs de descripteurs pour calculer une similarité visuelle entre images est intuitivement simple à justifier. En effet, si deux images représentent des objets/lieux similaires, ou plus généralement le même concept sémantique, elles doivent contenir des régions qui sont structurellement similaires (par exemple, même forme ou même texture). L'utilisation de descripteurs locaux et de leurs mise en correspondance permet de détecter et de quantifier le nombre de régions similaires entre les deux images. Ainsi plus le nombre de régions similaires est grand plus les images sont similaires.

De nombreuses méthodes ont été proposées pour calculer des similarités entre deux sacs de descripteurs. Elles peuvent être classées en deux grandes catégories : (i) les méthodes qui utilisent directement l'ensemble des descripteurs des sacs ou une approximation de ses descripteurs pour calculer une similarité ; (ii) les méthodes qui convertissent les sacs de descripteurs en signatures (vecteurs).

II.2.1 Similarité par vote

Dans les approches basées sur le vote [Mikolajczyk and Schmid, 2001, Gorisse, 2010], les descripteurs de l'image requête sont mis en correspondance avec l'ensemble des descripteurs de toutes les images de la base $\mathcal{B}$. Chaque descripteur de la requête vote pour ses k -Plus Proches Voisins (k -PPV) dans $\mathcal{B}$. Puis, chaque image de la base compte le nombre de votes obtenus par ses descripteurs. L'image avec le plus grande nombre de votes est l'image la plus similaire à l'image requête.

Le score de similarité du sac $\mathbf{B}_j$ relativement à une requête $\mathbf{B}_i$ est obtenu avec l'équation suivante :

$$k(\mathbf{B}_i, \mathbf{B}_j) = \sum_{\mathbf{b}_{ri} \in \mathbf{B}_i} \text{card} \left(k\text{-PPV}(\mathbf{b}_{ri}) \cap \mathbf{B}_j \right), \quad (\text{II.2})$$

avec $\text{card}(\cdot)$ le cardinal de l'ensemble, et $k\text{-PPV}(\mathbf{b}_{r_i})$ l'ensemble des k -Plus Proche Voisins de $\mathbf{b}_{r_i}$ dans l'ensemble $\mathcal{B}$ privé de $\mathbf{B}_i$.

Cette famille de méthodes est très efficace pour résoudre le problème de recherche de doublons. Par exemple dans une base de 42500 images où chaque image a subi une ou plusieurs transformations (rotations, tramage, bruitage, floutage, homographie, recadrage, occultation, recadrage, combinaison d'images avec mise à l'échelle, compression), la méthode par vote permet d'obtenir 98.33% de mAP (*mean Average Precision*) [do Valle Jr, 2008].

II.2.2 Les noyaux sur sac

Les méthodes par vote sont difficilement compatibles avec les méthodes d'apprentissage statistique et nécessitent la mise en œuvre de méthodes complexes pour les utiliser en classification automatique. Une bonne approche pour rendre les méthodes par vote compatibles avec les méthodes d'apprentissage statistique classique (par exemple les SVM) consiste à les approximer par une fonction noyau.

Les noyaux sur sac sont une famille de méthodes permettant de calculer un produit scalaire directement entre deux sacs, vu comme une mesure de similarité entre sacs [Schölkopf and Smola, 2002, Shawe-Taylor and Cristianini, 2004]. Pour cela, Shawe-Taylor *et al.* [Shawe-Taylor and Cristianini, 2004] utilisent la propriété suivante des fonctions noyaux : la somme de fonctions noyaux est un noyau. En calculant l'ensemble des similarités entre les descripteurs des deux sacs, ils proposent alors le noyau sur sac *softmax* pour calculer la similarité entre deux sacs :

$$S(B_i, B_j) = \frac{1}{|B_i|} \frac{1}{|B_j|} \sum_{\mathbf{x} \in B_i} \sum_{\mathbf{y} \in B_j} s(\mathbf{x}, \mathbf{y}) \quad (\text{II.3})$$

avec $s(\mathbf{x}, \mathbf{y})$ une fonction noyau utilisée pour évaluer la similarité entre deux descripteurs locaux et $|B_i|$, $|B_j|$ le nombre de descripteurs dans les sacs i et j . Ce noyau donne des bons résultats quand les descripteurs sont très discriminants ou quand les sacs contiennent peu de descripteurs. En effet, pour un grand nombre de descripteurs, le nombre de descripteurs similaires entre les deux sacs devient très petit par rapport au nombre de descripteurs dissimilaires. Les fortes similarités sont alors noyées par la grande quantité de petites similarités.

Pour pallier ce problème, Lyu *et al.* ont proposé d'améliorer la capacité de mise en correspondance de la fonction de similarité des descripteurs en la mettant à la puissance $p \geq 1$:

$$S(B_i, B_j) = \frac{1}{|B_i|} \frac{1}{|B_j|} \sum_{\mathbf{x} \in B_i} \sum_{\mathbf{y} \in B_j} s(\mathbf{x}, \mathbf{y})^p. \quad (\text{II.4})$$

Cela permet de mieux approcher les méthodes par votes.

Les noyaux sur sac permettent de conserver de très bonne performances en recherche par similarité, par exemple, Precioso *et al.* obtiennent un mAP de 76% sur la base Holidays [Precioso et al., 2011]. Les noyaux sur sac sont également utilisés en classification, par exemple, Gorisse *et al.* obtiennent plus de 90% de mAP sur la base VOC 2006 (mAP calculé uniquement sur les 100 premières images du classement) [Gorisse et al., 2008, Everingham et al., 2006].

II.2.3 Signatures et Similarité

À cause de la grande quantité de descripteur dans les sacs, l'utilisation de noyaux sur sac en indexation par le contenu entraîne de forts coûts calculatoires et de stockage. Pour réduire ces coûts des méthodes d'indexation et d'approximation des descripteurs locaux et de les noyaux sur sac sont utilisés [Gorisse et al., 2008]. Cependant, elles sont généralement complexes à mettre en œuvre et il est difficile d'obtenir des garanties sur les propriétés des fonctions noyaux après leur approximation. Une alternative à ces méthodes est d'agréger l'ensemble des descripteurs d'un sac en une signature (vecteur réel $\mathbf{x} \in \mathbb{R}^W$) et d'utiliser une fonction noyau dans l'espace vectoriel.

II.2.3.a La signature *Bag of Words*

La signature *Bag of Words* (BoW) ou encore *Bag of visual Words* (BoVW) est l'une des premières méthodes proposées pour résumer l'ensemble des descripteurs visuels d'une image en une unique signature [Sivic and Zisserman, 2003]. Elle est héritée des méthodes d'analyse de texte [Baeza-Yates et al., 1999], et consiste à calculer l'histogramme des occurrences de motifs prédéfinis. En analyse de texte, un dictionnaire de mots est prédéfini. Pour adapter cette méthode aux sacs de descripteurs locaux, il est donc nécessaire de construire un dictionnaire de mots visuels. Sivic *et al.* ont proposé de calculer une partition de l'espace des descripteurs visuels de telle sorte que chaque partie représente un concept visuel spécifique [Sivic and Zisserman, 2003]. La partition de l'espace des descripteurs locaux est obtenue à l'aide d'un algorithme de type k-means [Lloyd, 1982], où chaque partie est alors définie par un mot visuel (barycentre de la partie).

Pour calculer la signature d'un sac de descripteurs, il suffit de compter le nombre descripteurs du sac qui appartiennent à chaque partie. Pour une partition de K mots visuels (parties), la signature est de dimension $W = K$.

Cette méthode est très simple à mettre en œuvre et conserve de bonnes performances en classification, par exemple, elle obtient un mAP de 58.0% sur la base VOC 2007 avec 4000 mots visuels et le noyau χ_2 [van Gemert et al., 2010]. Elle conserve également de bonnes performances en recherche par similarité, par exemple, elle obtient un mAP de 54.0% sur la base Holidays avec 400000 mots visuels et le produit scalaire usuel [Jégou et al., 2012].

II.2.3.b La signature par *Coding/pooling*

Les signatures par *Coding/pooling* est une généralisation de la signature BoW, elle s'inspire des méthodes utilisées en codage [Lloyd, 1982]. Les signatures par *Coding/pooling* décomposent le calcul de la signature en deux étapes : le codage (ou *Coding*) ; et l'agrégation (ou *Pooling*).

L'étape de codage consiste à coder chaque descripteur du sac, généralement en appliquant un traitement non-linéaire sur les descripteurs qui les projette dans un espace de plus grande dimension.

L'étape de mise en commun consiste à rassembler l'ensemble des descripteurs codés en une unique signature.

Par exemple, pour la signature BoW, la fonction de codage s'écrit alors :

$$\begin{aligned} \mathbf{c}_{ri}^* &= \arg \min_{\mathbf{c}_{ri}} \|\mathbf{b}_{ri} - \mathbf{D}\mathbf{c}_{ri}\|^2 \\ \text{s.t. } &\|\mathbf{c}_{ri}\|_0 = 1, \|\mathbf{c}_{ri}\|_1 = 1, \mathbf{c}_{ri} \succeq 0, \forall r; \end{aligned} \tag{II.5}$$

avec $\mathbf{D}$ la matrice de K mots visuels, $\mathbf{b}_{ri}$ le $r^{\text{ème}}$ descripteur visuel du sac $\mathbf{B}_i$ et $\mathbf{c}_{ri}$ le vecteur du code du descripteur $\mathbf{x}_{ri}$. Le vecteur de codage est un vecteur parcimonieux avec une unique valeur unitaire correspondante au mot visuel le plus proche du descripteur. La fonction de mise en commun de la signature BoW est simplement la somme des codes :

$$\mathbf{c}_i = \sum \mathbf{c}_{ri}^* \quad (\text{II.6})$$

La première signature utilisant ce formalisme est la signature ScSPM proposée par Yang *et al.* [Yang et al., 2009]. Contrairement à la signature BoW, la signature ScSPM utilise plusieurs mots visuels pour coder un descripteur. Pour cela, les auteurs utilisent une contrainte de parcimonie moins stricte que celle de la signature BoW :

$$\mathbf{c}_{ri}^* = \arg \min_{\mathbf{c}_{ri}} \sum_r \|\mathbf{b}_{ri} - \mathbf{D}\mathbf{c}_{ri}\|^2 + \lambda \|\mathbf{c}_{ri}\|_1 \quad (\text{II.7})$$

Cela permet d'obtenir des codes qui conservent plus d'information qu'une simple quantification.

Par la suite, Wang *et al.* proposent une extension de ce codage : la signature *locality-constrained linear coding* (LLC) [Wang et al., 2010]. Les auteurs proposent d'ajouter une contrainte de localité pour que les descripteurs similaires soient codés par un ensemble de mots visuels similaires :

$$\begin{aligned} \mathbf{c}_{ri}^* = \arg \min_{\mathbf{c}_{ri}} \sum_r \|\mathbf{b}_{ri} - \mathbf{D}\mathbf{c}_{ri}\|^2 + \lambda \|\mathbf{d}_{ri} \odot \mathbf{c}_{ri}\|^2 \\ \text{s.c. } \mathbf{1}^\top \mathbf{c}_{ri} = 1, \forall r; \end{aligned} \quad (\text{II.8})$$

avec $\mathbf{d}_{ri}$ le vecteur des distances entre le descripteur $\mathbf{b}_{ri}$ et les mots visuels et $\odot$ le produit de Hadamard [Horn, 1990]. Cela permet d'obtenir des codes creux similaires quand les descripteurs sont similaires, contrairement à la signature ScSPM qui peut utiliser deux mots visuels différents pour coder deux descripteurs similaires. Ils proposent également différentes fonctions d'agrégation comme l'agrégation par somme ou par maximum. La signature alors obtenue est de dimension $W = K$.

De nombreuses méthodes de codage ont été proposés, notamment pour résoudre le problème de l'ambiguïté des mots visuels [van Gemert et al., 2010]. Par exemple, Gemert *et al.* proposent des méthodes de codage basées sur les dictionnaires de code à noyau [van Gemert et al., 2008, Van Gemert et al., 2010]. Ou encore, Avila *et al.* proposent une méthode de codage basée sur des histogrammes de distance au mot visuel [Avila et al., 2011].

Ces approches donnent de bonnes performances en classification, par exemple, la signature LLC obtient un mAP de 57.6% sur la base VOC 2007 avec 25000 mots visuels et avec le produit scalaire usuel [Chatfield et al., 2011].

II.2.3.c La signature *Fisher Vector*

La signature *Fisher Vector* (FV) a été proposée par Perronnin *et al.* en 2007 [Perronnin and Dance, 2007]. Les deux idées principales de cette méthode sont : la modélisation de l'espace de description ; et la déviation du modèle propre à une image par rapport à ce modèle général.

Perronnin *et al.* proposent pour cela de modéliser par une fonction de densité u_λ la distribution des descripteurs dans l'espace. On peut alors décrire un sac de descripteurs

B_i en calculant le gradient du logarithme de la fonction de vraisemblance des descripteurs du sac B_i :

$$G_\lambda^{B_i} = \frac{1}{|B_i|} \nabla_\lambda \log u_\lambda(B_i). \quad (\text{II.9})$$

Cela permet de mesurer la contribution des descripteurs du sac au modèle u_λ .

Pour mesurer la similarité entre deux sacs (B_i, B_j) , les auteurs proposent alors d'utiliser la fonction noyau suivante :

$$s(B_i, B_j) = G_\lambda^{B_i} F_\lambda^{-1} G_\lambda^{B_j} \quad (\text{II.10})$$

avec $F_\lambda \in \mathcal{M}_{M \times M}$ la matrice d'information de Fisher :

$$F_\lambda = E_{\mathbf{x} \sim u_\lambda} [G_\lambda^{\mathbf{x}} G_\lambda^{\mathbf{x}'}] \quad (\text{II.11})$$

Comme F_λ est symétrique et définie positive, ils proposent de l'intégrer au gradient :

$$\mathcal{G}_\lambda^{B_i} = L_\lambda G_\lambda^{B_i}, \quad (\text{II.12})$$

avec $F_\lambda^{-1} = L_\lambda^\top L_\lambda$ la décomposition de Cholesky de la matrice F_λ^{-1} . La fonction noyau (II.10) peut alors s'écrire comme un simple produit scalaire entre $\mathcal{G}_\lambda^{B_i}$ et $\mathcal{G}_\lambda^{B_j}$. Il est intéressant de noter que la dimension de cette signature dépend linéairement du nombre de paramètres de la fonction de densité.

Perronnin *et al.* proposent d'utiliser un mélange de gaussiennes comme modèle :

$$u_\lambda(\mathbf{x}) = \sum_{c=1}^K w_c f_{\mu_c, \Sigma_c}(\mathbf{x}), \quad (\text{II.13})$$

avec f_{μ_c, Σ_c} la loi normale multidimensionnelle avec les paramètres : μ_c la moyenne ; et Σ_c la matrice de covariance, $\{w_c\}_{c=1}^K$ les pondérations des lois et $\lambda = \{w_c, \mu_c, \Sigma_c, c = 1 \dots K\}$. Les auteurs proposent de considérer les matrices de covariance diagonales : $\sigma_c = \text{diag}(\Sigma_c)$. Pour cette signature, les mots visuels sont les gaussiennes du modèle.

Le gradient du logarithme de la fonction de vraisemblance s'écrit alors :

$$G_\lambda^{B_i} = \frac{1}{|B_i|} \sum_{\mathbf{x} \in B_i} \sum_{c=1}^K \nabla_{\lambda_c} \log u_{\lambda_c}(\mathbf{x}) \quad (\text{II.14})$$

la signature finale prend en compte uniquement les gradients des moyennes et les gradients des écarts types. Les gradients des pondérations des lois ne sont pas utilisés car ils n'apportent pas une information significative [Perronnin and Dance, 2007, Perronnin et al., 2010b].

Les gradients de la moyenne μ et de l'écart type σ pour chaque gaussienne c s'écrivent alors :

$$\mathcal{G}_{\mu, c}^{B_i} = \frac{1}{|B_i| \sqrt{w_c}} \sum_{\mathbf{x} \in B_i} \gamma_c(\mathbf{x}) \left(\frac{\mathbf{x} - \mu_c}{\sigma_c} \right), \quad (\text{II.15})$$

$$\mathcal{G}_{\sigma, c}^{B_i} = \frac{1}{|B_i| \sqrt{w_c}} \sum_{\mathbf{x} \in B_i} \gamma_c(\mathbf{x}) \left[\frac{(\mathbf{x} - \mu_c)^2}{\sigma_c^2} - 1 \right], \quad (\text{II.16})$$

avec $\gamma_c(\mathbf{x})$ la fonction d'appartenance du descripteur $\mathbf{x}$ à la gaussienne c :

$$\gamma_c(\mathbf{x}) = \frac{w_c f_{\mu_c, \sigma_c}(\mathbf{x})}{\sum_{j=1}^K w_j f_{\mu_j, \sigma_j}(\mathbf{x})} \quad (\text{II.17})$$

La signature finale est obtenue par concaténation des vecteurs gradients des différentes gaussiennes, elle a une dimension de $W = 2 \times D \times K$.

Pour déterminer les paramètres du mélange de gaussiennes, les auteurs proposent d'utiliser un algorithme de *Expectation-Maximization* (EM) sur un ensemble de descripteurs locaux extraits d'un grand nombre d'images sélectionnées aléatoirement [Bilmes et al., 1998].

Cette méthode permet d'obtenir de bons résultats en classification et en recherche par similarité avec un petit nombre de mots visuels. Par exemple en classification, elle obtient un mAP de 61.7% sur la base VOC 2007 avec 256 mots visuels et avec le produit scalaire usuel [Chatfield et al., 2011]. En recherche par similarité, elle obtient un mAP de 62.5% sur la base Holidays avec 256 mots visuels et avec le produit scalaire usuel [Jégou et al., 2012].

II.2.3.d La signature VLAD

La signature VLAD (*Vector of Locally Aggregated Descriptors*) est une signature proposée par Jegou *et al.* en 2010 [Jégou et al., 2010]. Cette signature peut être vue comme une approximation de la signature Fisher Vector.

Contrairement à la signature FV, la signature VLAD utilise comme modèle une partition de l'espace des descripteurs en K parties. Chaque mot visuel (partie) c est paramétré par son barycentre μ_c .

Pour obtenir ce modèle, les auteurs proposent d'utiliser un algorithme de type *k-means* [Lloyd, 1982] sur un ensemble de descripteurs locaux extraits d'un grand nombre d'images sélectionnées aléatoirement.

La signature VLAD est basée uniquement sur la déviation aux moyennes du modèle des descripteurs du sac. Pour construire la signature, chaque descripteur $\mathbf{x}_{ij}$ du sac $\mathbf{B}_i$ est affecté au mot visuel le plus proche. Le sac de descripteurs est alors divisé dans un ensemble de sous-sacs disjoints $\mathbf{B}_{ic}$:

$$\mathbf{B}_{ic} = \{\mathbf{x}_{ij} \in \mathbf{B}_i \mid \text{PPV}(\mathbf{x}_{ij}) = c\}, \quad (\text{II.18})$$

avec $\text{PPV}(\cdot)$ la fonction qui retourne la partie correspondant au mot visuel le plus proche. Contrairement à la signature Fisher Vector, l'affectation des descripteurs est obtenue par l'utilisation de la distance euclidienne et chaque descripteur est affecté à une seule partition, cela permet de fortement réduire les coûts de calcul de la signature.

Pour calculer la signature, les auteurs proposent de cumuler pour chaque mot visuel les déviations au barycentre des descripteurs de la partie :

$$\nu_{ci} = \sum_{\mathbf{x} \in \mathbf{B}_{ci}} (\mathbf{x} - \mu_c). \quad (\text{II.19})$$

La signature finale est obtenue par la concaténation de l'ensemble des mesures de déviation de chaque partie. La dimension de la signature finale est alors $W = K \times D$

Cette signature est simple à mettre en œuvre et donne de bonnes performances en classification et recherche par similarité. Par exemple en classification, elle obtient un mAP de 58.3% sur la base VOC 2007 avec 256 mots visuels et avec le produit scalaire usuel [Peng et al., 2014]. En recherche par similarité, elle obtient un mAP de 62.2% sur la base Holidays avec 64 mots visuels et avec le produit scalaire usuel [Delhumeau et al., 2013].

II.2.3.e La signature VLAT

La signature VLAT (*Vectors of Locally Aggregated Tensors*) est une signature proposée par Picard *et al.* en 2011 [Picard and Gosselin, 2011]. Cette signature est une approximation du noyau sur sac Gaussien :

$$S(B_i, B_j) = \sum_{\mathbf{x} \in B_i} \sum_{\mathbf{y} \in B_j} e^{-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{\sigma^2}} \quad (\text{II.20})$$

Elle permet de supprimer les coûts calculatoires engendrés par les comparaisons deux à deux des descripteurs. Pour cela, les auteurs proposent de construire une fonction d'injection :

$$\begin{aligned} \mathcal{B} &\rightarrow \mathbb{R}^W \\ B_i &\mapsto \Phi(B_i) \end{aligned} \quad (\text{II.21})$$

tel que le produit scalaire dans l'espace induit approxime le noyau sur sac Gaussien :

$$S(B_i, B_j) \approx \Phi(B_i)^\top \Phi(B_j). \quad (\text{II.22})$$

Le noyau sur sac Gaussien donne de très bonnes performances car l'utilisation de la fonction Gaussienne permet d'être fortement discriminant sur l'appariement des descripteurs. Un pré-traitement usuel des descripteurs consiste à les normaliser.

La norme ℓ_2 peut être développée et la fonction noyau s'écrit alors :

$$S(B_i, B_j) = e^{-\frac{2}{\sigma^2}} \sum_{\mathbf{x} \in B_i} \sum_{\mathbf{y} \in B_j} e^{\frac{2\mathbf{x}^\top \mathbf{y}}{\sigma^2}} \quad (\text{II.23})$$

Pour obtenir la fonction d'injection, les auteurs effectuent un développement en série de Taylor de la fonction noyau :

$$S(B_i, B_j) = \sum_{\mathbf{x} \in B_i} \sum_{\mathbf{y} \in B_j} \sum_{p=0}^{\infty} \alpha_p (\mathbf{x}^\top \mathbf{y})^p \quad (\text{II.24})$$

avec $\alpha_p = e^{-\frac{2}{\sigma^2}} \frac{2^p}{\sigma^{2p} p!}$. Cette expression est très intéressante car elle est sous la forme de somme de produits scalaires à la puissance p . Le produit scalaire à la puissance p est une fonction noyau bien connue et elle est équivalente au produit scalaire des tenseurs d'ordre p des variables :

$$(\mathbf{x}^\top \mathbf{y})^p = \text{vec}(\otimes_p \mathbf{x})^\top \text{vec}(\otimes_p \mathbf{y}) \quad (\text{II.25})$$

avec $\text{vec}(\cdot)$ la fonction qui convertit un tenseur en vecteur et $\otimes_p \mathbf{x}$ le produit tensoriel d'ordre p du vecteur $\mathbf{x}$. Cette propriété permet de décomposer les deux sommes imbriquées en deux deux sommes distinctes :

$$S(B_i, B_j) = \sum_{p=0}^{\infty} \alpha_p \Phi_p(B_i)^\top \Phi_p(B_j), \quad (\text{II.26})$$

avec $\Phi_p(\cdot)$ la fonction d'injection pour le produit tensoriel d'ordre p .

La fonction d'injection finale est la concaténation pondérée des sous-fonctions d'injection $\Phi_p(B_i)$ pour $p \in \mathbb{N}^*$. Cette fonction d'injection permet d'obtenir les mêmes résultats

que la fonction noyau d'origine sans le calcul de l'ensemble des combinaisons des descripteurs des deux sacs. Malheureusement, la dimension de l'espace d'injection est infinie. Les auteurs proposent alors de restreindre le développement en série de Taylor à l'ordre deux. La fonction d'injection s'écrit alors :

$$\Phi(B_i) = [\sqrt{\alpha_1}\Phi_1(B_i)^\top \sqrt{\alpha_2}\Phi_2(B_i)^\top]^\top. \quad (\text{II.27})$$

La dimension de l'espace d'injection est alors $D + D^2$.

Pour améliorer la capacité de discrimination de cette signature, ils proposent deux améliorations : (i) centrer les données ; (ii) utiliser une partition de l'espace des descripteurs.

Le centrage des données permet d'augmenter la dynamique du produit scalaire. En effet, si les données sont décentrées le produit scalaire n'a pas une dynamique maximale. Quand les données sont centrées, la dynamique angulaire est alors maximale. La sous-fonction d'injection devient alors :

$$\Phi_p(B_i) = \text{vec}(\sum_{\mathbf{x} \in B_i} \otimes_p(\mathbf{x} - \mu) - \mathcal{T}_p), \quad (\text{II.28})$$

Avec μ le barycentre des descripteurs et $\mathcal{T}_p$ le barycentre du tenseur d'ordre p .

Le partitionnement de l'espace des descripteurs est une méthode souvent employée dans les fonctions noyau sur sac. Elle consiste à construire une partition de l'espace des descripteurs en K parties et à calculer pour chaque partie la fonction noyau avec uniquement les descripteurs appartenant à cette partie. Le noyau final est la somme des noyaux dans chaque partie c :

$$S'(B_i, B_j) = \sum_{c \in \mathcal{D}} S(B_{ic}, B_{jc}) \quad (\text{II.29})$$

avec $B_{ic} = \{\mathbf{x} \in B_i | \text{PPV}_{\mathcal{D}}(\mathbf{x}) = c\}$. Cette amélioration permet de réduire considérablement les coûts calculatoires. En effet, elle réduit fortement le nombre de descripteurs dans les sous-sacs B_{ic} et donc le nombre de combinaisons de descripteurs. De plus, elle permet d'éviter les comparaisons entre les descripteurs trop dissimilaires. La sous-fonction d'injection devient alors :

$$\Phi_p^{\mathcal{D}}(B_i) = [\Phi_p(B_{i1})^\top \dots \Phi_p(B_{ic})^\top \dots \Phi_p(B_{iK})^\top]^\top \quad (\text{II.30})$$

Ces améliorations permettent de bien conserver la capacité de discrimination du noyau sur sac Gaussien malgré l'approximation du développement de Taylor d'ordre 2.

Les tenseurs d'ordre 1 apportent peu d'information en comparaison aux tenseurs d'ordre 2, car ils sont de très petites dimensions par rapport aux tenseurs d'ordre 2. Les auteurs ont donc proposées de ne pas les conserver. La signature finale s'écrit alors :

$$\mathbf{y}_i = \Phi_2^{\mathcal{D}}(B_i). \quad (\text{II.31})$$

La dimension de la signature VLAT est alors égale à $W = K \times D^2$. Cependant, le produit tensoriel des descripteurs locaux crée des tenseurs symétriques, et il n'est pas utile de stocker toutes les valeurs. En tenant compte de ces symétries, la dimension de la signature VLAT est alors de $W = K \times \frac{D \times (D+1)}{2}$.

Cette signature est simple à mettre en œuvre et donne de bonnes performances en classification et recherche par similarité. Par exemple en classification, elle obtient un

mAP de 60.8% sur la base VOC 2007 avec 256 mots visuels et avec le produit scalaire usuel [Picard and Gosselin, 2013]. En recherche par similarité, elle obtient un mAP de 64.0% sur la base Holidays avec 64 mots visuels et avec le produit scalaire usuel [Negrel et al., 2013].

II.2.4 Pré-traitements et Post-traitements

Dans cette section, nous décrivons quelques méthodes de pré-traitement des descripteurs locaux et de post-traitement sur les signatures. Ces traitements permettent d'améliorer les capacités de discrimination des descripteurs locaux [Safadi and Quénot, 2013] et des signatures.

II.2.4.a Pré-traitement

Le pré-traitement de plus courant sur les descripteurs locaux est la normalisation ℓ_2 des descripteurs [Mikolajczyk and Schmid, 2005, Delhumeau et al., 2013] :

$$\mathbf{x}' = \frac{\mathbf{x}}{\|\mathbf{x}\|_2}. \quad (\text{II.32})$$

Ceci permet d'obtenir une bonne performance avec les noyaux sur sac qui utilisent un produit scalaire comme fonction de similarité.

L'un des pré-traitement couramment utilisé est d'effectuer une analyse en composantes principales (ACP) sur les descripteurs locaux [Mikolajczyk and Schmid, 2005, Ke and Sukthankar, 2004, Picard and Gosselin, 2013, Perronnin et al., 2010b]. Pour cela, nous calculons la décomposition en valeurs et vecteurs propres de la matrice de covariance des descripteurs Σ :

$$\Sigma = \mathbf{P} \mathbf{D} \mathbf{P}^\top, \quad (\text{II.33})$$

avec $\mathbf{P}$ la matrice des vecteurs propres et $\mathbf{D}$ la matrice diagonale des valeurs propres. Nous projetons alors les descripteurs locaux dans le sous-espace propre de dimension D' :

$$\mathbf{x}' = \mathbf{P}_{D'}^\top (\mathbf{x} - \boldsymbol{\mu}), \quad (\text{II.34})$$

avec $\mathbf{P}_{D'}$ la matrice des D' vecteurs propres correspondant aux D' plus grandes valeurs propres et $\boldsymbol{\mu}$ la moyenne des descripteurs. Ce pré-traitement a deux avantages majeurs : il permet de supprimer les corrélations entre les composantes des descripteurs locaux ; et il permet de réduire la dimension des descripteurs locaux.

II.2.4.b Post-traitement

Il existe deux post-traitements effectués sur les signatures : la normalisation en loi de puissance ; et la normalisation ℓ_2 . Elles sont généralement appliquées successivement à la signature.

La normalisation en loi de puissance consiste à mettre à la puissance α chaque composante de la signature en conservant son signe [Perronnin et al., 2010b, Delhumeau et al., 2013, Picard and Gosselin, 2013, Perronnin et al., 2010b] :

$$\forall j, \mathbf{x}'_i[j] = \text{sign}(\mathbf{x}_i[j]) |\mathbf{x}_i[j]|^\alpha, \quad (\text{II.35})$$

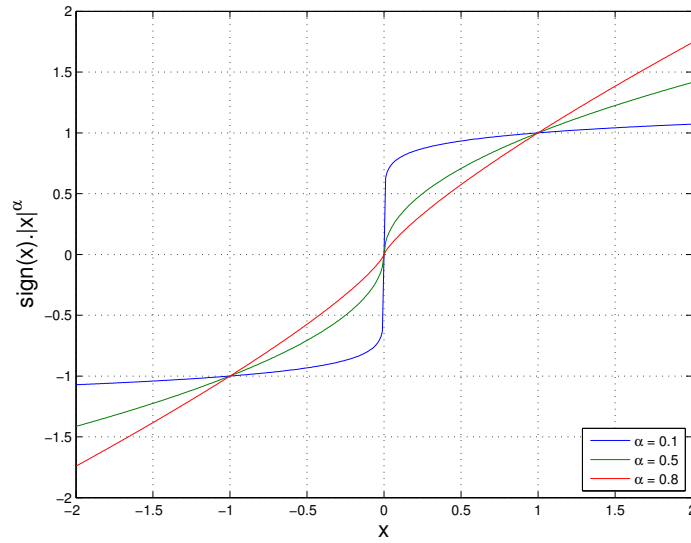


FIGURE II.3 – Effet de la normalisation en loi de puissance sur la dynamique des valeurs d’une signature.

avec $\alpha \in]0, 1]$. Cette méthode permet d’uniformiser l’importance de chaque dimension dans la signature. Plus α est petit plus les valeurs de la signature se regroupent en -1 et 1 (Figure II.3). Les expériences ont montré que plus une signature est de grande dimension, plus une normalisation en loi de puissance petite est appropriée [Perronnin et al., 2010a]. Nous remarquons que ce traitement est analogue à la fonction sigmoïde utilisé dans les réseaux de neurones [Bishop et al., 1995].

Quand le produit scalaire est utilisé pour calculer un score de similarité entre deux signatures, il est intéressant d’effectuer une normaliser ℓ_2 des signatures [Perronnin et al., 2010b, Delhumeau et al., 2013, Picard and Gosselin, 2013, Jégou et al., 2010]. En effet, cela permet d’obtenir deux propriétés intéressantes : la similarité entre une signature et elle même est égale à 1 ; et la similarité entre une signature et une signature différente est strictement inférieure à 1 . Dans le cas de la recherche par similarité, cela permet de garantir que l’image la plus similaire à la requête est elle même.

Méthode proposée

Chapitre III

Dans ce chapitre, nous présentons une nouvelle signature d'image obtenue par agrégation des descripteurs locaux. Notre signature est basée sur la signature VLAT qui est une approximation du noyau sur sac Gaussien. Dans la première partie de ce chapitre, nous montrons que la signature VLAT peut être vue comme une signature à déviation de modèle. Nous expliciterons le modèle utilisé ainsi que la mesure de déviation. Dans la deuxième partie, nous présenterons une optimisation du modèle utilisé dans la signature VLAT afin d'augmenter son pouvoir de discrimination, tout en réduisant la dimension de la signature obtenue. Dans la troisième partie, nous proposons deux méthodes pour réduire la dimension de la signature. Dans la dernière partie, nous effectuons un ensemble d'expériences pour évaluer les performances de cette signature.

III.1 Vision déviation de modèle de la signature VLAT

Dans la section de l'état de l'art II.2.3.e, nous avons présenté la signature VLAT. Cette signature est obtenue par l'approximation de la fonction d'injection associée au noyau sur sac Gaussien. Cependant, cette signature peut également être vue comme une signature à déviation de modèle. Comme nous l'avons vu avec les signatures Fisher Vector et VLAD, les signatures à déviation de modèle utilisent une modélisation de l'espace des descripteurs locaux et la signature est construite en calculant la déviation entre les descripteurs d'une image et le modèle.

Pour la signature VLAT le modèle est constitué d'une partition de l'espace des descripteurs et chaque partie c est caractérisée par la moyenne μ_c et la matrice de covariance $\mathcal{T}_c$ des descripteurs de la région.

Pour estimer le modèle, nous utilisons un ensemble B de descripteurs locaux extraits d'un grand nombre d'images. L'ensemble des descripteurs est partitionné en K parties à l'aide d'un algorithme de type k-means. Nous estimons alors les paramètres du modèle à l'aide des estimateurs empiriques de la moyenne et de la matrice de covariance :

$$\mu_c = \frac{1}{|B_c|} \sum_{\mathbf{x} \in B_c} \mathbf{x} \quad (\text{III.1})$$

$$\mathcal{T}_c = \frac{1}{|B_c|} \sum_{\mathbf{x} \in B_c} (\mathbf{x} - \boldsymbol{\mu}_c)(\mathbf{x} - \boldsymbol{\mu}_c)^\top, \quad (\text{III.2})$$

avec B_c l'ensemble des descripteurs de la partie c et $|B_c|$ le nombre de descripteurs de l'ensemble B_c .

Pour mesurer la déviation du sac de descripteur B_i de image i au modèle dans la signature VLAT, nous affectons chaque descripteur à la partie la plus proche, nous obtenons alors K sous ensembles B_{ci} du sac B_i . La mesure de déviation se fait séparément dans chaque partie, nous calculons la somme des écarts de chaque descripteurs au modèle :

$$\mathcal{T}_{ic} = \sum_{\mathbf{x} \in B_{ci}} [(\mathbf{x} - \boldsymbol{\mu}_c)(\mathbf{x} - \boldsymbol{\mu}_c)^\top - \mathcal{T}_c]. \quad (\text{III.3})$$

Sous cette forme, nous voyons que la signature VLAT est à mi-chemin entre la signature VLAD et la signature Fisher Vector. De même que la signature VLAD, la signature VLAT utilise un modèle obtenu par une partition de l'espace des descripteurs. Ce type de modèle permet d'avoir un faible coût calculatoire lors de l'affectation des descripteurs aux parties du modèle. De même que la signature Fisher Vector, la signature VLAT repose sur le moment statistique d'ordre 2. Cela permet d'obtenir une description plus fine des sacs de descripteurs.

Notons que la signature VLAT utilise la totalité du moment statistique d'ordre 2, la signature est alors de beaucoup plus grande dimension que les signatures Fisher Vector et VLAD.

III.2 Optimisation du modèle

Dans cette section, nous proposons une optimisation du modèle des signatures VLAT afin d'améliorer la discrimination de la signature tout en réduisant la dimension de la signature finale. Comme nous l'avons vu dans l'état de l'art, il est courant d'effectuer une analyse en composantes principales (ACP) en pré-traitement des descripteurs locaux, ce qui permet de réduire leur dimension et de supprimer les corrélations entre les composantes des descripteurs.

III.2.1 ACP dans les parties du modèle

Pour améliorer le rapport débit/distorsion de l'étape de réduction de dimension, nous proposons d'intégrer l'ACP dans le modèle. En effet, l'espace des descripteurs est complexe, il est donc intéressant d'effectuer en premier une segmentation de l'espace en parties plus simples puis d'effectuer une ACP dans chaque partie. Cela permet d'avoir plus de contrôle lors de la réduction de dimension sur les informations à supprimer.

En premier, nous effectuons une décomposition en valeurs et vecteurs propres de la matrice de covariance de chaque partie :

$$\mathcal{T}_c = \mathbf{V}_c \mathbf{D}_c \mathbf{V}_c^\top, \quad (\text{III.4})$$

avec $\mathbf{V}_c$ la matrice des vecteurs propres et $\mathbf{D}_c$ la matrice diagonale des valeurs propres de $\mathcal{T}_c$. Pour chaque partie c , nous conservons que les D'_c vecteurs propres correspondants aux plus grandes valeurs propres :

$$\mathcal{T}_{cD'_c} = \mathbf{V}_{cD'_c} \mathbf{D}_{cD'_c} \mathbf{V}_{cD'_c}^\top, \quad (\text{III.5})$$

avec $\mathcal{T}_{cD'_c}$ l'approximation de la matrice de covariance $\mathcal{T}_c$, $\mathbf{D}_{cD'_c}$ la matrice diagonale des D'_c plus grandes valeurs propres et $\mathbf{V}_{cD'_c}$ la matrice des D'_c vecteurs propres correspondants.

Nous projetons tous les descripteurs d'une partie dans son sous-espace propre :

$$\mathbf{B}'_{cD'_ci} = \left\{ \mathbf{V}_{cD'_c}^\top (\mathbf{x} - \boldsymbol{\mu}_c) \right\}_{\mathbf{x} \in \mathbf{B}_{ci}}. \quad (\text{III.6})$$

Les matrices de covariance des descripteurs locaux dans chaque partie sont alors des matrices diagonales et la mesure de déviation s'écrit :

$$\mathcal{T}'_{iD'_c} = \sum_{\mathbf{x}' \in B'_{cD'_ci}} \mathbf{x}' \mathbf{x}'^\top - \mathbf{D}_{cD'_c} \quad (\text{III.7})$$

$$= \sum_{\mathbf{x} \in B_{ci}} \mathbf{V}_{cD'_c}^\top (\mathbf{x} - \boldsymbol{\mu}_c) (\mathbf{x} - \boldsymbol{\mu}_c)^\top \mathbf{V}_{cD'_c} - \mathbf{D}_{cD'_c} \quad (\text{III.8})$$

$$= \sum_{\mathbf{x} \in B_{ci}} \mathbf{V}_{cD'_c}^\top (\mathbf{x} - \boldsymbol{\mu}_c) (\mathbf{x} - \boldsymbol{\mu}_c)^\top \mathbf{V}_{cD'_c} - \mathbf{V}_{cD'_c}^\top \mathcal{T}_c \mathbf{V}_{cD'_c} \quad (\text{III.9})$$

$$= \mathbf{V}_{cD'_c}^\top \left(\sum_{\mathbf{x} \in B_{ci}} (\mathbf{x} - \boldsymbol{\mu}_c) (\mathbf{x} - \boldsymbol{\mu}_c)^\top - \mathcal{T}_c \right) \mathbf{V}_{cD'_c} \quad (\text{III.10})$$

$$= \mathbf{V}_{cD'_c}^\top \mathcal{T}_{ic} \mathbf{V}_{cD'_c} \quad (\text{III.11})$$

Avec cette mesure de déviation, le coût de stockage du modèle est plus faible. Pour la signature VLAT, nous devons stocker pour chaque partie c son barycentre $\boldsymbol{\mu}_c$ de dimension D et la matrice de covariance $\mathcal{T}_c$ de dimension $D \times D$. Le coût total de stockage du modèle est alors de $K \times (D + D \times D)$. Avec notre optimisation, nous devons toujours stocker le barycentre des parties mais nous ne stockons plus la totalité de leur matrices de covariance. Nous ne stockons pour chaque partie c que D'_c vecteurs propres de dimension D ainsi que

D'_c valeurs propres. Le coût de stockage du modèle optimisé est alors de $\sum_{c=1}^K D + D \times D'_c + D'_c$.

Pour $D'_c < D$ le coût de stockage du modèle optimisé est plus faible que le modèle non-optimisé.

Cette optimisation du modèle permet également de réduire la dimension de la signature. Avec la signature VLAT la dimension de la signature finale est de $K \times \frac{D \times (D + 1)}{2}$, alors qu'avec le modèle optimisé la dimension de la signature est alors de

$$\sum_{c=1}^K \frac{D'_c \times (D'_c + 1)}{2}.$$

Nous proposons d'ajouter au modèle une étape de blanchiment des descripteurs dans chaque partie (variance des descripteurs égale à 1), ce qui correspond à :

$$\bar{\mathbf{B}}'_{cD'_ci} = \left\{ \mathbf{D}_{cD'_c}^{-1/2} \mathbf{x}' \right\}_{\mathbf{x}' \in \mathbf{B}'_{cD'_ci}} \quad (\text{III.12})$$

$$= \left\{ \mathbf{D}_{cD'_c}^{-1/2} \mathbf{V}_{cD'_c}^\top (\mathbf{x} - \boldsymbol{\mu}_c) \right\}_{\mathbf{x} \in \mathbf{B}_{ci}} \quad (\text{III.13})$$

$$= \left\{ \mathbf{P}_{cD'_c}^\top (\mathbf{x} - \boldsymbol{\mu}_c) \right\}_{\mathbf{x} \in \mathbf{B}_{ci}} \quad (\text{III.14})$$

avec $\mathbf{P}_{cD'_c} = \mathbf{V}_{cD'_c} \mathbf{D}_{cD'_c}^{-1/2}$. Avec cette étape de blanchiment, la mesure de déviation s'ex-

prime comme suit :

$$\bar{\mathcal{T}}'_{iD'_c} = \sum_{\mathbf{x}' \in \bar{\mathbf{B}}'} \mathbf{x}' \mathbf{x}'^\top - \mathbf{I} \quad (\text{III.15})$$

$$= \sum_{\mathbf{x} \in B_{ci}} \mathbf{P}_{cD'_c}^\top (\mathbf{x} - \boldsymbol{\mu}_c)(\mathbf{x} - \boldsymbol{\mu}_c)^\top \mathbf{P}_{cD'_c} - \mathbf{I} \quad (\text{III.16})$$

Le coût de stockage du modèle est alors de $\sum_{c=1}^K D + D \times D'_c$.

III.2.2 Réduction de dimension

Cette optimisation permet de supprimer les corrélations entre les composantes des descripteurs dans chaque partie du modèle de la signature VLAT. Nous avons également vu qu'il est possible de réduire la dimension de la signature finale en conservant un nombre réduit de composantes dans les parties. Nous proposons deux méthodes pour déterminer le nombre de composantes à conserver dans chaque partie : une méthode locale et une méthode globale.

La méthode locale consiste à fixer arbitrairement un nombre identique de composantes conservées dans chaque partie du modèle :

$$D'_1 = \dots = D'_c = \dots = D'_K = D'. \quad (\text{III.17})$$

La méthode globale consiste à conserver le même taux d'énergie dans chaque partie du modèle. L'énergie d'une partie est définie par la somme des valeurs propres de la matrice de covariance $\mathcal{T}_c$:

$$e_c = \sum_{i=1}^D \lambda_{ci} \quad (\text{III.18})$$

avec $[\lambda_{c1} \dots \lambda_{cD}]^\top = \text{diag}(\mathbf{D}_c)$ et $\lambda_{ci} \geq \lambda_{ci+1}$. Le taux de conservation d'énergie est alors défini par :

$$\delta_c(D'_c) = \frac{\sum_{i=1}^{D'_c} \lambda_{ci}}{e_c} \quad (\text{III.19})$$

Pour avoir le même ratio de conservation d'énergie dans chaque partie du modèle, nous résolvons le problème suivant :

$$D'^*_c = \arg \min D'_c \text{ s.c. } \delta_c(D'_c) \geq \delta_{global}, \quad (\text{III.20})$$

avec δ_{global} le ratio de conservation d'énergie global.

III.3 Expériences

Dans cette section, nous présentons les résultats obtenus avec la signature VLAT optimisée. Nous effectuons des expérimentations en recherche par similarité et en classification automatique. Les expérimentations en recherche par similarité sont effectuées sur trois bases d'image bien connues : Holidays [Jégou et al., 2008], Oxford [Philbin et al., 2007] et Paris [Philbin et al., 2008]. Les expérimentations en classification sont menées sur la base d'images VOC 2007 [Everingham et al.,].

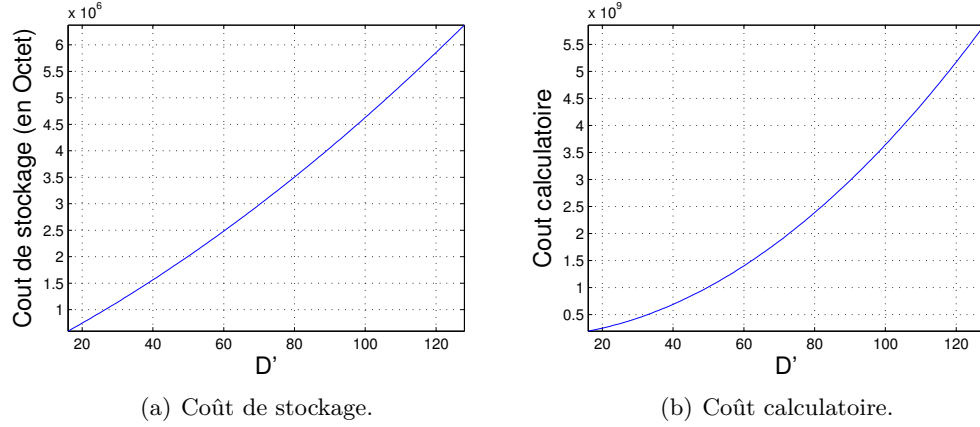


FIGURE III.1 – Coût de stockage et coût calculatoire en fonction du nombre de dimensions moyennes conservées par partie (signature VLAT avec un modèle optimisé de $K = 64$ partie pour un sac de $N = 1000$ descripteurs locaux de dimension $D = 128$).

Nous présentons les effets de la normalisation en loi de puissance, de la normalisation ℓ_2 , ainsi que des effets de l'optimisation du modèle VLAT en fonction des stratégies de réduction de dimension et les effets du blanchiment. Nous comparons ces résultats avec les méthodes de l'état de l'art.

III.3.1 Étude des coûts

Dans cette section, nous étudions les coûts engendrés par l'extraction d'une signature VLAT avec un modèle optimisé et les coûts engendrés par le calcul de la fonction de similarité entre deux signatures.

III.3.1.a Coût d'extraction

Nous considérons en tant que coûts d'extraction l'ensemble des coûts nécessaires pour obtenir la signature à partir du sac de descripteurs. Nous ne prenons pas en compte le coût d'extraction des descripteurs locaux car il ne dépend pas de la méthode d'agrégation des descripteurs locaux. Nous considérons un sac de N descripteurs locaux de dimension D et un modèle de K partitions dans lequel nous conservons en moyenne D' dimensions par partie.

Le coût de stockage lors du calcul de la signature provient du stockage du modèle et du stockage de la signature. Pour le modèle optimisé, il est nécessaire de stocker pour chaque partie du modèle : son vecteur de barycentre ; les D' vecteurs propres ; et les D' plus grandes valeurs de la diagonale de la matrice de covariance de la partie. Le coût de stockage du modèle optimisé est alors de $K \times (D + D' \times D + D')$.

Pour le modèle optimisé blanchi, il n'est plus nécessaire de stocker la diagonale de la matrice de covariance de chaque partie car ce sont des matrices identités. Le coût de stockage du modèle optimisé blanchi est alors de $K \times (D + D' \times D)$.

Pour les deux cas, le coût de stockage de la signature est de $K \times (D' \times D')$. En tenant compte des symétries des matrices de déviation, le coût de stockage de la signature peut être réduit à $K \times \frac{(D' \times (D' + 1))}{2}$.

Le coût calculatoire de la signature VLAT dépend linéairement du nombre de descripteurs locaux dans le sac. Pour chaque descripteur, nous devons déterminer à quelle partie il appartient. Le coût calculatoire de cette étape est au pire de $\mathcal{O}(K \times D)$. Puis nous centrons et projetons le descripteur dans le sous espace défini par la ACP de la partie. Le coût calculatoire de cette étape est de $\mathcal{O}(D + D \times D')$. Puis nous calculons le tenseur d'ordre 2 du descripteur et nous centrons. Le coût calculatoire de ces étapes est de $\mathcal{O}(D' + D' \times D')$. Puis nous agrégeons le tenseur d'ordre 2 dans la signature finale. Le coût calculatoire de cette étape est de $\mathcal{O}(D' \times D')$. L'étape finale de calcul d'une signature est l'étape de post-traitement de la signature (normalisation en loi de puissance et normalisation ℓ_2). Le coût calculatoire de cette étape est de $\mathcal{O}(K \times \frac{D' \times (D' + 1)}{2})$.

Le coût calculatoire total est alors de $\mathcal{O}((K + 1) \times D + (D + 1) \times D' + 2 \times D' \times D' + K \times \frac{D' \times (D' + 1)}{2})$.

Pour étudier l'influence de l'optimisation du modèle sur le coût d'extraction, nous prenons le jeu de paramètres suivant $N = 10000$, $K = 64$ et $D = 128$. La Figure III.1(a) montre l'évolution du coût de stockage en fonction du nombre moyen D' de vecteurs propres conservés par partie. La figure III.1(b) montre l'évolution du coût calculatoire en fonction de D' . Pour $D' = D$, le coût de stockage et le coût calculatoire sont presque les mêmes entre le modèle optimisé et le modèle standard. De plus, pour des valeurs de D' plus petites, le coût de stockage et le coût calculatoire diminuent fortement.

III.3.1.b Coût d'utilisation

Nous considérons en tant que coût d'utilisation l'ensemble des coûts nécessaires pour obtenir l'ensemble des scores de similarités entre une signature et toutes les signatures de la base. Nous considérons une base de T signatures de dimension L .

Le coût de stockage est directement dû au stockage de la base de signatures dans la mémoire. Le coût de stockage est alors de $T \times L$.

Pour obtenir la similarité entre deux signatures, nous utilisons le produit scalaire usuel. Le coût calculatoire d'un produit scalaire est de $\mathcal{O}(L)$. Le coût calculatoire total est alors de $\mathcal{O}(T \times L)$.

III.3.2 Recherche par similarité

Dans cette section nous présentons le protocole expérimental utilisé pour évaluer les performances obtenues avec le modèle optimisé de la signature VLAT en recherche par similarité.

III.3.2.a Protocole expérimental

Pour évaluer les performances de la signature VLAT optimisée en recherche par similarité, nous utilisons la base d'image Holidays (Annexe A.3.1). Nous utilisons les descripteurs locaux SIFT fournis avec la base d'image Holidays. Ces descripteurs SIFT ont été extraits à partir de points d'intérêts obtenus avec le détecteur SIFT. Nous effectuons une normalisation ℓ_2 des descripteurs locaux.

Pour calculer les paramètres du modèle, nous utilisons l'ensemble des descripteurs SIFT extraits de 10000 images de la base Flickr60k (Annexe A.1.1). Le partitionnement

Signature		Mots visuels	Dim.	mAP (en %)
BoW	[Jégou et al., 2010]	20	20k	40.4
FV	[Jégou et al., 2010]	64	8192	49.5
VLAD	[Jégou et al., 2010]	64	8192	52.6
	[Douze et al., 2011]	-	6755	69.9
VLAT		32	264k	60,1
VLAT		64	520k	64,0
VLAT		128	1M	67,9

TABLE III.1 – Performances en recherche par similarité des signatures de l'état de l'art sur la base Holidays

des données est effectué à l'aide d'un algorithme de type k-means. Puis nous estimons la moyenne et la variance dans chaque partie.

Pour calculer une similarité entre deux descripteurs, nous effectuons deux pré-traitements sur les signatures : une normalisation en loi de puissance ; puis une normalisation ℓ_2 . Puis nous utilisons le produit scalaire usuel comme fonction de similarité.

III.3.2.b Résultats de références

Pour évaluer les gains obtenus par l'utilisation du modèle optimisé, nous utilisons plusieurs résultats obtenus avec des signatures de l'état de l'art.

Nous reportons dans le Tableau III.1 plusieurs résultats obtenus par des signatures de l'état de l'art, ainsi que des résultats obtenus par nos réimplantations des signatures de l'état de l'art. La première colonne du tableau est le nom de la signature (FV : *Fisher Vector*, BoW : *Bag of Words*, VLAD : *Vector of Locally Aggregated Descriptors*, VLAT : *Vectors of Locally Aggregated Tensors*). La deuxième colonne est le nombre de partitions utilisées pour calculer la signature. La troisième colonne est la dimension de la signature finale. La quatrième colonne est la performance de la signature en mAP (*mean Average Precision*). Le tableau est divisé en deux parties, la partie du haut correspond à des résultats reportés de l'état de l'art. La partie du bas correspond aux résultats obtenus par notre implantation de la signature VLAT.

Nous notons également que plus le partitionnement de l'espace est fin, plus la signature VLAT obtient de bonnes performances. Les meilleures performances sont obtenues avec la signature proposée par Douze *et al.* [Douze et al., 2011], cette signature est obtenue par la combinaison d'une signature FV et d'attribut sémantique nécessitant un apprentissage supervisé.

III.3.2.c Signature VLAT optimisée

Pour évaluer les gains obtenus avec la signature VLAT optimisée, nous calculons la signature pour des modèles avec 32, 64 et 128 parties. Nous nous intéressons à deux paramètres : la dimension de la signature obtenue et ses performances en recherche par similarité sur la base Holidays. Nous évaluons les deux méthodes de sélection des dimensions dans les parties du modèle : la méthode locale et la méthode globale (Section III.2.2). Pour la méthode locale, nous calculons les signatures en conservant : 16, 32, 48, 64, 80, 96, 112 et 128 dimensions par partie. Pour la méthode globale, nous calculons les signatures en conservant : 20%, 40%, 60%, 80%, 85%, 90%, 95% et 100% de l'énergie dans chaque

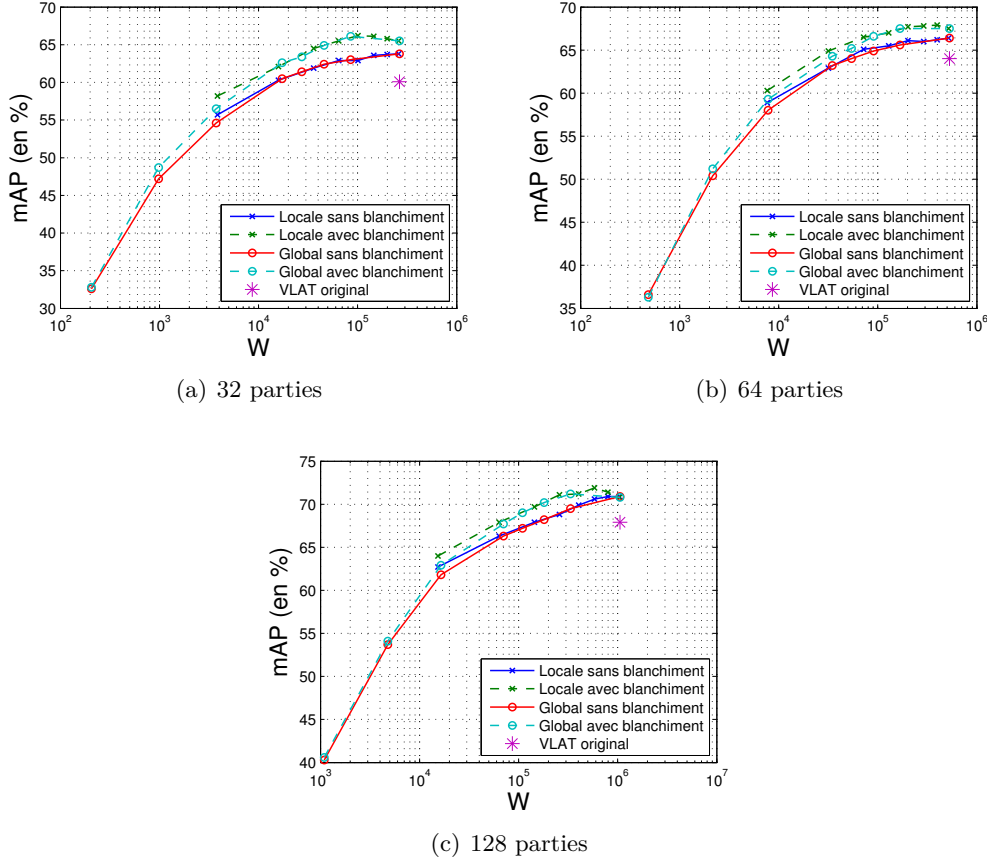


FIGURE III.2 – Performances (mAP) de la signature VLAT optimisée sur Holidays en fonction de sa dimension W .

partie. Nous évaluons également l'effet du blanchiment des descripteurs dans les parties du modèle.

Les trois graphiques de la Figure III.2 montrent l'évolution du mAP en fonction de la dimension pour des signatures VLAT optimisées calculées avec des partitions de 32, 64 et 128 parties. Dans chaque graphique, l'étoile correspond au mAP obtenu avec la signature VLAT non optimisée. Les courbes en trait continu sont les signatures sans blanchiment. Les courbes en pointillés sont les signatures avec blanchiment. Les courbes avec des marqueurs en forme de croix sont les signatures obtenues avec la méthode locale de sélection des dimensions. Les courbes avec des marqueurs en forme de rond sont les signatures obtenues avec la méthode globale de sélection des dimensions.

Nous observons que la signature VLAT optimisée a le même comportement quelque soit le nombre de parties dans le modèle. Nous voyons qu'à dimension équivalente la signature VLAT optimisée obtient de bien meilleures performances que la signature VLAT non-optimisée. Par exemple pour la signature avec 64 parties, nous obtenons un gain de 3.5% de mAP avec la signature VLAT optimisée blanchie. Nous constatons que les deux méthodes de sélection donnent des performances très similaires. Nous observons que la signature VLAT optimisée avec le blanchiment obtient de meilleures performances que sans le blanchiment. Il est très intéressant de noter qu'avec le blanchiment, il est préférable de ne

pas conserver la totalité des dimensions. Par exemple avec la méthode de sélection globale avec blanchiment avec une signature avec 32 parties, nous obtenons un mAP de 66.1% en conservant 95% des dimensions ($W = 85k$) par rapport à 65.5% de mAP en conservant 100% des dimensions ($W = 264k$). Notons également que l'on peut fortement réduire la dimension de la signature tout en conservant de très bonnes performances. Par exemple avec la méthode de sélection globale avec blanchiment pour une signature avec 64 parties, nous pouvons réduire la signature de $W = 398k$ dimensions à $W = 129k$ dimensions avec seulement une perte de 0.9% de mAP.

En comparant les performances obtenues avec la signature VLAT optimisée et les signatures de l'état de l'art (Tableau III.1), nous constatons qu'à dimension équivalente, nous obtenons de meilleures performances. Par exemple, la signature VLAD de $W = 8192$ dimensions obtient un mAP de 52.6%, la signature VLAT optimisée de $W = 7816$ dimensions obtient un mAP de 59.3%. Nous observons également qu'avec un nombre identique de parties dans les modèles, la signature VLAT optimisée obtient de meilleures performances que les signatures de l'état de l'art.

III.3.3 Classification automatique

Dans cette section nous présentons le protocole expérimental utilisé pour évaluer les performances obtenues avec le modèle optimisé de la signature VLAT en classification automatique. Nous utilisons la base d'images VOC2007 (Annexe A.2.1).

III.3.3.a Protocole expérimental

Chaque image est convertie en niveau de gris puis nous effectuons comme pré-traitement une égalisation d'histogramme. Nous utilisons le descripteur local HoG, nous effectuons une extraction dense des descripteurs locaux. Nous calculons un HoG tous les 3 ou 6 pixels sur 4 échelles (4, 6, 8 et 10 pixels par cellule du HoG). Par la suite nous effectuons une normalisation ℓ_2 des descripteurs locaux.

Pour calculer les paramètres du modèle, nous utilisons 10000 images extraites de la base Flickr60k. Le partitionnement des données est effectué à l'aide d'un algorithme de type k-means sur un échantillon de plusieurs millions de descripteurs HoG. La moyenne et la variance dans chaque partie sont estimées à partir des descripteurs utilisés pour obtenir le partitionnement.

Nous utilisons un classificateur linéaire par hyperplan. Nous entraînons le classificateur avec le protocole un contre tous à l'aide d'un SVM à descente de gradient stochastique avec l'hyperparamètre du SVM $C = 10$.

III.3.3.b Résultats de références

Pour évaluer les gains obtenus par l'utilisation du modèle optimisé, nous utilisons plusieurs résultats obtenus avec des signatures de l'état de l'art. Nous nous intéressons uniquement aux résultats obtenus avec un classificateur linéaire.

Nous reportons dans le Tableau III.2 plusieurs résultats obtenus par des signatures de l'état de l'art, ainsi que des résultats obtenus par nos réimplantations des signatures de l'état de l'art. La première colonne du tableau est le nom de la signature (FV : *Fisher kernel*, SV : *Super Vector coding*, LLC : *Locally-Constrained Linear coding*, VQ : *Vector Quantization*, VLAD : *Vector of Locally Aggregated Descriptors*, VLAT : *Vectors of Locally*

	Signature	Mots visuels	Dim.	mAP (en %)
FV	[Chatfield et al., 2011]	256	65.5k	61.7
SV	[Chatfield et al., 2011]	1024	33k	58.2
LLC	[Chatfield et al., 2011]	25k	25k	57.3
VQ	[Chatfield et al., 2011]	4k	4k	46.5
VLAD	[Picard and Gosselin, 2011]	16	2k	40.4
VLAD	[Picard and Gosselin, 2011]	32	4k	42.2
VLAD	[Picard and Gosselin, 2011]	64	8k	43.6
VLAT	[Picard and Gosselin, 2011]	16	130k	44.5
VLAT	[Picard and Gosselin, 2011]	32	260k	46.1
VLAT	[Picard and Gosselin, 2011]	64	520k	46.4
VLAT		64	520k	47.8
VLAT		128	1M	49.3

TABLE III.2 – Performances en classification des signatures de l'état de l'art sur la base VOC2007

Aggregated Tensors). La deuxième colonne est le nombre de partitions utilisé pour calculer la signature. La troisième colonne est la dimension de la signature finale. La quatrième colonne est la performance de la signature en mAP (*mean Average Precision*). Le tableau est divisé en trois parties, la partie du haut correspond à des signatures calculées avec le descripteur SIFT [Lowe, 1999] extrait tous les 3 pixels sur 4 échelles. La partie du milieu correspond à des signatures calculées avec le descripteur HOG extrait tous les 8 pixels sur 3 échelles. La partie du bas correspond aux résultats obtenus par notre implantation de la signature VLAT calculées avec le descripteur HOG extrait tous les 6 pixels sur 4 échelles.

Nous voyons que plus l'extraction des descripteurs est dense, plus les signatures ont de bonnes performances. Nous notons également que plus le partitionnement de l'espace est fin, plus les signatures ont de bonnes performances.

III.3.3.c Signature VLAT optimisée

Pour évaluer les gains obtenus avec la signature VLAT optimisée, nous utilisons un modèle avec un partitionnement de l'espace des descripteurs en 64 parties. Nous nous intéressons à deux paramètres : la dimension de la signature obtenue et ses performances en classification sur la base VOC2007. Nous évaluons les deux méthodes de sélection des dimensions dans les parties du modèle : la méthode locale et la méthode globale (Section III.2.2). Pour la méthode locale, nous calculons les signatures en conservant : 16, 32, 48, 64, 80, 112 et 128 dimensions par partie. Pour la méthode globale, nous calculons les signatures en conservant : 70%, 75%, 80%, 85%, 90%, 95% et 100% de l'énergie dans chaque partie.

La Figure III.3 montre l'évolution des performances de classification sur la base VOC2007 en fonction W de la dimension de la signature obtenue. L'étoile rouge est le mAP obtenu avec la signature VLAT non optimisée, 47.8% de mAP pour $W = 528k$ dimensions. La courbe continue représente les performances obtenues avec la méthode de sélection locale. La courbe en pointillée représente les performances obtenues avec la méthode de sélection globale. Nous voyons qu'à dimension équivalente la signature VLAT optimisée obtient de bien meilleures performances que la signature VLAT non-optimisée,

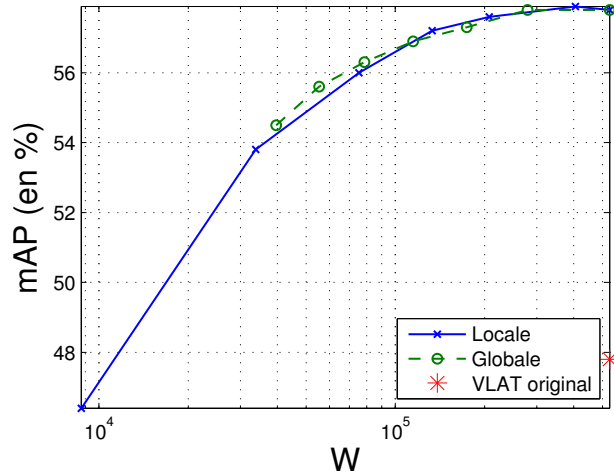


FIGURE III.3 – Performances (mAP) de la signature VLAT optimisée sur VOC2007 en fonction sa dimension W .

plus de 10% de gain. Nous constatons que les deux méthodes de sélection donnent des performances très similaires. Il est très intéressant de noter que l'on peut fortement réduire la dimension de la signature obtenue tout en conservant de très bonnes performances. Par exemple avec la méthode de sélection globale, nous pouvons réduire la signature de $W = 528k$ dimensions à $W = 133k$ dimensions avec seulement une perte de 0.6% de mAP.

Pour nous comparer à l'état de l'art, nous calculons une signature VLAT optimisée avec un partitionnement de 256 parties et une extraction des descripteurs HoG tous les 3 pixels. Nous utilisons la méthode de sélection globale avec laquelle nous conservons 90% de l'énergie dans chaque partie du modèle sans blanchiment. La Tableau III.3 montre pour différents signatures de l'état de l'art et notre signature VLAT optimisée les performances de classification obtenues pour chaque classe de la base VOC2007. Nous observons que notre signature VLAT optimisée obtient des performances très similaires aux performances de la signature Fisher Vector avec le même nombre de parties.

III.4 Conclusions

Nous avons présenté dans ce chapitre une nouvelle signature d'image basée sur la mesure de déviation à un modèle. Cette signature est une extension de la signature VLAT obtenue par l'approximation du noyau sur sac Gaussien. Nous avons explicité le modèle utilisé par la signature VLAT et nous avons proposé une amélioration de ce modèle. Cette amélioration permet de rendre la signature plus discriminante tout en réduisant la dimension de la signature.

Nous avons effectué une étude des coûts calculatoires et des coûts de stockage nécessaires au calcul de la signature. Nous avons observé que l'utilisation du modèle optimisé permet de réduire ces coûts.

Les résultats expérimentaux ont montré qu'à dimension équivalente, les performances de notre signature sont bien supérieures à la signature VLAT d'origine. Nous avons constaté que la dimension de la signature peut être fortement réduite avec un minimum de perte de performance. Nous avons observé qu'il est possible d'obtenir des performances

	VQ	SV	LLC	FV	Nos résultats
Dim.	4k	33k	25k	65.5k	718k
aeroplane	60.6	74.3	71.4	78.9	80.3
bicycle	48.8	36.7	62.7	67.4	72.2
bird	32.8	47.0	46.1	51.9	51.4
boat	58.5	69.4	69.0	70.9	71.4
bottle	16.3	29.1	26.0	30.8	28.1
bus	50.4	66.5	63.9	72.2	72.1
car	68.4	77.3	77.0	79.9	81.6
cat	46.0	60.2	59.7	61.4	63.1
chair	48.3	50.2	54.0	56.0	54.4
cow	28.3	46.5	46.3	49.6	47.5
table	41.7	51.9	52.1	58.4	57.8
dog	30.2	44.1	42.4	44.8	46.5
horse	70.5	77.9	77.2	78.8	81.1
bike	55.2	67.1	67.2	70.8	70.3
person	77.8	83.1	83.4	85.0	86.8
plant	14.8	27.6	23.1	31.7	30.8
sheep	31.4	48.5	44.5	51.0	41.2
sofa	39.7	51.1	52.1	56.4	54.0
train	38.3	75.5	75.4	80.2	84.1
tv	42.7	52.3	52.2	57.5	54.9
mAP	64.5	58.2	57.3	61.7	61.5

TABLE III.3 – Comparaisons aux résultats de l'état de l'art en classification d'image sur VOC 2007 extraient de [Chatfield et al., 2011].

supérieures aux signatures de l'état de l'art tout en réduisant la signature à des dimensions du même ordre de grandeur. Cependant, la réduction de dimension nécessaire pour permettre le passage à l'échelle des méthodes d'indexation par le contenu entraîne une forte perte de performance de la signature. Il est alors préférable de rechercher un compromis entre la dimension de la signature obtenue, les coûts d'extraction et les performances de la signature. Les signatures de grandes dimensions alors obtenues doivent être réduites par un post-traitement.

Deuxième partie

Réduction de dimension

Réduction de dimension non-supervisée

Chapitre IV

Comme nous l'avons présenté dans la partie précédente (III.3), l'utilisation de signatures pour le calcul de similarités entre les images permet d'obtenir de bonnes performances en recherche par similarité et en classification. Cette approche se décompose en 2 étapes : l'extraction d'un ensemble de descripteurs locaux dans chaque document ; puis le calcul d'une signature pour chaque document par agrégation des descripteurs locaux. La similarité entre les documents est alors obtenue en évaluant une fonction de similarité entre les signatures.

Les expérimentations montrent que les signatures de grande dimension permettent d'obtenir de meilleures performances. Cependant, l'utilisation des signatures de grande dimension entraîne de forts coûts de stockage et de calcul. Lors du passage à l'échelle ces coûts deviennent prohibitifs, et rendent très complexe la mise en œuvre des méthodes d'indexation automatique. Pour permettre le passage à l'échelle des méthodes d'indexation automatique basées sur les signatures d'images, il est indispensable de réduire ces coûts.

De nombreuses méthodes ont été proposées pour réduire ces coûts. Par exemples des méthodes basées sur la réduction de la dimensionnalité des signatures, ou encore des méthodes basées sur le codage des signatures.

Dans cette partie, nous présentons puis proposons un ensemble de méthodes non supervisées pour fortement réduire la dimensions de ces signatures.

IV.1 État de l'art

Dans cette section, nous présentons un ensemble de méthodes non supervisées utilisées en indexation pour réduire les coûts d'utilisation des signatures de grande dimension. Nous nous intéressons particulièrement aux méthodes de réduction de dimension.

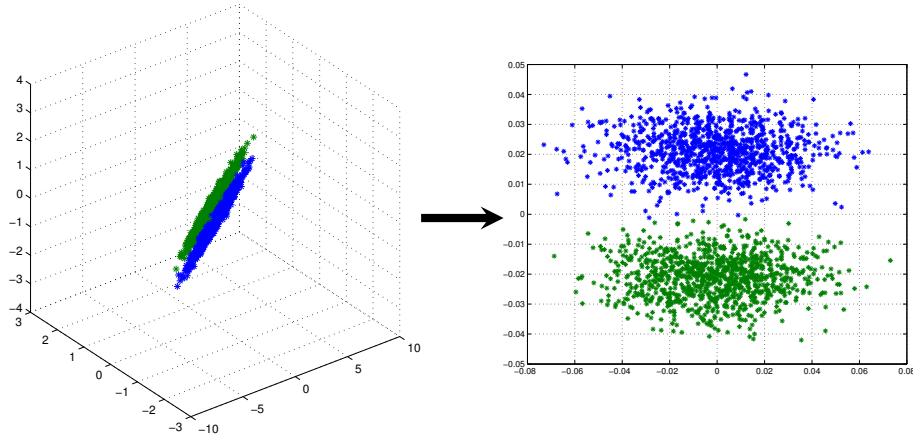


FIGURE IV.1 – Exemple de projection linéaire de vecteurs de dimension 3 dans un espace de dimension 2.

IV.1.1 Réduction de dimension

De nombreuses applications utilisant une représentation vectorielle des données nécessitent la mise en œuvre de méthodes de réduction de leur dimension [Van der Maaten et al., 2009]. Pour réduire la dimension d'une signature, nous considérons l'ensemble des fonctions qui, à chaque vecteur d'un espace vectoriel de dimension W , lui associent un vecteur d'un espace vectoriel plus petit de dimension $N < W$:

$$\begin{aligned} f: \mathbb{R}^W &\rightarrow \mathbb{R}^N \\ \mathbf{x} &\mapsto f(\mathbf{x}) \end{aligned} \tag{IV.1}$$

où $f(\cdot)$ est la fonction de projection et $f(\mathbf{x})$ le projeté du vecteur $\mathbf{x} \in \mathbb{R}^W$ dans $\mathbb{R}^N$.

Les fonctions de réduction de dimension sont le plus souvent construites ou choisies de façon à minimiser un critère d'erreur de reconstruction ou à maximiser un critère de conservation de l'information. Dans notre cas, nous voulons réduire la dimension des signatures tout en conservant leurs performances en classification et recherche par similarité.

IV.1.1.a Projection linéaire

L'une des approches les plus simples pour la réduction de la dimension est la projection linéaire (Figure IV.1) :

$$\mathbf{y} = f(\mathbf{x}) = \mathbf{P}^\top \mathbf{x}, \tag{IV.2}$$

avec $\mathbf{y} \in \mathbb{R}^N$ le vecteur réduit, $\mathbf{x} \in \mathbb{R}^W$ le vecteur d'origine et $\mathbf{P} \in \mathcal{M}_{W,N}$ la matrice des projecteurs. Elle est utilisée dans de nombreuses méthodes comme *Principal component analysis* (PCA) [Pearson, 1901, Hotelling, 1933, Jolliffe, 2005], *Canonical correlation analysis* (CCA) [Hotelling, 1936], *Linear discriminant analysis* (LDA) [Dudoit et al., 2002], ou encore *Independent component analysis* (ICA) [Hyvärinen et al., 2004].

L'analyse en composantes principales (ACP) [Jolliffe, 2005] est une méthode très utilisée en analyse de données. Cette méthode est basée sur une projection linéaire des données (IV.6). Chaque composante y_i du vecteur $\mathbf{y}$ est appelée une composante principale. Chaque composante principale y_i est obtenue par la projection de la donnée $\mathbf{x}$ sur le projecteur $\mathbf{p}_i$ appelé axe principal. Les axes principaux sont calculés de telle sorte que les

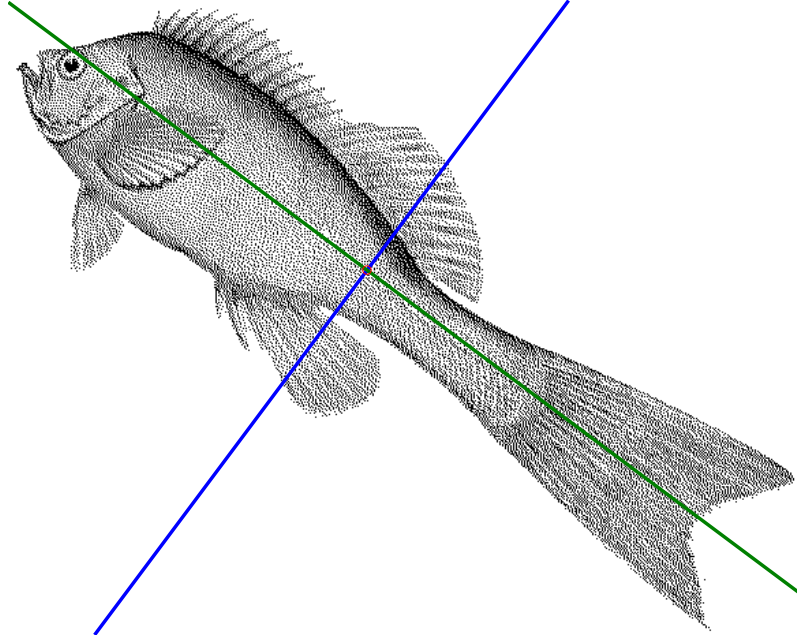


FIGURE IV.2 – Exemple d'axes principaux obtenus par l'ACP

composantes principales soit décorrélées ($\text{Corr}(y_i, y_j) = 0, \forall j \neq i$) et qu'ils permettent de conserver le maximum de variance présente dans l'ensemble des composantes initiales. Les composantes principales sont ordonnées de tel sorte que les premières composantes principales conservent la majeure partie de la variance (Figure IV.2). Ces propriétés sont très intéressantes pour la réduction de dimension car elle garantit de minimiser l'erreur de reconstruction des données [Jolliffe, 2005].

Nous pouvons obtenir la matrice de projection $\mathbf{P}$ (dans laquelle chaque colonne $\mathbf{p}_i$ est un axe principal) par une méthode itérative. Le premier axe principal est calculé en choisissant l'axe qui maximise la variance des données projetées sur cet axe. Les axes principaux suivants sont obtenus itérativement : en premier, nous retirons de l'espace des données les dimensions correspondant aux axes principaux précédemment calculés ; l'axe principal suivant est alors obtenu en choisissant l'axe qui maximise la variance des données dans l'espace restant. Les projecteurs ainsi obtenus sont orthonormés : $\mathbf{P}^\top \mathbf{P} = \mathbf{I}$.

Une méthode pour obtenir cette matrice de projection $\mathbf{P}$ s'appuie sur la décomposition en vecteurs et valeurs propres de la matrice de covariance des données [Horn and Johnson, 2012]. Pour estimer la matrice de projection à partir d'un ensemble d'échantillons de données $\{\mathbf{x}_i\}_{i=1}^L \in \mathbb{R}^W$, nous calculons la matrice de covariance des données :

$$\mathbf{\Sigma} = \frac{1}{L} \sum_{i=1}^L (\mathbf{x}_i - \bar{\mathbf{x}})^\top (\mathbf{x}_i - \bar{\mathbf{x}}) \quad (\text{IV.3})$$

avec $\bar{\mathbf{x}} = \frac{1}{L} \sum_{i=1}^L \mathbf{x}_i$ la moyenne empirique des données. Puis nous calculons sa décomposition en valeurs et vecteurs propres :

$$\mathbf{\Sigma} = \mathbf{P} \mathbf{D} \mathbf{P}^\top, \quad (\text{IV.4})$$

où la matrice des vecteurs propres $\mathbf{P}$ correspond à la matrice de projection de l'ACP et les valeurs propres ($\mathbf{D}$) correspondent aux variances des données dans ces directions.

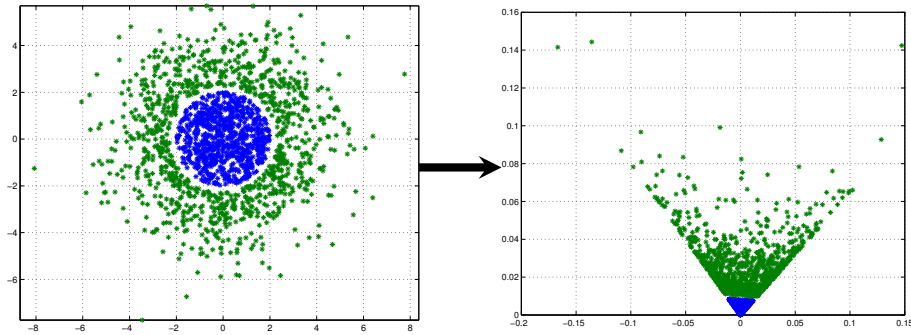


FIGURE IV.3 – Exemple de projection non-linéaire

Dans le cas de l'ACP, les données doivent être centrées avant d'être projetées :

$$\mathbf{y} = f(\mathbf{x}) = \mathbf{P}^\top (\mathbf{x} - \bar{\mathbf{x}}). \quad (\text{IV.5})$$

Il est également possible de blanchir des données :

$$\mathbf{y} = f(\mathbf{x}) = \mathbf{D}^{-1/2} \mathbf{P}^\top (\mathbf{x} - \bar{\mathbf{x}}), \quad (\text{IV.6})$$

Cela permet d'avoir une variance unitaire pour toutes les dimensions du vecteur $\mathbf{y}$.

Jégou *et al.* ont proposé dans [Jégou et al., 2010] d'utiliser une ACP pour réduire la dimension des signatures VLAD. Ils ont montré qu'en conservant un petit nombre de dimensions par rapport à la dimension de l'espace de départ, ils conservent une très bonne performance pour la recherche de quasi-copies d'images.

IV.1.1.b Projection non-linéaire

De nombreuses distributions de données ne peuvent pas être modélisées par un modèle linéaire. Prenons l'exemple de données distribuées en deux spirales imbriquées, une analyse en composantes principales ne permet pas de déterminer une direction d'analyse privilégiée, ni de rendre séparable les données linéairement. Pour ce genre de distribution, il existe des méthodes de projection non-linéaire qui permettent de déplier les données afin d'en avoir une représentation simple et de les rendre séparable linéairement (Figure IV.3). Il existe de nombreuses méthodes de transformation non-linéaire, par exemple : *Isomap* [Teng et al., 2005], *Kernel-PCA* [Schölkopf et al., 1997], *Locally Linear Embedding* (LLE) [Saul et al., 2006].

L'analyse en composantes principales peut être étendue aux représentations non linéaires grâce au formalisme des fonctions noyaux [Schölkopf et al., 1997]. L'idée principale de l'ACP non-linéaire est simple, nous effectuons une injection non-linéaire des données dans un espace de grande dimension pour les rendre modélisables linéairement. Puis, nous effectuons une ACP dans l'espace de grande dimension. Il est bien-sûr possible d'utiliser cette approche (injection non-linéaire + modélisation linéaire) avec d'autres modèles linéaires que l'ACP. Cependant, l'ACP est particulièrement intéressante car en utilisant le formalisme des fonctions noyaux, il est possible d'effectuer l'ACP non-linéaire sans effectuer d'opération dans l'espace de grande dimension [Shawe-Taylor and Cristianini, 2004].

L'utilisation du formalisme des fonctions noyaux est possible car il existe un lien direct entre les valeurs/vecteurs propres de la matrice de covariance et les valeurs/vecteurs

propres de la matrice de Gram (matrice des produits scalaires). Nous définissons $\mathbf{X} \in \mathcal{M}_{W \times L}$ un ensemble de données et $\mathbf{X}_\Phi \in \mathcal{M}_{W' \times L}$ l'ensemble $\mathbf{X}$ injecté dans un espace de grande dimension $W' \gg W$. Cette relation est alors donnée par la décomposition en valeurs singulières des données $\mathbf{X}_\Phi$:

$$\mathbf{X}_\Phi = \mathbf{P}\mathbf{D}^{1/2}\mathbf{U}^\top, \quad (\text{IV.7})$$

avec $\mathbf{P}$ les vecteurs propres de la matrice de covariance $\Sigma_\Phi = \mathbf{X}_\Phi \mathbf{X}_\Phi^\top = \mathbf{P}\mathbf{D}\mathbf{P}^\top$ et $\mathbf{U}$ les vecteurs propres de la matrice de Gram $\mathbf{G}_\Phi = \mathbf{X}_\Phi^\top \mathbf{X}_\Phi = \mathbf{U}\mathbf{D}\mathbf{U}^\top$. Nous pouvons alors obtenir les projecteurs $\mathbf{P}$ de l'ACP par la composition des données $\mathbf{X}_\Phi$ par les vecteurs propres $\mathbf{U}$ de la matrice de Gram :

$$\mathbf{P} = \mathbf{X}_\Phi \mathbf{U} \mathbf{D}^{-1/2}. \quad (\text{IV.8})$$

Et la projection d'un vecteur $\mathbf{z} \in \mathbb{R}^W$ s'écrit alors :

$$\mathbf{y} = \mathbf{P}^\top \mathbf{z}_\Phi = \mathbf{D}^{-1/2} \mathbf{U}^\top \mathbf{X}_\Phi^\top \mathbf{z}_\Phi, \quad (\text{IV.9})$$

avec $\mathbf{z}_\Phi \in \mathbb{R}^{W'}$ le vecteur $\mathbf{z}$ injecté dans un espace de grande dimension.

Nous pouvons voir que tous les termes de l'équation (IV.9) peuvent être obtenus en utilisant la fonction noyau : $(\mathbf{U}, \mathbf{D})$ sont obtenus par décomposition de la matrice de Gram dans l'espace induit qui peut être directement obtenu avec la fonction noyau ; et $\mathbf{X}_\Phi^\top \mathbf{z}_\Phi$ le vecteur des produits scalaires entre les données $\mathbf{X}$ et le vecteur $\mathbf{z}$ dans l'espace induit qui peut être également obtenu avec la fonction noyau.

IV.1.2 Codage de signature

Les méthodes de codage de signature produisent une représentation efficace des signatures en mémoire. Cela permet de réduire le coût de stockage des signatures, mais également de réduire le coût calculatoire des calculs de similarité entre deux signatures. Ces méthodes proviennent de la théorie du codage et plus particulièrement de la théorie du codage de source [Cover and Thomas, 2012]. Le codage de source rassemble les méthodes utilisées pour réduire la dimension des messages à transmettre à travers un canal de communication. Ces méthodes sont généralement appelées algorithmes de compression.

De nombreuses méthodes de compression utilisent une approche par dictionnaire de code. La première étape consiste à construire le dictionnaire soit en connaissant à priori la distribution statistique des données, soit en utilisant un ensemble échantillon des données. Pour coder une donnée, la fonction de codage utilise le dictionnaire pour fournir une représentation très compacte de la donnée.

Jégou *et al.* ont proposé d'utiliser une quantification vectorielle pour réduire les coûts entraînés par les signatures de grande dimension [Jégou et al., 2011]. Les auteurs proposent de découper la signature en bloc de taille fixe et d'effectuer dans chaque bloc une quantification vectorielle.

La quantification vectorielle consiste à construire un dictionnaire des codes (vecteurs prototypes) : $\mathcal{C} = \{\mathbf{c}_i\}_{i \in \mathcal{I}}$ avec $\mathcal{I} = 0 \dots K-1$ l'ensemble des indices. Puis, pour quantifier un vecteur, nous recherchons dans le dictionnaire le vecteur prototype le plus proche :

$$\mathbf{q}_\mathcal{C}(\mathbf{x}) = \arg \min_{\mathbf{c}_i \in \mathcal{C}} \|\mathbf{x} - \mathbf{c}_i\|_2 \quad (\text{IV.10})$$

Les vecteurs prototypes sont obtenus en minimisant l'erreur d'approximation du vecteur par le vecteur prototype :

$$\mathcal{C} = \arg \min_{\mathcal{C}} \sum_{\mathbf{x} \in \mathcal{T}} \|\mathbf{x} - \mathbf{q}_{\mathcal{C}}(\mathbf{x})\|_2 \quad (\text{IV.11})$$

avec $\mathcal{T}$ un ensemble de vecteurs d'entraînement. Le dictionnaire est généralement construit à l'aide d'un algorithme de type k-means [Jain, 2010]. Une fois le vecteur quantifié, seul l'indice i du vecteur prototype $\mathbf{c}_i$ est stocké.

Le coût de stockage d'une signature quantifiée par bloc est alors égal au coût de stockage des indices des blocs. Par exemple, pour une signature découpée en P blocs et avec des dictionnaires de K vecteurs prototypes, le coût de stockage est alors de $P \times \lfloor \log_2(K) \rfloor$ bits.

De plus, si nous utilisons une fonction de similarité linéaire, il n'est pas nécessaire de reconstruire les deux signatures pour calculer leur similarité. En effet, si la fonction de similarité est linéaire alors la similarité de la signature reconstruite est égale à la somme des similarités de Chacun des blocs. L'ensemble des similarités entre les K vecteurs prototypes de chaque bloc peut être pré-calculé. Le coût de calcul d'une similarité est fortement réduit car elle ne dépend plus de la dimension de la signature W , mais du nombre de blocs P utilisé pour coder la signature.

IV.2 Méthodes proposées

Dans cette section, nous proposons un ensemble de méthodes non supervisées pour diminuer les coûts engendrés par les signatures de grande dimension. Nous proposons de réduire la dimension des signatures par une projection linéaire.

Nous notons $\mathbf{x}_i \in \mathbb{R}^W$ la signature de la $i^{\text{ème}}$ image. Nous traitons la signature $\mathbf{x}_i$, pour obtenir $\mathbf{y}_i \in \mathbb{R}^N$ la signature de dimension réduite $N \ll W$. Pour réduire la dimension des signatures, nous utilisons une projection linéaire : $\mathbf{y}_i = \mathbf{P}^\top \mathbf{x}_i$ avec $\mathbf{P}_N \in \mathcal{M}_{N \times W}$ la matrice des projecteurs. Pour entraîner la matrice $\mathbf{P}_N$, nous utilisons L signatures calculées sur des images tirées aléatoirement. Nous notons $\mathbf{X} \in \mathcal{M}_{W \times L}$ la matrice des signatures d'entraînement.

La première méthode que nous proposons permet de réduire fortement la dimension des signatures tout en conservant leur pouvoir de discrimination. Puis, nous proposons une méthode pour réduire fortement les coûts engendrés par la procédure de réduction de grandes signatures par projection linéaire. En dernier, nous proposons une méthode pour effectuer la correction des erreurs introduites par l'étape de réduction des signatures.

IV.2.1 Projecteurs denses

Nous proposons en premier lieu une méthode pour obtenir des projecteurs linéaires denses.

IV.2.1.a Objectif et contraintes

Pour obtenir des signatures réduites qui conservent de bonnes performance en classification et en recherche par similarité, il est important de bien choisir le critère à optimiser pour obtenir les projecteurs. Dans le cas de l'apprentissage non supervisé, nous utilisons uniquement un ensemble de signatures pour entraîner nos projecteurs.

En recherche par similarité et en classification automatique, l'information la plus importante à conserver est le score de similarité entre deux signatures. Nous proposons de calculer les projecteurs qui minimisent l'erreur de reconstruction des valeurs de similarité entre des signatures de l'ensemble d'entraînement.

IV.2.1.b Problème de reconstruction

Avec les signatures de grande dimension, la fonction de similarité couramment utilisé est le produit scalaire normalisé. Nous proposons d'utiliser la même fonction de similarité dans l'espace réduit. Pour éviter les problèmes liés à l'utilisation du produit scalaire normalisé dans un problème d'optimisation, nous proposons de pré-normaliser les signatures d'entraînement, de post-normaliser les signatures réduites et de considérer le produit scalaire usuel dans le problème d'optimisation.

La fonction objectif que nous cherchons alors à minimiser est l'erreur quadratique moyenne de reconstruction des valeurs de similarités entre toutes les paires d'images et s'écrit :

$$h(\mathbf{P}_N) = \sum_{i=1}^L \sum_{j=1}^L (\mathbf{x}_i^\top \mathbf{x}_j - (\mathbf{P}_N^\top \mathbf{x}_i)^\top \mathbf{P}_N^\top \mathbf{x}_j)^2, \quad (\text{IV.12})$$

$$= \|\mathbf{X}^\top \mathbf{X} - \mathbf{X}^\top \mathbf{P}_N \mathbf{P}_N^\top \mathbf{X}\|_F^2, \quad (\text{IV.13})$$

avec $\mathbf{X}$ la matrice des signatures et $\|\cdot\|_F$ la norme de Frobenius. Notons que $\mathbf{G} = \mathbf{X}^\top \mathbf{X}$ est la matrice de Gram des signatures d'origine et $\mathbf{G}_{\mathbf{P}_N} = \mathbf{X}^\top \mathbf{P}_N \mathbf{P}_N^\top \mathbf{X}$ la matrice de Gram des signatures réduites.

La grande dimension des projecteurs par rapport au nombre d'exemples d'apprentissage peut entraîner des phénomènes de sur-apprentissage. En effet sans contrainte supplémentaire sur les projecteurs, nous pouvons apprendre des projecteurs qui reconstruisent parfaitement les valeurs de similarités. Cependant, ces projecteurs n'offrent aucune capacité de généralisation. Pour éviter les phénomènes de sur-apprentissage, nous proposons d'ajouter une contrainte d'orthonormalité des projecteurs. Le problème de minimisation s'écrit :

$$\begin{aligned} \mathbf{P}_N^* &= \arg \min_{\mathbf{P}_N} \|\mathbf{X}^\top \mathbf{X} - \mathbf{X}^\top \mathbf{P}_N \mathbf{P}_N^\top \mathbf{X}\|_F^2 \\ \text{s.t. } \mathbf{P}_N &\in \mathcal{M}_{W,N} \text{ avec } N < L \ll W \\ \mathbf{P}_N^\top \mathbf{P}_N &= \mathbf{I} \end{aligned} \quad (\text{IV.14})$$

Ce problème d'optimisation a une infinité de solutions équivalentes. En effet, pour toute solution $\mathbf{P}_N^*$ qui minimise notre fonction objectif $h(\mathbf{P}_N)$, nous pouvons calculer une infinité de solutions équivalentes à une rotation près :

$$\mathbf{P}_N'^* = \mathbf{P}_N^* \mathbf{A}, \quad (\text{IV.15})$$

avec $\mathbf{A}$ une matrice de rotation ($\mathbf{A}^\top \mathbf{A} = \mathbf{A} \mathbf{A}^\top = \mathbf{I}$) car l'objectif et les contraintes sont invariants par rotation.

IV.2.1.c Solution par approximation de rang faible

La solution du problème d'optimisation (IV.14) est obtenue en calculant une approximation de rang faible de la matrice de Gram $\mathbf{G}$ de l'ensemble des signatures d'entraînement :

$$\mathbf{G} = \mathbf{X}^\top \mathbf{X} \quad (\text{IV.16})$$

Nous calculons une décomposition en valeurs propres et vecteurs propres cette matrice :

$$\mathbf{G} = \mathbf{U}\mathbf{L}\mathbf{U}^\top \quad (\text{IV.17})$$

$$\mathbf{L} = \text{diag}(\lambda_1, \dots, \lambda_L) \quad (\text{IV.18})$$

$$\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_L) \quad (\text{IV.19})$$

$$\mathbf{U}^\top \mathbf{U} = \mathbf{U}\mathbf{U}^\top = \mathbf{I} \quad (\text{IV.20})$$

avec $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_L \geq 0$ les valeurs propres et $\mathbf{u}_i$ le vecteur propre associé à la valeur propre λ_i , $\forall i$.

Pour obtenir une approximation de rang faible de la matrice de Gram $\mathbf{G}$, nous sélectionnons les vecteurs propres correspondants aux N plus grandes valeurs propres avec $N < L$. L'utilisation de ces vecteurs propres permet d'avoir l'approximation de rang faible la plus proche de la matrice de Gram d'origine. Nous notons par $\mathbf{L}_N$ la matrice diagonale avec les N plus grandes valeurs propres :

$$\mathbf{L}_N = \text{diag}(\lambda_1, \dots, \lambda_N) \quad (\text{IV.21})$$

et nous notons par $\mathbf{U}_N$ la matrice des N vecteurs propres correspondants :

$$\mathbf{U}_N = (\mathbf{u}_1, \dots, \mathbf{u}_N). \quad (\text{IV.22})$$

L'approximation de rang N de la matrice de Gram s'écrit alors :

$$\mathbf{G}_N = \mathbf{U}_N \mathbf{L}_N \mathbf{U}_N^\top \quad (\text{IV.23})$$

Nous obtenons la matrice des projecteurs en projetant les vecteurs propres sélectionnés sur l'ensemble des signatures d'apprentissage :

$$\mathbf{P}_N = \mathbf{X}\mathbf{U}_N \mathbf{L}_N^{-1/2}. \quad (\text{IV.24})$$

Si nous remplaçons ces projecteurs dans la fonction objectif (IV.12), nous obtenons :

$$h(\mathbf{P}_N) = \|\mathbf{X}^\top \mathbf{X} - \mathbf{X}^\top \mathbf{X} \mathbf{U}_N \mathbf{L}_N^{-1/2} \mathbf{L}_N^{-1/2} \mathbf{U}_N^\top \mathbf{X}^\top \mathbf{X}\|_F^2, \quad (\text{IV.25})$$

$$= \|\mathbf{U}\mathbf{L}\mathbf{U}^\top - \mathbf{U}\mathbf{L}\mathbf{U}^\top \mathbf{U}_N \mathbf{L}_N^{-1} \mathbf{U}_N^\top \mathbf{U}\mathbf{L}\mathbf{U}^\top\|_F^2, \quad (\text{IV.26})$$

$$= \|\mathbf{U}\mathbf{L}\mathbf{U}^\top - \mathbf{U}_N \mathbf{L}_N \mathbf{L}_N^{-1} \mathbf{L}_N \mathbf{U}_N^\top\|_F^2, \quad (\text{IV.27})$$

$$= \|\mathbf{U}\mathbf{L}\mathbf{U}^\top - \mathbf{U}_N \mathbf{L}_N \mathbf{U}_N^\top\|_F^2, \quad (\text{IV.28})$$

$$= \|\mathbf{U}_N \mathbf{L}_N \mathbf{U}_N^\top + \bar{\mathbf{U}}_N \bar{\mathbf{L}}_N \bar{\mathbf{U}}_N^\top - \mathbf{U}_N \mathbf{L}_N \mathbf{U}_N^\top\|_F^2, \quad (\text{IV.29})$$

$$= \|\bar{\mathbf{U}}_N \bar{\mathbf{L}}_N \bar{\mathbf{U}}_N^\top\|_F^2, \quad (\text{IV.30})$$

$$= \|\bar{\mathbf{L}}_N\|_F^2, \quad (\text{IV.31})$$

$$= \sum_{k=N+1}^L \lambda_k^2 \quad (\text{IV.32})$$

avec $\mathbf{U}\mathbf{L}\mathbf{U}^\top = \mathbf{U}_N \mathbf{L}_N \mathbf{U}_N^\top + \bar{\mathbf{U}}_N \bar{\mathbf{L}}_N \bar{\mathbf{U}}_N^\top$. Nous supprimons tous les axes correspondants aux plus fortes valeurs propres de la matrices de Gram. L'erreur de reconstruction alors obtenue est égale à la somme des plus petites valeurs propres au carré de la matrice de Gram $\mathbf{G}$.

Cette matrice de projection permet de produire des signatures de très petite dimension tout en conservant au mieux la similarité obtenue avec les projecteurs d'origines. Remarquons que les projecteurs obtenus sont les N premiers vecteurs propre de la matrice de covariance des signatures. Notre méthode est analogue à une ACP sans effectuer le centrage des signatures avant la projection. En effet, si l'ACP permet de conserver au mieux la matrice des distances ℓ_2 dans l'espace réduit (*Multi Dimensional scaling* [Torgerson, 1958]), elle ne permet pas de conserver la matrice des produits scalaires car le produit scalaire est sensible au centrage des données.

IV.2.1.d Similarité dans l'espace réduit

Dans le problème d'optimisation (IV.14), nous considérons le produit scalaire usuel comme fonction de similarité. Cela nous permet d'obtenir une solution analytique du problème. Cependant, la faible dimension de l'espace réduit nous permet d'utiliser des fonctions de similarité plus robustes.

Nous proposons d'utiliser le produit scalaire associé à la distance de Mahalanobis :

$$s(\mathbf{y}_i, \mathbf{y}_j) = \mathbf{y}_j^\top \boldsymbol{\Sigma}^{-1} \mathbf{y}_i. \quad (\text{IV.33})$$

avec $\mathbf{y}_i$ et $\mathbf{y}_j$ les signatures réduites correspondant aux $i^{\text{ème}}$ et $j^{\text{ème}}$ images et $\boldsymbol{\Sigma}$ la matrice de covariance des signatures réduites obtenues sur l'ensemble d'entraînement. Cette fonction de similarité a les mêmes avantages que la distance de Mahalanobis [Mahalanobis, 1936]. En prenant en compte la covariance des données, elle permet de calculer la similarité dans l'espace des axes principaux des données. Mais également, de blanchir ces axes (tous les axes ont une variance unitaire). Dans le cas des signatures réduites avec nos projecteurs $\mathbf{P}_N$, les composantes sont décorréliées, leur matrice de covariance $\boldsymbol{\Sigma}$ est alors diagonale. Elle est égale à la matrice des valeurs propres conservée pour l'approximation de rang faible : $\boldsymbol{\Sigma} = \mathbf{L}_N$.

Nous proposons également d'utiliser la fonction de similarité normée associée qui s'écrit :

$$\begin{aligned} s'(\mathbf{y}_i, \mathbf{y}_j) &= \frac{s(\mathbf{y}_i, \mathbf{y}_j)}{s'(\mathbf{y}_i, \mathbf{y}_i) s(\mathbf{y}_j, \mathbf{y}_j)} \\ &= \frac{\mathbf{y}_j^\top \boldsymbol{\Sigma}^{-1} \mathbf{y}_i}{\mathbf{y}_i^\top \boldsymbol{\Sigma}^{-1} \mathbf{y}_i \mathbf{y}_j^\top \boldsymbol{\Sigma}^{-1} \mathbf{y}_j}. \end{aligned} \quad (\text{IV.34})$$

L'utilisation de la fonction de similarité normée permet d'obtenir les propriétés suivantes : si $s'(\mathbf{y}_i, \mathbf{y}_j) = 1$ alors $\mathbf{y}_i = \mathbf{y}_j$; et pour tout $\mathbf{y}_i \neq \mathbf{y}_j$ alors $s'(\mathbf{y}_i, \mathbf{y}_j) < 1$. Ces propriétés sont très intéressantes en indexation car elles garantissent que le plus fort score de similarité entre deux images est obtenu entre l'image et elle même.

En comparaison avec le produit scalaire usuel, cette fonction de similarité ajoute un coût calculatoire lors de la phase de recherche au sein de la base. Bien que l'augmentation de coût soit très faible lors du calcul d'une similarité, il peut rapidement devenir prohibitif quand une requête dans la base nécessite le calcul de millions de similarités. Nous proposons alors d'effectuer un post-traitement des signatures réduites afin d'obtenir les mêmes scores de similarités avec un simple produit scalaire qu'avec le produit scalaire normée associé à la distance de Mahalanobis. La matrice de covariance $\boldsymbol{\Sigma}$ est définie positive, nous pouvons alors la décomposer en deux :

$$\boldsymbol{\Sigma} = \mathbf{L}_N = \mathbf{L}_N^{1/2} \mathbf{L}_N^{1/2} \quad (\text{IV.35})$$

Nous pouvons alors réécrire la fonction de similarité (IV.34) de la manière suivante :

$$\begin{aligned} s'(\mathbf{y}_i, \mathbf{y}_j) &= \frac{\mathbf{y}_j^\top \mathbf{L}_N^{-1/2} \mathbf{L}_N^{-1/2} \mathbf{y}_i}{\|\mathbf{L}_N^{-1/2} \mathbf{y}_i\|_2 \|\mathbf{L}_N^{-1/2} \mathbf{y}_j\|_2} \\ &= \left(\frac{\mathbf{L}_N^{-1/2} \mathbf{y}_i}{\|\mathbf{L}_N^{-1/2} \mathbf{y}_i\|_2} \right)^\top \left(\frac{\mathbf{L}_N^{-1/2} \mathbf{y}_j}{\|\mathbf{L}_N^{-1/2} \mathbf{y}_j\|_2} \right) \end{aligned} \quad (\text{IV.36})$$

Le post-traitement des signatures réduites s'écrit alors :

$$\mathbf{z}_i = \frac{\mathbf{L}_N^{-1/2} \mathbf{y}_i}{\|\mathbf{L}_N^{-1/2} \mathbf{y}_i\|_2}. \quad (\text{IV.37})$$

La similarité entre les signatures est alors calculée avec le produit scalaire usuel.

IV.2.1.e Analyse des coût

Comme nous l'avons introduit, cette étape de réduction de dimension des signatures permet de réduire les coûts calculatoire et de stockage lors de la phase de recherche au sein de la base. Cependant, cette étape augmente les coûts lors de l'extraction des signatures.

L'étape de réduction de dimension des signatures rajoute à l'extraction : un coût de stockage de $\mathcal{O}(N \times W)$ (nombres à virgule flottante simple précision), correspondant au stockage de la matrice de projection ; et un coût calculatoire de $\mathcal{O}(N \times (W + 1))$ opérations unitaires, correspondant au calcul de la projection de la signature d'origine et à la normalisation de la signature réduite.

Comme les coûts d'utilisation sont directement proportionnels à la dimension de la signature, la forte réduction de la dimension des signature permet d'obtenir une forte réduction des coûts d'utilisation.

IV.2.2 Projecteurs creux

L'étape de réduction a permis de fortement réduire les coûts d'utilisation des signatures. Cependant, cela ajoute des coûts à l'extraction qui peuvent devenir prohibitifs quand la signature originale est de très grande dimension et que le nombre de signatures est grand. Dans cette section, nous proposons une méthode pour réduire ces coûts en produisant une matrice de projection creuse.

IV.2.2.a Projecteurs parcimonieux

Notre matrice de projection est basée sur la minimisation de l'erreur de reconstruction de la matrice de Gram d'un ensemble de signatures d'entraînement avec une contrainte de parcimonie.

Afin d'obtenir les projecteurs creux qui fournissent une meilleure approximation de la matrice de Gram, nous devons résoudre le problème (IV.14) contraint avec la norme ℓ_0 :

$$\begin{aligned} \mathbf{P}_N^* &= \arg \min_{\mathbf{P}_N} \|\mathbf{X}^\top \mathbf{X} - \mathbf{X}^\top \mathbf{P}_N \mathbf{P}_N^\top \mathbf{X}\|_F^2 \\ \text{s.t. } \mathbf{P}_N &\in \mathcal{M}_{W,N} \text{ avec } N < L \ll W \\ \mathbf{P}_N^\top \mathbf{P}_N &= \mathbf{I} \\ \|\mathbf{p}_i\|_0 &= M, \forall i \end{aligned} \quad (\text{IV.38})$$

avec M le nombre de valeurs non-nulles par projecteur de la matrice $\mathbf{P}_N$. Cependant, la résolution de ce problème est NP-Complet, il est très difficile (voir impossible) d'obtenir la solution globale en un temps raisonnable [Natarajan, 1995].

Nous proposons de décomposer ce problème en deux sous-problèmes pour obtenir une solution approximée. Le premier sous-problème consiste à calculer une approximation parcimonieuse de projecteurs denses. Le deuxième sous-problème consiste à calculer une matrice de post-traitement des signatures réduites avec les projecteurs creux, pour corriger les erreurs de similarités introduit par l'approximation des projecteurs.

Pour calculer la matrice de projection parcimonieuse, nous proposons de calculer l'approximation parcimonieuse des projecteurs denses $\mathbf{P}_N^*$. Pour cela, nous résolvons le problème suivant :

$$\begin{aligned} \tilde{\mathbf{P}}_N^* = \arg \min_{\tilde{\mathbf{P}}_N} \|\mathbf{P}_N^* - \tilde{\mathbf{P}}_N\|_F^2 \\ \text{s.t. } \|\tilde{\mathbf{p}}_i\|_0 = M, \forall i. \end{aligned} \quad (\text{IV.39})$$

Ce problème (IV.39) a une solution analytique obtenue par seuillage des M plus petites valeurs des projecteurs originaux :

$$\tilde{p}_{ki} = p_{ki} \, h(|p_{ik}| - \nabla_i), \forall k \quad (\text{IV.40})$$

avec $h(\cdot)$ la fonction de Heaviside, $\nabla_i \in \mathbb{R}^+$ le seuil sélectionné pour satisfaire la contrainte de parcimonie.

Pour chaque projecteur $\mathbf{p}_i$ de la matrice de projection, nous effectuons un tri des valeurs absolues de ces composantes. Puis, nous sélectionnons comme valeur de seuil ∇_i , la $M^{\text{ème}}$ valeurs des projecteurs triés.

IV.2.3 Correction d'erreurs de similarité

La méthode d'approximation parcimonieuse permet d'obtenir à faible coût une matrice de projection creuse. Cependant, cette méthode introduit des erreurs dans les similarités entre les signatures réduites. Dans cette section, nous proposons d'introduire un post-traitement pour réduire les erreurs de similarité entre les signatures réduites.

IV.2.3.a Matrice de correction

Avec la matrice de projection dense, les valeurs des signatures dans l'espace réduit sont dé-corrélées. L'approximation parcimonieuse de la matrice de projection a introduit des corrélations entre les dimensions des signatures réduites. Nous supposons qu'il est possible d'utiliser ces corrélations pour corriger les erreurs de similarité.

Nous proposons alors de construire une matrice de correction $\mathbf{R}_N$ dans l'espace réduit pour corriger les erreurs de similarité :

$$\mathbf{y}' = \mathbf{R}_N^\top \mathbf{y}. \quad (\text{IV.41})$$

IV.2.3.b Problèmes de reconstruction

Pour calculer la matrice de correction $\mathbf{R}_N$, nous proposons de minimiser l'erreur entre $\mathbf{G}_{\mathbf{P}_N'^*}$ la matrice de Gram dans l'espace réduit sans approximation et $\mathbf{G}_{\tilde{\mathbf{P}}_N^* \mathbf{R}_N}$ la matrice de Gram des descripteurs corrigés. L'expression de la matrice $\mathbf{G}_{\mathbf{P}_N'^*}$ est :

$$\mathbf{G}_{\mathbf{P}_N'^*} = \mathbf{X}^\top \mathbf{P}_N'^* \mathbf{P}_N'^*{}^\top \mathbf{X}. \quad (\text{IV.42})$$

avec $\mathbf{X}$ les signatures d'entraînements et $\mathbf{P}_N'^*$ la matrice de projection pleine. L'expression de la matrice $\mathbf{G}_{\tilde{\mathbf{P}}_N^* \mathbf{R}_N}$ est :

$$\mathbf{G}_{\tilde{\mathbf{P}}_N^* \mathbf{R}_N} = \mathbf{X}^\top \tilde{\mathbf{P}}_N^* \mathbf{R}_N \mathbf{R}_N^\top \tilde{\mathbf{P}}_N^{*\top} \mathbf{X}. \quad (\text{IV.43})$$

avec $\mathbf{X}$ les signatures d'entraînement, $\tilde{\mathbf{P}}_N^*$ la matrice de projection creuse et $\mathbf{R}_N$ la matrice de correction.

Pour obtenir la matrice de correction, nous résolvons donc le problème suivant :

$$\mathbf{W}^* = \arg \min_{\mathbf{W}} \|\mathbf{G}_{\mathbf{P}_N'^*} - \mathbf{Y}_{\tilde{\mathbf{P}}_N^*}^\top \mathbf{W} \mathbf{Y}_{\tilde{\mathbf{P}}_N^*}\|_F^2, \quad (\text{IV.44})$$

avec $\mathbf{Y}_{\tilde{\mathbf{P}}_N^*} = \tilde{\mathbf{P}}_N^{*\top} \mathbf{X}$ et $\mathbf{W} = \mathbf{R}_N \mathbf{R}_N^\top$. Ce problème est convexe et pour $N < L$, il a une solution analytique :

$$\mathbf{W}^* = \mathbf{Y}_{\tilde{\mathbf{P}}_N^*}^+ \mathbf{G}_{\mathbf{P}_N'^*} \mathbf{Y}_{\tilde{\mathbf{P}}_N^*}^{+\top}, \quad (\text{IV.45})$$

avec $\mathbf{Y}_{\tilde{\mathbf{P}}_N^*}^+ = (\mathbf{Y}_{\tilde{\mathbf{P}}_N^*} \mathbf{Y}_{\tilde{\mathbf{P}}_N^*}^\top)^{-1} \mathbf{Y}_{\tilde{\mathbf{P}}_N^*}$ la pseudo-inverse de la matrice $\mathbf{Y}_{\tilde{\mathbf{P}}_N^*}$. Nous obtenons la matrice de correction $\mathbf{R}_N^*$ en factorisant $\mathbf{W}^*$. Pour cela nous proposons d'utiliser la décomposition en valeurs et vecteurs propres :

$$\mathbf{R}_N^* = \mathbf{V} \mathbf{D}^{1/2}, \quad (\text{IV.46})$$

avec $\mathbf{V}$ les vecteurs propres de $\mathbf{W}^*$ et $\mathbf{D}$ les valeurs propres de $\mathbf{W}^*$.

IV.2.3.c Réduction de dimension avec une matrice de correction

Avec la matrice de correction, la réduction de dimension se fait en deux étapes :

$$\begin{aligned} \hat{\mathbf{y}}_i &= \hat{\mathbf{P}}_N^{*\top} \mathbf{x}_i \\ \mathbf{y}_i &= \mathbf{R}_N^{*\top} \hat{\mathbf{y}}_i. \end{aligned} \quad (\text{IV.47})$$

En premier, nous réduisons la signature par projection sur la matrice creuse. Puis, nous corrigeons la signature réduite par projection sur la matrice de correction. Ce traitement en deux étapes est indispensable et les deux matrices ne peuvent pas être recombinaées, puisque les recombinaisons des projecteurs ne donnent pas de projecteur creux.

Le coût calculatoire des deux projections est $\mathcal{O}((N + (1 - \tau) \times W) \times N)$. Pour des forts taux de parcimonie (*e.g.* $\tau = 99\%$), le coût de réduction est vraiment très faible en comparaison aux projecteurs denses. Le coût de stockage des deux matrices de projections est : $\mathcal{O}((N + 2 \times (1 - \tau) \times W) \times N)$.

IV.3 Expériences

Dans cette section, nous présentons notre protocole expérimental et l'évaluation de la méthode proposée.

IV.3.1 Recherche par similarité

Dans cette section, nous présentons les résultats obtenus en recherche par similarité avec les signatures réduites. Nous étudions trois méthodes différentes : la réduction de dimension par projection linéaire (présenté dans la section IV.2.1) ; la réduction de dimension par projection linéaire parcimonieuse (présenté dans la section IV.2.2) ; et les effets de la matrice de correction (présenté dans la section IV.2.3). Nous évaluons les performances des méthodes sur les bases Holidays (Annexe A.3.1) et Oxford (Annexe A.3.2).

IV.3.1.a Protocole expérimental

Chaque image est convertie en niveaux de gris puis, nous effectuons une égalisation d'histogramme comme pré-traitement. Nous utilisons le descripteur local HOG et le descripteur local couleur. Pour chaque descripteur, nous effectuons une extraction dense tous les 3 pixels sur 4 échelles (4, 6, 8 et 10 pixels par cellule). Par la suite, nous effectuons une normalisation ℓ_2 des descripteurs locaux.

Pour chaque image, nous calculons deux signatures VLAT avec le modèle optimisé : une signature avec le descripteur HOG ; et une autre avec le descripteur couleur. Les paramètres du modèle sont estimés avec 1 million de descripteurs extraits aléatoirement de 10000 images de la base Flickr60k. Nous utilisons des modèles avec 32, 64 et 256 parties, les signatures sont calculées pour deux cas : toutes les dimensions sont conservées, uniquement 80 dimensions sont conservées dans chaque partie du modèle. Nous effectuons sur chaque signature une normalisation en loi de puissance et une normalisation ℓ_2 .

Pour entraîner les méthodes de réduction de dimension et la méthode de correction, nous utilisons comme base d'entraînement 10000 signatures extraites d'images de la base Flickr60k.

Chaque signature réduite est pré-normalisée ℓ_2 avant le calcul des similarités. Puis, nous utilisons le produit scalaire usuel comme fonction de similarité.

Pour évaluer les performances des signatures dans des bases de grande dimension, nous utilisons les signatures extraites de la base Flickr1M. Ces signatures sont introduites parmi les signatures de la base de test lors de la recherche par similarité ajouter un bruit de mesure.

IV.3.1.b Projecteurs denses

Tout d'abord, nous proposons d'évaluer les performances des signatures réduites avec les projecteurs denses obtenus par notre méthode de reconstruction de la matrice de Gram (section IV.2.1).

Pour étudier l'influence des paramètres sur les performances des signatures, nous utilisons uniquement les signatures VLAT calculées avec le descripteur HOG.

Le Tableau IV.1 montre l'influence des différentes étapes de notre méthode sur le mAP. Les lignes représentent les différentes configurations des étapes de notre méthode, "VLAT" pour la signature VLAT non-réduite et "PVLAT" pour la signature VLAT avec l'optimisation du modèle et non-réduite. Le préfixe "C" définit les signatures réduites avec les projecteurs linéaires que nous avons proposé. Le suffixe "-M" définit l'utilisation du produit scalaire associé à la distance de Mahalanobis comme fonction de similarité. Chaque colonne représente la dimension N des signatures réduites ("FULL" sont les signatures non-réduites). Nous observons un gain de 2.4% entre la signature VLAT non-optimisée et

	32	64	128	256	9000	FULL
VLAT	-	-	-	-	-	64.0
PVLAT	-	-	-	-	-	66.4
CVLAT	46.3	49.6	53.3	54.9	58.2	-
CPVLAT	47.1	51.9	53.9	55.6	55.0	-
CVLAT-M	47.1	50.0	55.1	57.5	70.0	-
CPVLAT-M	48.5	54.3	57.3	60.6	70.0	-

TABLE IV.1 – Étude de l’influence des paramètres en recherche par similarité sur la base Holidays avec $D = 64$ (mAP).

la signature VLAT optimisée, ce qui met en évidence l’amélioration apportée par l’optimisation du modèle. Les lignes 3 et 4 du tableau montrent que le modèle optimisé permet de mieux conserver l’information pour les forts taux de réduction de dimension (typiquement $N \leq 256$). Nous pouvons constater que l’utilisation du produit scalaire associé avec la distance de Mahalanobis augmente fortement les performances avec les signatures réduites. Pour $D = 64$ et $N = 256$, nous divisons par 2000 la dimension des signatures avec une perte uniquement de 3.4% de mAP.

Pour étudier l’influence du nombre de parties D dans le modèle de la signature VLAT, nous fixons la dimension de la signature réduite à $N = 256$. La Figure IV.4 montre la variation de mAP en fonction de la taille de la base. Nous observons que pour des bases avec un petit nombre d’images, les performances sont meilleures avec un petit nombre de parties dans le modèle. Ceci met en évidence la limite imposée par le nombre d’exemples utilisés pour entraîner les projecteurs. En effet, plus la dimension de la signature est grande plus il est difficile d’apprendre les projecteurs. Cependant, les résultats deviennent similaires quand le nombre d’images dans la base augmente. Cela montre que même pour des grandes bases la signature VLAT réduite à 256 dimensions permet d’avoir de bonnes performances avec un petit dictionnaire visuel.

Maintenant, nous comparons notre méthode avec les résultats de [Jégou et al., 2012] sur la base Holidays étendue à 100k images et avec les résultats de [Jégou et al., 2012] sur la base Oxford étendue à 100k images.

Pour la base Holidays, nous utilisons la signature VLAT avec le modèle optimisé de 64 parties. Nous calculons les signatures réduites à $N = 96$ et $N = 256$ dimensions. Les résultats obtenus sont illustrés dans la Figure IV.5.

Comparé au BoW calculé avec un partitionnement de l’espace des descripteurs en 200k parties, nos signatures calculées avec un partitionnement beaucoup plus petit donnent de bien meilleures performances. Avec nos signatures réduites à $N = 96$ dimensions, nous avons un gain de mAP d’environ 5%, alors que notre signature est environ 2000 fois plus petite.

Comparé à la signature Fisher Vectors calculée avec un GMM de 64 Gaussiennes et avec une réduction à 96 dimensions des descripteurs locaux en utilisant une ACP, nous obtenons des meilleurs résultats avec le même nombre de parties dans le modèle et la même taille de signature.

Comparé à la signature Fisher Vectors avec un index IVFADC et un GMM de 256 Gaussiennes, nous obtenons de moins bons résultats pour une base de petite dimension. Cependant, cette signature est plus sensible à l’augmentation du nombre d’images dans la

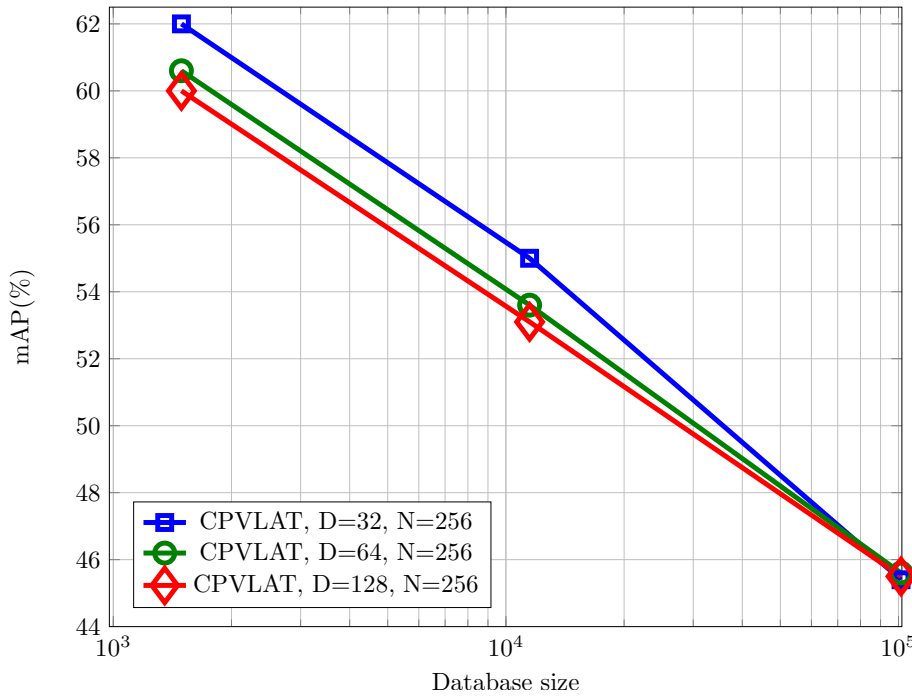


FIGURE IV.4 – Étude l'influence du nombre de parties D dans le modèle de la signature VLAT sur les performances en recherche par similarité en fonction du nombre d'images dans la base.

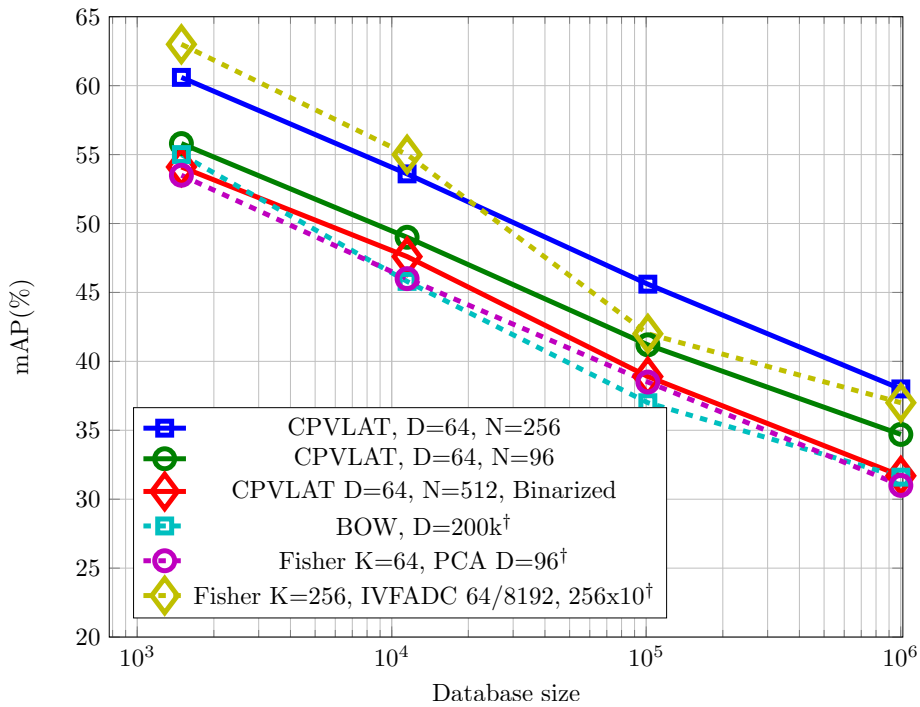


FIGURE IV.5 – Comparaison de notre signature avec les signatures de l'état de l'art en fonction du nombre d'images dans la base ([†] extrait [Jégou et al., 2012]).

	Oxford	Oxford + 100k
Fisher [Jégou et al., 2012]	31.7	-
VLAD [Jégou et al., 2012]	30.4	-
Fisher-PCA (N=128) [Jégou et al., 2012]	24.3	-
VLAD-PCA (N=128) [Jégou et al., 2012]	25.7	-
VLAD@HOG	36.6	-
VLAT@HOG	50.7	-
PVLAT@HOG	54.2	-
VLAD@HOG-PCA (N=128)	32.7	25.6
CPVLAT@HOG (N=128)	54.3	46.6

TABLE IV.2 – Performances en recherche par similarité de notre signature réduite par projection dense et des signatures de l’état de l’art la base Oxford et Oxford avec 100k distracteurs (mAP).

base. Pour plus de 10k images, nous avons de meilleurs résultats avec un partitionnement de l’espace des descripteurs plus petit.

Pour tester si notre méthode obtient de bonnes performances sur d’autres bases d’image, nous utilisons la base Oxford avec les paramètres des signatures par défaut. Nous calculons les signatures VLAD et VLAT avec le modèle optimisé sur les descripteurs HOG extraits de manière dense (tous les 3 pixels). Nous utilisons le même partitionnement de l’espace des descripteurs pour toutes les signatures. Pour compresser la signature VLAD, nous utilisons le même protocole que dans [Jégou et al., 2012]. Les résultats sont illustrés dans la Tableau IV.2.

Nous pouvons remarquer que l’utilisation des descripteurs HOG extraits de manière dense permet d’augmenter les performances de la signature VLAD de 6% de mAP. La méthode de compression proposée dans [Jégou et al., 2012] produit une perte de performance équivalente entre la signature VLAD obtenue par Jegou *et al.* dans [Jégou et al., 2012] et la signature VLAD obtenue avec l’extraction dense. Nous observons que la signature VLAT avec le modèle optimisé donne de bien meilleures performances que la signature VLAD. Avec ces paramètres, nous observons qu’à dimension équivalente, notre méthode a de bien meilleures performances sur les bases de grande taille que la méthode de l’état de l’art (une amélioration d’environ 20% mAP).

IV.3.1.c Projecteurs parcimonieux et matrice de correction

Maintenant, nous proposons d’évaluer les performances de nos projecteurs parcimonieux et l’apport de la matrice de correction (proposée en section IV.2.2 et IV.2.3).

Nous utilisons pour cela la signature VLAT avec une réduction des descripteurs locaux à 96 dimensions par partie dans le modèle et la signature Fisher Vector (FV) avec une pré-réduction à 96 dimensions des descripteurs. Nous utilisons les signatures calculées sur les descripteurs HOG et les descripteurs couleurs. Les signatures VLAT et FV finaux sont obtenues par la concaténation des signatures VLAT et FV obtenues avec les deux descripteurs locaux. Nous effectuons alors la normalisation en loi de puissance (de valeur 0.1) et la normalisation ℓ_2 sur les signatures concaténées.

La Tableau IV.3 montre les performances obtenues (en mAP) sur la base Holidays et

Sign.	Dim.	Holidays	Oxford
FV-64[Jégou et al., 2012]	4k	59.5	31.7
VLAD-64[Jégou et al., 2012]	4k	55.6	30.4
VLAD-64[Delhumeau et al., 2013]	8k	62.2	50.0
VLAD-64[Negrel et al., 2013]	8k	-	36.6
VLAT-64[Negrel et al., 2013]	528k	66.4	54.2
FV-64	20k	78.6	43.3
FV-256	82k	82.0	49.3
VLAT-64	415k	81.6	58.4
VLAT-256	1.7M	84.0	59.8

TABLE IV.3 – Performances en recherche par similarité de signature VLAT et des signatures de l’état de l’art la base Holidays et Oxford (mAP).

Oxford par les signatures non-réduites. Ce tableau est divisé en deux parties : dans la partie du haut, nous reportons les résultats obtenus dans l’état de l’art ; et, dans la partie du bas, nous reportons les résultats que nous avons obtenus avec notre implémentation. En comparaison avec les résultats de l’état de l’art, nous observons que nous obtenons de bien meilleures performances. Cela est principalement dû à notre utilisation de deux types de descripteurs locaux et également à l’utilisation d’un partitionnement de l’espace des descripteurs avec plus de parties que dans l’état de l’art. La signature VLAT avec 256 parties dans le modèle obtient les meilleures performances sur les deux bases, 84% de mAP sur Holidays et 59.8% de mAP sur Oxford mais elle est de très grande dimension (plus de 1.7×10^6).

Nous commençons par étudier l’influence des paramètres sur les performances en recherche par similarité avec la base Holidays.

Nous considérons le contexte des bases de grande dimension dans lesquelles la dimension finale des signatures doit être petite, nous fixons la dimension finale à $N = 256$. Nous calculons les projecteurs parcimonieux $\hat{\mathbf{P}}_N^*$ et la matrice de correction $\mathbf{R}_N^*$ pour plusieurs taux de parcimonie des projecteurs $\tau \in \{99.9\%, 99\%, 90\%, 0\%\}$ (notons que pour $\tau = 0\%$: $\hat{\mathbf{P}}_N^* = \mathbf{P}_N^*$ et $\mathbf{R}_N^* = \mathbf{I}$).

La Figure IV.6 montre l’évolution du mAP sur la base Holidays en fonction du coût calculatoire engendré par notre méthode de réduction de dimension. Ce coût est induit par les deux étapes de projections : la projection de la signature sur les projecteurs parcimonieux ; et la projection sur la matrice de correction. Les courbes en pointillé représentent les performances des signatures réduites sans correction. Les courbes continues représentent les performances des signatures réduites après l’étape de correction. Nous remarquons que pour des coûts calculatoires équivalents, les projecteurs de grande dimension sont moins sensibles aux forts taux de parcimonie. Nous voyons que l’étape de correction permet bien corriger les erreurs introduites par l’approximation parcimonieuse des projecteurs. De plus, nous observons que pour des forts taux de parcimonie, le gain de performance obtenu par l’étape de correction est plus grand. Nous observons également que le mAP évolue de la même manière pour tout type de signatures. Ainsi, notre méthode n’est pas sensible au choix de la signature utilisée. Par exemple, avec la signature FV avec 256 Gaussiennes et un taux de parcimonie de 90%, le coût calculatoire est divisé par un facteur d’environ 9 sans la moindre perte de performance. Avec la signature VLAT à 256 parties et un taux de parcimonie de 99.9%, le coût calculatoire est divisé par un facteur d’environ 1000 avec

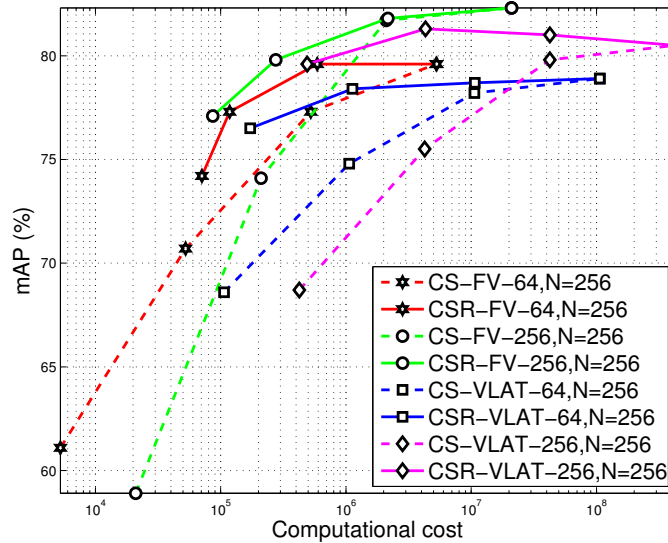


FIGURE IV.6 – Évolution du mAP en recherche par similarité avec la base Holidays en fonction du coût calculatoire de l'étape de réduction de dimension pour différents taux de parcimonie (*i.e.*, $\tau \in \{99.9\%, 99\%, 90\%, 0\%\}$).

une perte de mAP seulement de 0.9%.

Dans la suite, nous comparons les performances de notre méthode de réduction de dimension à faible coût avec les méthodes de l'état de l'art. Nous calculons pour cela les signatures FV et VLAT avec des modèles de 64 et 256 parties/Gaussiennes. Puis nous réduisons ces signatures à 128 dimensions. Nous calculons les performances de ces signatures sur les bases Holidays et Oxford avec et sans l'ajout de 100k distracteurs.

Les performances alors obtenues sont reportées dans le Tableau V.1 qui est divisé en trois parties. Dans la partie du haut, nous reportons plusieurs résultats obtenus dans les travaux de l'état de l'art. La partie du milieu illustre les résultats obtenus avec les projecteurs denses. Dans la partie du bas, nous reportons les résultats obtenus avec l'approximation parcimonieuse des projecteurs suivie de la matrice de correction.

Nous notons que notre méthode de réduction permet bien de préserver les performances des signatures originelles pour la base Holidays mais pour la base Oxford les performances sont dégradées. Cela est probablement causé par l'utilisation des descripteurs locaux couleur qui ne sont pas pertinents pour cette base.

Nous observons pour les deux bases, que la méthode de réduction parcimonieuse a la même robustesse que la méthode de réduction avec les projecteurs denses quand on ajoute les distracteurs. Cependant, notre méthode de projection parcimonieuse a le grand avantage d'avoir un plus faible coût de calculatoire et de stockage. Par exemple, dans le cas de la signature VLAT-256 avec une réduction par projection dense, la perte de mAP est de 6.4% sur la base Holidays et de 0.6% sur Oxford. Avec notre méthode de réduction par projecteurs creux, la perte de mAP est de 6.3% sur la base Holidays et de 1.0% sur la base Oxford avec une réduction du coût calculatoire d'un facteur d'environ 1000.

Name	Dim.	Holidays		Oxford	
			+100k		+100k
FV-64-PCA[Jégou et al., 2012]	128	56.5	38.0	24.3	-
VLAD-64-PCA[Jégou et al., 2012]	64	44.7	32.0	-	-
VLAD-64-PCA[Delhumeau et al., 2013]	128	-	-	32.5	26.6
C-VLAT-64[Negrel et al., 2012a]	256	72.3	58.0	-	-
VLAD-64-PCA[Negrel et al., 2013]	128	-	-	32.7	25.6
C-VLAT-64[Negrel et al., 2013]	128	57.3	-	54.3	46.6
C-FV-64	128	77.4	69.9	36.6	35.7
C-FV-256	128	79.9	74.0	38.0	37.2
C-VLAT-64	128	75.4	69.3	42.1	41.3
C-VLAT-256	128	78.2	71.8	39.0	38.4
CSR-FV-64, $\tau = 0.9$	128	77.5	69.9	35.9	34.7
CSR-FV-256, $\tau = 0.9$	128	80.6	73.9	37.8	36.9
CSR-VLAT-64, $\tau = 0.99$	128	75.6	68.8	40.9	39.9
CSR-VLAT-256, $\tau = 0.999$	128	76.9	70.6	35.6	34.6

TABLE IV.4 – Évaluation de la robustesse des signatures en recherche par similarité par ajout de 100k distracteurs et comparaison avec l'état de l'art dans les base Holidays et Oxford (mAP).

IV.3.2 Classification automatique

Dans cette section nous présentons les performance obtenus avec les signatures réduites avec nos projecteurs linéaires. Nous utilisons la base d'images VOC2007 (Annexe A.2.1).

IV.3.2.a Protocole expérimental

Chaque image est convertie en niveau de gris puis nous effectuons comme pré-traitement une égalisation d'histogramme. Nous utilisons le descripteur local HoG, nous effectuons une extraction dense des descripteurs locaux. Nous calculons un HoG tous les 3 pixels sur 4 échelles (4, 6, 8 et 10 pixels par cellule du HoG). Par la suite nous effectuons une normalisation ℓ_2 des descripteurs locaux.

Pour calculer les paramètres du modèle, nous utilisons 10000 images extraites de la base Flickr60k. Le partitionnement des données est effectué à l'aide d'un algorithme de type k-means sur un échantillon de millions de descripteurs HoG. La moyenne et la variance dans chaque partie du modèle sont estimées à partir des descripteurs utilisés pour obtenir le partitionnement. Pour chaque image, nous calculons la signature VLAT avec 64 parties dans le modèle et nous conservons 90% de l'énergie dans chaque partie. Chaque signature est post-traitée avec une normalisation en loi de puissance ($\alpha = 0.5$) et une normalisation ℓ_2 .

Pour entraîner les projecteurs de réduction de dimension, nous utilisons les 10000 images de la base VOC2007. Nous calculons les signatures réduites pour plusieurs valeurs de N . Nous effectuons une normalisation ℓ_2 de toute les signatures.

Nous utilisons un classificateur linéaire par hyperplan. Nous entraînons le classificateur avec le protocole un contre tous à l'aide d'un SVM à descente de gradient stochastique avec l'hyperparamètre du SVM $C = 10$.

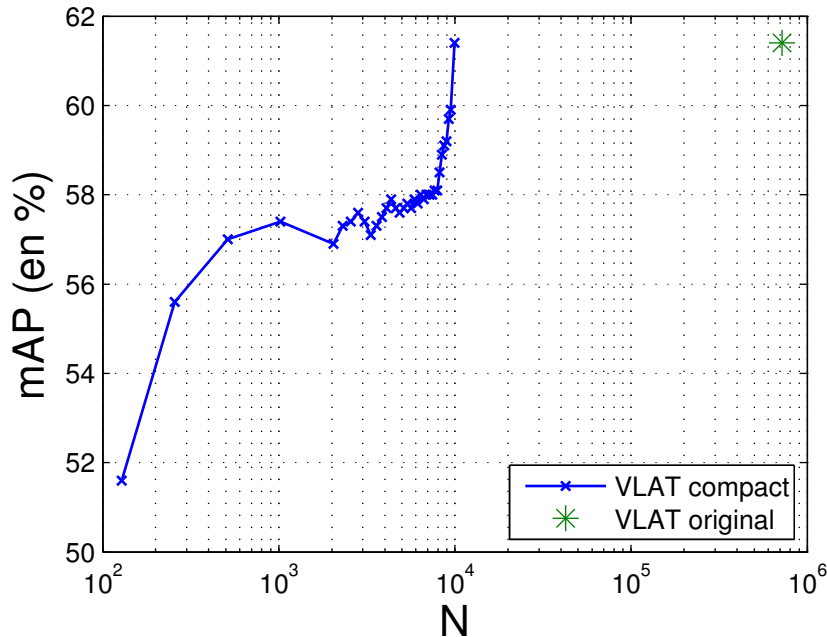


FIGURE IV.7 – Performance en mAP de la signature VLAT réduite en fonction de la dimensions de la signature N .

IV.3.2.b Étude des paramètres

La Figure IV.7 montre l'évolution des performances de classification sur la base VOC2007 en fonction de N la dimension de la signature réduite. L'étoile rouge est le mAP obtenu avec la signature VLAT non réduite, 61.4% de mAP pour $W = 718k$ dimensions. Nous constatons que quand nous conservons tous les projecteurs $N = 10000$ la signature réduite conserve exactement les mêmes performances que la signature d'origine. Il y a reconstruction parfaite des scores de similarité. Nous remarquons que nous pouvons encore diviser la dimensions de la signature par 10 et conserver de bonnes performances. Nous obtenons une mAP de 57.0% pour une signature de $N = 1024$ dimensions soit une perte de 4.4% de mAP.

IV.4 Conclusions

Nous avons présenté dans ce chapitre une nouvelle méthode pour réduire la dimension des signatures basées sur la conservation de la similarité entre les signatures. Nous avons proposés d'utiliser une projection linéaire pour réduire la dimension des signatures. Les projecteurs sont obtenus par un apprentissage non supervisé permettant de conserver les valeurs des produits scalaires entre les signatures d'origine et les signatures réduites. Pour réduire les coûts induits par la grande dimension des projecteurs, nous avons proposé d'introduire une contrainte de parcimonie pour obtenir des projecteurs creux. Il est très complexe d'obtenir la solution exacte aux problèmes d'optimisations avec une contrainte de parcimonie stricte. Nous avons alors proposé une méthode permettant d'obtenir simplement une approximation de la solution. Cette méthode se déroule en deux étapes :

le calcul d'une approximation parcimonieuse des projecteurs pleins ; puis le calcul d'une matrice de correction permettant de corriger les erreurs introduites par l'approximation parcimonieuse.

Les résultats expérimentaux ont montré que les projecteurs pleins obtenus permettent de fortement réduire la dimension de la signature tout en conservant les performances de la signature d'origine. Nous avons montré que l'utilisation de projecteurs creux permet de fortement réduire les coûts induits par la projection des signatures dans l'espace de petite dimension. Nous avons également constaté que la projection en deux étapes avec les projecteurs creux puis la matrice de correction permet de fortement réduire les coûts de l'étape de réduction tout en conservant de très bonnes performances.

Réduction de dimension supervisée

Dans le chapitre précédent, nous avons présenté un ensemble de méthodes pour réduire la dimension des signatures d'images basées sur un apprentissage non-supervisé. Dans ce chapitre, nous nous intéressons aux méthodes supervisées de réduction de dimension. Ces méthodes s'appuient sur un ensemble d'exemples annotés pour effectuer l'apprentissage des paramètres de la fonction de projection. Dans le cas de la réduction de signature pour l'indexation multimédia, nous utilisons pour cela un ensemble de signatures et un ensemble d'annotations (vérité terrain).

Nous présentons en premier lieu l'état de l'art sur les méthodes basées sur l'apprentissage de métrique et de fonction de similarité pour l'indexation (*i.e.* les applications de partitionnement, les applications de classification par la méthode des k -Plus Proches Voisins). Puis, nous proposons une méthode basée sur l'apprentissage de fonction de similarité pour obtenir les projecteurs de réduction de dimension. Pour l'apprentissage de la fonction de similarité, nous utilisons une base d'images de recherche de quasi-copie pour construire des contraintes sur la fonction de similarité. Dans la troisième section nous évaluons les performances des signatures réduites en recherche de quasi-copie.

V.1 État de l'art

Dans cette section, nous présentons l'état de l'art sur les méthodes d'apprentissage de métrique et de similarité, ainsi que leurs applications pour la réduction de dimension de signature. Nous présentons tout d'abord l'apprentissage de métrique, puis nous présentons l'apprentissage de fonctions de similarité.

V.1.1 Métriques

Une métrique (ou distance) est une fonction permettant de comparer deux objets dans un espace métrique. Les fonctions métriques sont de la forme suivante :

$$\begin{aligned} d: \mathcal{X} \times \mathcal{X} &\rightarrow \mathbb{R} \\ (\mathbf{x}, \mathbf{y}) &\mapsto d(\mathbf{x}, \mathbf{y}) \end{aligned} \quad (\text{V.1})$$

avec $\mathcal{X}$ un espace quelconque. Une métrique au sens mathématique du terme doit respecter les conditions suivantes :

- Contrainte de positivité : $\forall (x, y) \in \mathcal{X} \times \mathcal{X}, d(\mathbf{x}, \mathbf{y}) \geq 0$.
- Contrainte d'égalité : $\forall (x, y) \in \mathcal{X} \times \mathcal{X}, d(\mathbf{x}, \mathbf{y}) = 0 \implies \mathbf{x} = \mathbf{y}$.
- Contrainte de symétrie : $\forall (x, y) \in \mathcal{X} \times \mathcal{X}, d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$.
- Inégalité triangulaire : $\forall (x, y, z) \in \mathcal{X} \times \mathcal{X} \times \mathcal{X}, d(\mathbf{x}, \mathbf{z}) \geq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$.

Cependant, dans de nombreux problèmes la métrique considérée ne respecte pas toutes ces contraintes. Quand la contrainte d'égalité n'est pas respectée, nous parlons alors de pseudo-métrique mais sans cette propriété, il est impossible de déterminer si $\mathbf{x}_i$ et $\mathbf{x}_j$ sont identique ou non. Quand la contrainte de symétrie n'est pas respectée, nous parlons alors de quasi-métrique, mais sans cette propriété, si $\mathbf{x}_i$ est le point le plus proche de $\mathbf{x}_j$ il est impossible de garantir que $\mathbf{x}_j$ est le point le plus proche de $\mathbf{x}_i$. Ou encore quand la contrainte d'inégalité triangulaire n'est pas respectée, nous parlons alors de semi-métrique.

V.1.1.a Apprentissage de métrique

Les méthodes d'apprentissage supervisé de métrique permettent de construire des métriques qui ont certaines propriétés. Les propriétés recherchées sont définies à partir du domaine d'application de la métrique. Par exemple pour les méthodes de classification basées sur la recherche des k -Plus Proches Voisins, il est intéressant de construire une métrique qui minimise la distance entre les documents appartenant à la même classe tout en maximisant la distance entre les documents de classe différente. L'apprentissage supervisée de métrique consiste à construire un problème d'optimisation visant à apprendre les propriétés recherchées à partir d'un ensemble d'exemples annotés. Le problème d'optimisation est généralement formalisé sous cette forme :

$$\arg \min_{\mathbf{A}} L(d_{\mathbf{A}}, \mathbf{X}) \quad (\text{V.2})$$

avec $d_{\mathbf{A}}$ la métrique de paramètre $\mathbf{A}$, $\mathbf{X}$ les données d'apprentissage et L le critère à optimiser.

V.1.1.b Distance de Mahalanobis

Une méthode bien connue en apprentissage de métrique s'appuie sur la distance de Mahalanobis [Mahalanobis, 1936]. C'est une extension de la distance euclidienne qui permet de prendre en compte la variance et la corrélation des composantes des données. Elle est définie comme suit :

$$d_{\mathbf{W}}(\mathbf{x}, \mathbf{y}) = ((\mathbf{x} - \mathbf{y})^{\top} \mathbf{W} (\mathbf{x} - \mathbf{y}))^{1/2} \quad (\text{V.3})$$

Avec $\mathbf{W}$ la matrice carrée des paramètres de la distance. La matrice $\mathbf{W}$ est obtenue par un apprentissage non-supervisé sur un échantillon de données. Pour la distance de

Mahalanobis, la matrice $\mathbf{W}$ est obtenue par inversion de la matrice de covariance Σ des données :

$$\mathbf{W} = \Sigma^{-1} = \left[\frac{1}{|\mathcal{T}|} \sum_{\mathbf{x} \in \mathcal{T}} (\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^\top \right]^{-1} \quad (\text{V.4})$$

avec $\mathcal{T}$ l'ensemble des données d'entraînement, $|\mathcal{T}|$ le nombre de données dans l'ensemble $\mathcal{T}$, et $\boldsymbol{\mu}$ la moyenne des données d'entraînement.

Cette distance est souvent utilisée en apprentissage supervisé de métrique [Xing et al., 2002, Davis et al., 2007, Guillaumin et al., 2009, Weinberger and Saul, 2009, Mignon and Jurie, 2012]. L'apprentissage supervisé consiste alors à optimiser le paramètre $\mathbf{W}$, tel que la métrique respecte l'ensemble des contraintes induites par la vérité terrain. De plus pour que (V.3) soit une métrique, une seule contrainte doit être respectée : $\mathbf{W} \succeq 0$ doit être semi-définie positive. Le problème d'optimisation s'écrit alors :

$$\begin{aligned} \arg \min_{\mathbf{W}} L(d_{\mathbf{W}}, \mathbf{X}) \\ \text{s.t. } \mathbf{W} \succeq 0 \end{aligned} \quad (\text{V.5})$$

Notons que le nombre de paramètres de cette métrique (nombre de valeurs dans la matrice $\mathbf{W}$) évolue quadratiquement avec la dimension des données. Par exemple, pour des données de 10^5 dimensions, la matrice $\mathbf{W}$ contient 10^{10} de paramètres.

Pour diminuer le nombre de paramètres, deux approches peuvent être utilisées : l'ajout d'une contrainte de parcimonie sur la matrice $\mathbf{W}$; et la décomposition de la matrice $\mathbf{W}$ en deux matrices de plus petite dimension (ce qui contraint le rang de $\mathbf{W}$). La décomposition de $\mathbf{W}$ s'écrit alors comme suit :

$$\mathbf{W} = \mathbf{P}^\top \mathbf{P} \quad (\text{V.6})$$

avec $\mathbf{P}$ une matrice de dimensions $N \times W$ (avec $W \gg N$). Cette décomposition est très intéressante car la métrique (V.3) peut être reformulée comme suit :

$$d_{\mathbf{P}}(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^\top \mathbf{P} \mathbf{P}^\top (\mathbf{x} - \mathbf{y}) \quad (\text{V.7})$$

$$= (\mathbf{P}^\top \mathbf{x} - \mathbf{P}^\top \mathbf{y})^\top (\mathbf{P}^\top \mathbf{x} - \mathbf{P}^\top \mathbf{y}) \quad (\text{V.8})$$

$$= d(\mathbf{P}^\top \mathbf{x}, \mathbf{P}^\top \mathbf{y}) \quad (\text{V.9})$$

avec $d(\cdot, \cdot)$ la distance euclidienne. Nous pouvons alors effectuer une réduction de la dimension des données par projection linéaire via la matrice $\mathbf{P}$ et utiliser la distance euclidienne comme métrique.

Xing *et al.* proposent une méthode d'apprentissage de métrique basée sur des couples d'exemples pour des applications de partitionnement. Ils proposent de minimiser la distance entre exemples similaires, tout en séparant par une marge les couples dissimilaires [Xing et al., 2002]. Guillaumin *et al.* proposent une méthode d'apprentissage de métrique basée sur le maximum de vraisemblance pour la reconnaissance de visage [Guillaumin et al., 2009]. Davis *et al.* proposent l'utilisation de critères de la théorie de l'information pour l'apprentissage de métrique dans le cadre de la classification par la méthode des k -Plus Proches Voisins [Davis et al., 2007]. Mignon *et al.* proposent une méthode d'apprentissage de métrique dans un contexte où le nombre d'exemples d'apprentissage est restreint [Mignon and Jurie, 2012]. Weinberger *et al.* proposent une méthode d'apprentissage de métrique basée sur des triplet d'exemples dans le cadre de la classification par la méthode des k -Plus Proches Voisins [Weinberger et al., 2005, Weinberger

and Saul, 2009]. Law *et al.* proposent des méthodes d'apprentissage de métrique basée sur des triplés d'exemples et une nouvelle méthode permettant d'obtenir une contrainte stricte sur le rang de la matrice des paramètres. Ces méthodes sont appliquées en reconnaissance de visage [Law et al., 2013, Law et al., 2014].

Des méthodes d'apprentissage de métrique basées sur d'autres métriques que Mahalanobis ont également été proposées. Par exemple, Kedem *et al.* proposent d'utiliser une distance sur des histogrammes comme la distance du χ^2 [Kedem et al., 2012].

V.1.2 Apprentissage de similarité

L'apprentissage de similarité est similaire à l'apprentissage de métrique, Cela consiste à optimiser les paramètres de la fonction de similarité afin d'obtenir des propriétés intéressantes. L'apprentissage de la similarité peut être non-supervisé, comme les méthodes basées sur la conservation du produit scalaire présentées dans la section IV.2. Dans cette section, nous nous intéressons aux méthodes d'apprentissage supervisé de similarité. Comme pour l'apprentissage de métrique, l'apprentissage de similarité utilise un ensemble de données externes (vérité terrain/annotation) pour optimiser un critère sur la fonction de similarité. Le problème s'écrit alors comme suit :

$$\arg \min_{\mathbf{A}} L(\mathbf{s}_{\mathbf{A}}, \mathbf{X}) \quad (\text{V.10})$$

avec $\mathbf{s}_{\mathbf{A}}$ la fonction de similarité de paramètre $\mathbf{A}$, $\mathbf{X}$ les données d'apprentissage et L le critère à optimiser.

V.1.2.a Produit scalaire associé à la distance de Mahalanobis

La fonction de similarité couramment utilisée pour l'apprentissage est le produit scalaire associé à la distance de Mahalanobis (V.3). Ce produit scalaire s'écrit :

$$\mathbf{s}_{\mathbf{W}}(\mathbf{x}, \mathbf{y}) = \mathbf{x}^{\top} \mathbf{W} \mathbf{y} \quad (\text{V.11})$$

avec $\mathbf{W}$ la matrice carrée des paramètres du produit scalaire. Comme pour la distance de Mahalanobis, la matrice $\mathbf{W}$ est l'inverse de la matrice de covariance des données (V.4). Cela permet de prendre en compte la variance et la corrélation des composantes des données.

De nombreuses méthodes proposent d'utiliser une fonction de similarité de cette forme (V.11), mais d'effectuer l'apprentissage la matrice de paramètre $\mathbf{W}$ avec une méthode supervisée. Comme pour la métrique associée, pour que la fonction (V.11) soit une fonction de similarité, la matrice de paramètre $\mathbf{W}$ doit être semi-définie positive. Le problème d'optimisation s'écrit alors :

$$\begin{aligned} \arg \min_{\mathbf{W}} L(\mathbf{s}_{\mathbf{W}}, \mathbf{X}) \\ \text{s.t. } \mathbf{W} \succeq 0 \end{aligned} \quad (\text{V.12})$$

Comme pour la métrique associée, le nombre de paramètres évolue quadratiquement en fonction de la dimension des données. Les mêmes méthodes que pour l'apprentissage de métrique peuvent être utilisées pour réduire le nombre de paramètres.

La méthode qui nous intéresse ici consiste à décomposer la matrice $\mathbf{W}$ en deux (ce qui revient à contraindre le rang de $\mathbf{W}$) :

$$\mathbf{W} = \mathbf{P}^{\top} \mathbf{P} \quad (\text{V.13})$$

avec $\mathbf{P}$ une matrice de dimensions $N \times W$ (avec $W \gg N$). Cela permet d'effectuer une réduction de dimension de la signature par projection linéaire :

$$s_{\mathbf{P}}(\mathbf{x}, \mathbf{y}) = \mathbf{x}^{\top} \mathbf{P} \mathbf{P}^{\top} \mathbf{y} \quad (\text{V.14})$$

$$= (\mathbf{P}^{\top} \mathbf{x})^{\top} (\mathbf{P}^{\top} \mathbf{y}) \quad (\text{V.15})$$

$$= s(\mathbf{P}^{\top} \mathbf{x}, \mathbf{P}^{\top} \mathbf{y}) \quad (\text{V.16})$$

avec $s(\cdot, \cdot)$ le produit scalaire usuel.

V.1.2.b Reconstruction du rang

L'apprentissage de reconstruction de rang est utilisé en recherche par similarité. Nous rappelons que les méthodes de recherche par similarité consistent à trier dans l'ordre décroissant de similarité par rapport à l'image requête. Le rang d'une image est alors sa position dans le tri pour cette requête. L'apprentissage de fonction de similarité pour la reconstruction du rang consiste alors à construire une fonction de similarité telle que pour chaque requête toutes les images ont le rang désiré [Chechik et al., 2010, Bai et al., 2009, Gordoa et al., 2012].

Par exemple, pour chaque image requête $q \in \mathcal{Q}$, nous avons une fonction de vérité terrain $\text{rang}_q(\cdot)$ qui pour chaque rang r retourne l'indice de l'image. Le but est alors d'apprendre à respecter les contraintes suivantes :

$$s_{\mathbf{W}}(\mathbf{x}_q, \mathbf{x}_{\text{rang}_q(1)}) > s_{\mathbf{W}}(\mathbf{x}_q, \mathbf{x}_{\text{rang}_q(2)}) > \dots > s_{\mathbf{W}}(\mathbf{x}_q, \mathbf{x}_{\text{rang}_q(L)}), \quad (\text{V.17})$$

il y a donc au total $|\mathcal{Q}| \times (L - 1)$ contraintes à satisfaire avec $|\mathcal{Q}|$ le nombre de requêtes. Pour apprendre une fonction de similarité qui respecte ces contraintes, nous construisons une fonction objective à minimiser telle que l'optimum de cette fonction respecte toutes les contraintes.

Pour construire cette fonction objectif, chaque contrainte d'inégalité : $s_{\mathbf{W}}(\mathbf{x}_q, \mathbf{x}_i) > s_{\mathbf{W}}(\mathbf{x}_q, \mathbf{x}_j)$ est convertie en une fonction objectif :

$$h_{qij}(\mathbf{W}) = [s_{\mathbf{W}}(\mathbf{x}_q, \mathbf{x}_j) - s_{\mathbf{W}}(\mathbf{x}_q, \mathbf{x}_i) + \epsilon]_+ \quad (\text{V.18})$$

avec $[x]_+ = \max(0, x)$ et ϵ une marge qui est utilisée pour convertir l'inégalité inférieure stricte en une inégalité inférieure ou égale. Nous pouvons alors poser le problème d'optimisation suivant :

$$\begin{aligned} \mathbf{W}^* = \arg \min_{\mathbf{W}} \sum_{q \in \mathcal{Q}} \sum_{i=1}^{L-1} [s_{\mathbf{W}}(\mathbf{x}_q, \mathbf{x}_{\text{rang}_q(i+1)}) - s_{\mathbf{W}}(\mathbf{x}_q, \mathbf{x}_{\text{rang}_q(i)}) + 1]_+ \\ \text{s.t. } \mathbf{W} \succeq 0 \end{aligned} \quad (\text{V.19})$$

avec $\mathcal{Q}$ l'ensemble des requêtes, $\{\mathbf{x}_i\}_{i=1}^L$ l'ensemble des signatures des L images.

Ce problème est généralement résolu par une descente de gradient stochastique avec une projection sur les contraintes. La descente de gradient stochastique est effectuée sur le tirage aléatoire d'une image requête q et le tirage aléatoire d'un couple d'image $(i, i + 1)$. Si nous utilisons comme fonction de similarité le produit scalaire associé à la distance de Mahalanobis $s_{\mathbf{W}}(\mathbf{x}_q, \mathbf{x}_i) = \mathbf{x}_q^{\top} \mathbf{W} \mathbf{x}_i$, le gradient de la fonction $h_{\mathbf{x}_q, \mathbf{x}_i, \mathbf{x}_j}(\mathbf{W})$ s'écrit :

$$\frac{\partial h_{qij}(\mathbf{W})}{\partial \mathbf{W}} = h(s_{\mathbf{W}}(\mathbf{x}_q, \mathbf{x}_j) - s_{\mathbf{W}}(\mathbf{x}_q, \mathbf{x}_i) + \epsilon) \mathbf{x}_q (\mathbf{x}_j - \mathbf{x}_i)^{\top}, \quad (\text{V.20})$$

avec $h(\cdot)$ la fonction de Heaviside. Le nombre de tirages à effectuer en moyenne pour prendre en compte toutes les contraintes à satisfaire est $Q \times L$ avec Q le nombre de requêtes dans $\mathcal{Q}$.

Cette approche permet de bien comprendre le problème de reconstruction de rang, mais il est rare d'avoir une vérité terrain aussi précise. En général, la vérité terrain d'une base d'images en recherche par similarité est constituée d'un ensemble de requêtes $\mathcal{Q}$. Pour chaque requête $q \in \mathcal{Q}$, deux ensembles d'images sont définis : un ensemble d'images similaires à la requête $\mathcal{S}_q$; et un ensemble d'images dissimilaires à la requête $\mathcal{D}_q$. Il est important de noter que ces deux ensembles ne sont pas ordonnés. Il est donc impossible de construire l'ensemble de contraintes (V.17).

Avec cette vérité terrain, les contraintes de rang qui peuvent être définies pour chaque requête $q \in \mathcal{Q}$ sont alors :

$$\{s_{\mathbf{W}}(\mathbf{x}_q, \mathbf{x}_i) > s_{\mathbf{W}}(\mathbf{x}_q, \mathbf{x}_j)\}_{i \in \mathcal{S}_q, j \in \mathcal{D}_q}, \quad (\text{V.21})$$

les scores de similarité entre la requête et les images similaires doivent être strictement supérieurs aux scores de similarité entre la requête et les images dissimilaires. Le nombre de contraintes à satisfaire est alors de :

$$\sum_{q \in \mathcal{Q}} |\mathcal{S}_q| \times |\mathcal{D}_q| \quad (\text{V.22})$$

avec $|\cdot|$ le nombre d'images dans l'ensemble.

V.1.2.c Méthode "*Supervised Semantic Indexing*"

Bai *et al.* ont proposé d'apprendre une fonction de similarité de manière à respecter pour chaque requête q les contraintes définies par l'équation (V.21) [Bai et al., 2009, Gordoa et al., 2012]. Ils proposent également de ne plus apprendre directement la matrice $\mathbf{W}$ mais sa décomposition : $\mathbf{W} = \mathbf{P}^\top \mathbf{P}$.

Pour cela, ils proposent de résoudre le problème suivant :

$$\mathbf{P}^* = \arg \min_{\mathbf{P} \in \mathcal{M}_{N \times W}} \sum_{q \in \mathcal{Q}} \sum_{i \in \mathcal{S}_q} \sum_{j \in \mathcal{D}_q} [s_{\mathbf{P}^\top \mathbf{P}}(\mathbf{x}_q, \mathbf{x}_j) - s_{\mathbf{P}^\top \mathbf{P}}(\mathbf{x}_q, \mathbf{x}_i) + 1]_+ \quad (\text{V.23})$$

Ils proposent de résoudre ce problème par une descente de gradient stochastique. A chaque itération, ils tirent aléatoirement un triplet : une image requête $q \in \mathcal{Q}$, une image similaire à la requête $\mathbf{x}_i \in \mathcal{S}_q$ et une image dissimilaire à la requête $\mathbf{x}_j \in \mathcal{D}_q$. Si ce tirage viole la contrainte qui lui est associé : $s_{\mathbf{W}}(\mathbf{x}_q, \mathbf{x}_i) \geq s_{\mathbf{W}}(\mathbf{x}_q, \mathbf{x}_j) + 1$, la matrice $\mathbf{P}$ est mise à jour avec l'équation suivante :

$$\mathbf{P} \leftarrow \mathbf{P} + \lambda \mathbf{P} \mathbf{x}_q (\mathbf{x}_i - \mathbf{x}_j)^\top \quad (\text{V.24})$$

avec λ le pas d'apprentissage. Le nombre moyen d'itérations pour prendre en compte toutes les contraintes à satisfaire est $|\mathcal{Q}| \times |\mathcal{S}| \times |\mathcal{D}|$.

V.2 Méthode proposée

Dans cette section, nous proposons un ensemble de méthodes supervisées pour l'apprentissage de similarité et la réduction de dimension. Ces méthodes sont basées sur le produit scalaire associé à distance de Mahalanobis (V.11), et la reconstruction du rang.

Nous utilisons une méthode pour réduire le nombre de contraintes de rang utilisées pour l'apprentissage afin de réduire le temps d'entraînement de la fonction de similarité. Nous proposons également d'introduire une contrainte de parcimonie pour obtenir des projecteurs creux.

V.2.1 Problème d'optimisation pour la reconstruction du rang

Les méthodes d'apprentissage de fonction de similarité pour la reconstruction de rang de l'état de l'art utilisent l'ensemble des contraintes définies par l'équation (V.21). Le principal problème de cet ensemble de contraintes est sa cardinalité (equation (V.22)). Quand la base d'images contient un grand nombre d'images, il devient impossible de prendre en compte toutes les contraintes lors de l'apprentissage. Si nous prenons l'exemple une base de 10^5 images avec 10^3 requêtes et pour chaque requête l'ensemble des images positives représente 10% des images de la base et l'ensemble des images négatives le reste, le nombre totale de contraintes a prendre en compte est alors de $\sim 10^{12}$. Nous proposons un méthodes basé sur l'introduction d'une image pivot pour chaque requête. Si nous reprenons notre exemple, le nombre de contraintes avec notre méthode est alors de 10^8 . Nous proposons également d'introduire une contrainte de parcimonie pour obtenir des projecteurs creux.

V.2.1.a Introduction d'images pivots pour déduire le nombre de contraintes

Nous proposons de reformuler les contraintes de rangs (V.21) afin d'en réduire le nombre. Pour cela, nous proposons d'introduire pour chaque requête q une image pivot que nous notons p_q . L'image pivot est choisie parmi l'ensemble des images dissimilaires à la requête : $p_q \in \mathcal{D}_q$. L'ensemble des contraintes peut alors s'écrire :

$$\begin{cases} \{\mathbf{sP}(\mathbf{x}_q, \mathbf{x}_i) > \mathbf{sP}(\mathbf{x}_q, \mathbf{x}_{p_q})\}_{i \in \mathcal{P}_q}, \\ \{\mathbf{sP}(\mathbf{x}_q, \mathbf{x}_{p_q}) \geq \mathbf{sP}(\mathbf{x}_q, \mathbf{x}_j)\}_{j \in \mathcal{N}_q \setminus p_q}. \end{cases} \quad (\text{V.25})$$

Pour toutes les images similaires à la requête, les scores de similarité à la requête doivent être plus grands que le score de similarité entre la requête et le pivot. Et pour toutes les images dissimilaires à la requête, les scores de similarité à la requête doivent être plus petits ou égal aux scores de similarité entre la requête et le pivot. Si ces contraintes sont respectées, alors les contraintes d'origine sont respectées.

Avec l'introduction de l'image pivot, le nombre de contraintes par requête à satisfaire est alors bien inférieur au nombre de contraintes d'origines. En effet, le nombre total de contraintes est de :

$$\sum_{q \in \mathcal{Q}} (|\mathcal{S}_q| + |\mathcal{D}_q| - 1). \quad (\text{V.26})$$

V.2.1.b Fonction objectif du problème d'optimisation

Nous pouvons alors définir un problème d'optimisation qui a un nombre plus faible de contraintes. Pour apprendre la fonction de similarité $\mathbf{sP}(\cdot, \cdot)$ qui tend à respecter l'ensemble des contraintes, nous définissons pour chaque requête q une fonction objectif à minimiser :

$$f_q(\mathbf{P}) = \alpha_q \sum_{i \in \mathcal{P}_q} [\varepsilon + \mathbf{sP}(\mathbf{x}_q, \mathbf{x}_i - \mathbf{x}_{p_q})]_+^2 + \beta_q \sum_{j \in \mathcal{N}_q \setminus p_q} [\mathbf{sP}(\mathbf{x}_q, \mathbf{x}_{p_q} - \mathbf{x}_j)]_+^2 \quad (\text{V.27})$$

avec

$$\alpha_q = \frac{1}{\sum_{i \in \mathcal{P}_q} h(\varepsilon + \mathbf{sP}(\mathbf{x}_q, \mathbf{x}_i - \mathbf{x}_{p_q}))}, \quad (\text{V.28})$$

$$\beta_q = \frac{1}{\sum_{j \in \mathcal{N}_q \setminus \mathcal{P}_q} h(\mathbf{sP}(\mathbf{x}_q, \mathbf{x}_{p_q} - \mathbf{x}_j))}. \quad (\text{V.29})$$

et $h(\cdot)$ la fonction de Heaviside et ε la marge (*e.g.*, $\varepsilon = 10^{-6}$).

Les coefficients de pondération α_q et β_q permettent de donner le même poids aux erreurs de rang des images similaires et des images dissimilaires. En effet, en recherche de quasi-copie il y a un fort déséquilibre entre les images similaires et les images dissimilaires $\mathcal{P}_q \ll \mathcal{N}_q$. Sans cet équilibrage les contraintes correspondantes aux images dissimilaires sont prépondérantes, les projecteurs alors obtenus permettent de respecter toutes les contraintes correspondantes aux images dissimilaires mais très peu de contraintes correspondantes aux images similaires. Les projecteurs alors obtenus ne permettent de respecté qu'un faible nombres de contraintes de l'ensemble de contraintes d'origines (V.21).

Nous utilisons la fonction de coût $([\cdot]_+)$ au carré pour donner plus d'importance aux images qui ont la plus grande erreur de rang car ce sont les images qui ont le plus d'influence sur le résultat de l'évaluation de la méthode de recherche par similarité (mAP). Nous remarquons qu'en fonction de la marge, l'ensemble des contraintes peut être respecté sans que la fonction objectif n'ait atteinte la valeur 0. Le problème d'optimisation global s'écrit de la manière suivante :

$$\mathbf{P}^* = \arg \min_{\mathbf{P}} \sum_{q \in \mathcal{Q}} f_q(\mathbf{P}) \quad (\text{V.30})$$

V.2.2 Résolution par descente de gradient stochastique

Pour résoudre ce problème, nous proposons d'utiliser une descente de gradient stochastique (SGD) [Bottou, 2004, Shalev-Shwartz et al., 2007]. La fonction objectif de notre problème (V.30) est constituée de la somme des fonctions objectifs de chaque requête $q \in \mathcal{Q}$ (V.27). Nous proposons alors de calculer le gradient uniquement sur un sous ensemble $\tilde{\mathcal{Q}}^{[t]} \subset \mathcal{Q}$ de requêtes.

Notons que le coût calculatoire du gradient est élevé, car pour chaque contrainte violée il est nécessaire de calculer un gradient de la dimension de la matrice de projection $\mathbf{P}_N \in \mathcal{M}_{N \times W}$. Cependant le coût calculatoire de la fonction objectif est très faible, car elle peut être calculée directement sur la matrice de Gram des signatures réduites. Nous proposons donc d'utiliser des heuristiques basées sur la fonction objectif pour accélérer la vitesse de convergence de l'algorithme.

Pour accélérer la vitesse de convergence de l'algorithme de descente de gradient stochastique, nous proposons d'utiliser plusieurs améliorations :

- l'utilisation de projecteurs initiaux proche du point optimal ;
- une heuristique de sélection des images pivots ;
- une heuristique de sélection de l'ensemble $\tilde{\mathcal{Q}}^{[t]}$;
- le calcul du pas $\mu^{[t]}$.

L'algorithme 1 montre les étapes de notre algorithme de descente de gradient.

Nous arrêtons la descente de gradient quand la valeur de la fonction objectif est proche de la précision numérique (*e.g.*, 10^{-16}) ou quand le nombre maximum d'itérations est atteint (*e.g.*, 2000).

Algorithm 1 Apprentissage de similarité pour la reconstruction de rang.

```

 $t \leftarrow 0.$ 
Initialisation de la matrice  $\mathbf{P}^{[0]}$  avec les projecteurs proposés section IV.2.
Choix des images pivots  $p_{q \in \mathcal{Q}}$ .
répéter
  Sélection du sous ensemble  $\tilde{\mathcal{Q}}^{[t]} \subset \mathcal{Q}$  de  $P$  requêtes.
  Calcul du gradient sur l'ensemble de requête  $\tilde{\mathcal{Q}}^{[t]}$ .
  Calcul du pas optimale  $\mu^{[t]}$ .
  Mise à jour de la matrice des projecteurs  $\mathbf{P}^{[t+1]}$ .
   $t \leftarrow t + 1.$ 
jusqu'à Fonction objectif  $< 10^{-12}$  ou  $t > \text{maxIter}$ 

```

V.2.2.a Projecteurs initiaux

Pour initialiser la descente de gradient, Bing *et al.* proposent d'initialiser la matrice $\mathbf{P}^{[0]}$ aléatoirement [Bai et al., 2009]. Nous proposons d'utiliser comme matrice initiale $\mathbf{P}^{[0]}$ la matrice de projecteurs permettant de conserver le produit scalaire que nous avons présenté dans la section IV.2. Cette matrice de projection permet une forte réduction de la dimensionnalité de la signature tout en conservant de très bonnes performances en recherche par similarité. Le nombre de contraintes de rang violées est donc faible. Cela nous permet d'avoir une initialisation plus proche du point optimal qu'une matrice tirée aléatoirement.

V.2.2.b Heuristique de sélection des images pivots

Dans notre algorithme, nous sélectionnons les images pivots à son initialisation et nous les conservons jusqu'à la fin. L'image pivot p_q de chaque requête $q \in \mathcal{Q}$ est sélectionnée dans l'ensemble des images dissimilaires $\mathcal{D}_q$ à la requête.

Pour sélectionner les images pivots, nous proposons une heuristique qui vise à minimiser le nombre de contraintes violées à l'initialisation de l'algorithme. Pour chaque requête $q \in \mathcal{Q}$, nous choisissons donc comme image pivot p_q , l'image qui a le plus fort score de similarité avec la requête parmi les images dissimilaire $\mathcal{D}_q$ à la requête :

$$p_q^* = \arg \max_{p_q \in \mathcal{D}_q} \mathbf{x}_q \mathbf{P}^{[0]\top} \mathbf{P}^{[0]} \mathbf{x}_{p_q}. \quad (\text{V.31})$$

V.2.2.c Heuristique de sélection des requêtes

Pour accélérer la vitesse de convergence de l'algorithme, nous proposons d'utiliser une méthode d'apprentissage actif basée sur une heuristique de sélection des échantillons utilisés pour calculer le gradient [Tong and Koller, 2002].

À chaque itération, nous sélectionnons un sous ensemble de requêtes $\tilde{\mathcal{Q}} \subset \mathcal{Q}$ de P requêtes. Pour sélectionner le sous ensemble de requêtes $\tilde{\mathcal{Q}}^{[t]}$, nous combinons deux approches : une sélection des requêtes qui ont le plus de contraintes violées et une sélection aléatoire. L'ensemble $\tilde{\mathcal{Q}}^{[t]}$ des P requêtes est alors constitué des $\lfloor P/2 \rfloor$ requêtes qui ont les plus fortes influences dans la fonction objectif $(\{f_q(\mathbf{P})\}_{q \in \mathcal{Q}})$ et de $\lceil P/2 \rceil$ requêtes choisies aléatoirement.

Cette combinaison d'une sélection active et d'une sélection arbitraire est une méthode simple pour éviter les problèmes liés à une sélection active pure. En effet, avec une sélection

objectif pure la descente de gradient peut être bloquée avant la convergence. Par exemple à l'itération t , nous obtenons avec la sélection active pure un ensemble de requête $\tilde{\mathcal{Q}}^{[t]}$ tel que le pas d'apprentissage $\mu^{[t]*}$ est nul (la direction du gradient n'optimise pas le problème global) alors $\mathbf{P}^{[t+1]} = \mathbf{P}_t$ et à l'itération $t + 1$, nous obtenons $\tilde{\mathcal{Q}}^{[t+1]} = \tilde{\mathcal{Q}}^{[t]}$. La descente de gradient est alors bloquée.

V.2.2.d Gradient

Pour chaque requête $q \in \tilde{\mathcal{Q}}$, nous calculons le gradient de la fonction objectif (V.27) :

$$\begin{aligned} \frac{\partial f_q(\mathbf{P})}{\partial \mathbf{P}} = & \alpha_q \sum_{i \in \mathcal{P}_q} h(\varepsilon + \Delta_{\mathbf{S}\mathbf{P}}(q, i, p_q))(\varepsilon + \Delta_{\mathbf{S}\mathbf{P}}(q, i, p_q)) \mathbf{A}_{qi} \mathbf{P} \\ & + \beta_q \sum_{j \in \mathcal{N}_q} h(\Delta_{\mathbf{S}\mathbf{P}}(q, p_q, j)) \Delta_{\mathbf{S}\mathbf{P}}(q, p_q, j) \mathbf{B}_{qj} \mathbf{P}, \end{aligned}$$

avec $\mathbf{A}_{qi} = \mathbf{x}_q \Delta \mathbf{x}_{p_q i}^\top + \Delta \mathbf{x}_{p_q i} \mathbf{x}_q^\top$ et $\mathbf{B}_{qj} = \mathbf{x}_q \Delta \mathbf{x}_{j p_q}^\top + \Delta \mathbf{x}_{j p_q} \mathbf{x}_q^\top$. Puis nous mettons à jour la matrice $\mathbf{P}$:

$$\mathbf{P}^{[t+1]} = \mathbf{P}_t + \mu^{[t]} \sum_{q \in \tilde{\mathcal{Q}}^{[t]}} \frac{\partial f_q(\mathbf{P}^{[t]})}{\partial \mathbf{P}} \quad (\text{V.32})$$

avec $\mu^{[t]}$ le pas d'apprentissage.

V.2.2.e Calcul du pas optimal

Pour obtenir le pas d'apprentissage optimal, nous proposons de résoudre le problème suivant :

$$\mu^{[t]*} = \arg \min_{\mu^{[t]}} \sum_{q \in \mathcal{Q}} f_q \left(\mathbf{P}_t + \mu^{[t]} \sum_{q \in \tilde{\mathcal{Q}}^{[t]}} \frac{\partial f_q(\mathbf{P}^{[t]})}{\partial \mathbf{P}} \right) \quad (\text{V.33})$$

Nous résolvons ce problème avec la méthode du nombre d'or [Avriel and Wilde, 1966, Gerald and Wheatley, 1994].

V.2.3 Introduction d'une contrainte de parcimonie

Comme nous l'avons déjà évoqué dans la section IV.2.2, l'utilisation de projecteurs pleins pour la réduction de la dimensionnalité des signatures entraîne des coûts mémoires et calculatoires prohibitifs. L'introduction d'une contrainte de parcimonie sur les projecteurs permet de fortement réduire ces coûts.

Nous proposons alors de contraindre les colonnes $\mathbf{p}_i$ de la matrice $\mathbf{P}$ avec la norme ℓ_0 . Nous pouvons réécrire le problème d'optimisation global (V.30) comme suit :

$$\begin{aligned} \mathbf{P}^* = & \arg \min_{\mathbf{P}} \sum_{q \in \mathcal{Q}} f_q(\mathbf{P}) \\ \text{s.t. } & \|\mathbf{p}_i\|_0 = M, \forall i; \end{aligned} \quad (\text{V.34})$$

avec $\|\cdot\|_0$ la norme ℓ_0 et M le nombre de valeurs non-nulles dans chaque colonne de la matrice $\mathbf{P}$.

Pour résoudre ce problème, nous proposons d'ajouter à notre algorithme de descente de gradient (Algorithme 1) une étape de projection sur la contrainte de parcimonie. La

projection sur la contrainte est effectuée à chaque itération après la mise à jour des projecteurs. Pour projeter chaque projecteur $\mathbf{p}_i$ sur la contrainte, nous résolvons le problème suivant :

$$\begin{aligned} \bar{\mathbf{p}}_i^* &= \arg \min_{\bar{\mathbf{p}}_i} \|\mathbf{u}_i - \bar{\mathbf{p}}_i\|_2^2 \\ \text{s.t. } &\|\bar{\mathbf{p}}_i\|_0 = M \end{aligned} \quad (\text{V.35})$$

Ce problème est résolu par seuillage des valeurs les plus proches de zero du vecteur :

$$\hat{p}_{ki} = p_{ki} \, h(|p_{ki}| - \nabla_i), \forall k \quad (\text{V.36})$$

avec $\nabla_i \in \mathbb{R}^+$ le seuil sélectionné pour respecter la contrainte de parcimonie $\|\bar{\mathbf{p}}_i\|_0 = M$.

Algorithm 2 Apprentissage de similarité pour la reconstruction de rang avec la contrainte ℓ_0 .

```

 $t \leftarrow 0$ .
Initialisation de la matrice  $\mathbf{P}^{[0]}$  avec les projecteurs proposés section IV.2.
Choix des images pivots  $p_{q \in \mathcal{Q}}$ .
Projection de  $\mathbf{P}^{[0]}$  sur la contrainte  $\|\mathbf{p}_i\|_0 = M$ .
répéter
  Sélection du sous ensemble  $\tilde{\mathcal{Q}}^{[t]} \subset \mathcal{Q}$  de  $P$  requêtes.
  Calcul du gradient sur l'ensemble de requête  $\tilde{\mathcal{Q}}^{[t]}$ .
  Calcul du pas optimale  $\mu^{[t]}$ .
  Mise à jour de la matrice des projecteurs  $\mathbf{P}^{[t+1]}$ .
  Projection de  $\mathbf{P}^{[t+1]}$  sur la contrainte  $\|\mathbf{p}_i\|_0 = M$ .
   $t \leftarrow t + 1$ .
jusqu'à Fonction objectif  $< 10^{-12}$  ou  $t > \text{maxIter}$ 

```

L'algorithme 2 montre les étapes de descente de gradient avec la projection sur la contrainte pour obtenir des projecteurs creux.

V.3 Expériences

Dans cette section, nous présentons notre protocole expérimental et l'évaluation de la méthode proposée.

V.3.1 Recherche par similarité

Nous évaluons en première les performances obtenues par nos méthodes en recherche de quasi-copie.

V.3.1.a Base d'évaluation et protocole expérimental

Pour évaluer notre méthode, nous utilisons la base d'images Holidays (Annexe A.3.1) [Jégou et al., 2008].

Puisque que nous avons besoin d'un ensemble d'apprentissage pour apprendre nos projecteurs, nous divisons l'ensemble des images de la base Holidays en deux ensembles distincts : un ensemble d'entraînement et un ensemble de test. Les deux ensembles sont composés de 250 groupes d'images similaires tirées au hasard. Les deux ensembles sont

indépendants. Nous utilisons le même protocole d'évaluation que pour la base Holidays pour évaluer à la fois le sous-ensemble d'entraînement et le sous-ensemble de test.

V.3.1.b Signatures

Pour toutes nos expériences, nous effectuons deux étapes de pré-traitement sur les images : un redimensionnement des images (largeur maximum d'une image de 512 pixels) ; une égalisation de l'histogramme des images. Nous utilisons le descripteur local HOG, avec une extraction dense tous les 3 pixels sur 4 échelles. Nous utilisons ces descripteurs pour calculer une signature VLAD et une signature FV avec des dictionnaires visuels de 64, 128, 256 and 512 mots visuels. Les dictionnaires visuels sont entraînés en utilisant uniquement les images de l'ensemble d'entraînement que nous avons défini. Nous calculons la signature VLAD avec une ACP dans chaque partie du modèle (comme nous l'avons proposé pour la signature VLAT dans la partie II.2.3.e) dans lequel nous préservons 80 dimensions par partie. Pour la signature FV, nous effectuons une ACP sur les descripteurs locaux dans lequel nous préservons 80 dimensions. En dernier, nous effectuons une normalisation power-normalization ($\alpha = 0.1$ pour toutes les expériences) et une normalisation ℓ_2 des signatures.

Pour nous comparer aux méthodes d'apprentissage non-supervisé, nous utilisons les projecteurs proposés Section IV.2. Nous entraînons les projecteurs sur l'ensemble d'entraînement et les performances obtenues sur l'ensemble de test sont reportées dans le Tableau V.1. Les lignes représentent les différentes signatures (par exemple, VLAD-64 est une signature VLAD avec un dictionnaire visuel de 64 mots visuels). La deuxième colonne est la dimension de la signature originale, la troisième colonne est le mAP obtenu avec la signature originale et les colonnes quatre à sept sont le mAP obtenu avec les signatures réduites pour différentes dimensions N de la signature finale. Nous pouvons voir que la signature FV obtient de meilleures performances que la signature VLAD. Le meilleur résultat avec une signature non réduite est obtenu avec FV-256 (88.1% de mAP). Pour les signatures réduites, les meilleurs résultats sont toujours obtenus avec les plus grande dimensions de la signature finale ($N = 256$). Cela montre que pour obtenir de bonnes performances en recherche de quasi-copie, il est préférable de calculer les signatures de plus grande dimension possible.

Nous utilisons également les projecteurs creux proposés à la Section IV.2.2.a. Nous obtenons ces projecteurs creux en calculant les approximations parcimonieuses des projecteurs denses que nous avons utilisé pour calculer les performances du Tableau V.1. Les performances obtenues sur l'ensemble de test sont reportées dans le Tableau V.2. Les lignes sont les différentes signatures. De la deuxième à la cinquième colonne sont reportés les mAP avec un taux de parcimonie de $\tau = 90\%$ pour différentes valeurs de N , et pour la sixième à la neuvième colonne sont reportés les mAP pour un taux de parcimonie de $\tau = 99\%$ pour différentes valeurs de N . Nous constatons que, plus la signature d'origine est de grande dimension, plus la perte de performance causée par l'approximation parcimonieuse est faible.

V.3.1.c Analyse de convergence

Dans un premier temps, nous analysons la convergence de l'algorithme que nous avons proposé avec et sans contrainte de parcimonie. Pour toutes les expériences sur la

Sign.	Dim.	Full	Réduction avec la méthode section IV.2			
			32	64	128	256
VLAD-64	5k	86.23	76.90	80.87	83.18	84.25
VLAD-128	10k	87.59	76.86	81.52	83.23	84.82
VLAD-256	20k	87.52	75.36	80.80	83.16	84.59
VLAD-512	41k	87.75	72.96	79.39	82.19	83.99
FV-64	10k	85.17	78.78	81.64	84.41	85.00
FV-128	20k	86.15	78.34	82.26	84.05	85.41
FV-256	41k	88.10	78.39	82.57	84.84	86.38
FV-512	82k	87.09	77.76	81.62	83.85	85.82

TABLE V.1 – Performances des signatures VLAD et FV non réduites et réduites avec la méthode proposé dans la section IV.2

Sign.	Réduit avec 90% de parcimonie				Réduit avec 99% de parcimonie			
	32	64	128	256	32	64	128	256
VLAD-64	74.64	79.98	82.23	82.85	67.26	74.16	76.76	78.23
VLAD-128	74.89	79.88	81.65	83.17	68.11	74.91	77.67	80.05
VLAD-256	72.71	78.98	81.33	82.89	62.23	71.63	74.78	77.78
VLAD-512	69.54	76.44	80.05	81.96	57.23	67.90	73.26	76.51
FV-64	77.50	80.88	82.75	84.46	64.67	74.56	79.42	81.36
FV-128	76.75	80.77	82.7	84.22	65.19	73.61	78.60	80.83
FV-256	75.82	81.01	83.39	85.49	64.71	74.82	79.08	81.98
FV-512	75.03	80.51	82.37	84.13	53.26	68.44	73.51	84.13

TABLE V.2 – Performances des signatures réduites sur l'ensemble de teste avec différent taux de parcimonie avec les projecteurs non-supervisés (Section IV.2.2.a) (mAP en %).

convergence, nous utilisons la signature FV avec une dictionnaire visuel de 512 mots visuels.

La Figure V.1(a) montre l'évolution de la fonction objectif sans la contrainte de parcimonie pour différentes valeurs de N . Nous voyons que l'algorithme que nous proposons minimise bien la fonction objectif. De plus, le taux de convergence semble être exponentiel. Nous notons que l'algorithme que nous proposons nécessite plus d'itérations pour converger avec les petites valeurs de N . Cependant, même pour les très petites valeurs de N , notre algorithme converge, ce qui signifie que toutes les contraintes sont respectées.

La Figure V.1(b) montre l'évolution du mAP sur l'ensemble d'entraînement sans contrainte de parcimonie et pour différentes valeurs de N . Nous voyons que le mAP augmente bien jusqu'à 100% (tri parfait des images pour toutes les requêtes), ce qui est cohérent avec la convergence vers "0" de la fonction objectif.

La Figure V.2(a) montre l'évolution de la fonction objectif avec la contrainte de parcimonie pour différents taux de parcimonie $\tau(\mathbf{P})$. Nous voyons que l'algorithme que nous proposons permet toujours de minimiser la fonction objectif, avec toutefois une vitesse de convergence plus lente. Nous notons que plus le taux de parcimonie est fort, plus la convergence est lente.

La Figure V.2(b) montre l'évolution du mAP sur l'ensemble d'entraînement pour différents taux de parcimonie $\tau(\mathbf{P})$. Encore, nous voyons que notre algorithme est capable

Sign.	Réduit avec 90% de parcimonie				Réduit avec 99% de parcimonie			
	32	64	128	256	32	64	128	256
VLAD-64	75.87	80.17	82.21	82.97	70.36	75.67	77.67	79.37
VLAD-128	75.95	80.32	82.36	82.96	72.03	77.32	79.78	81.08
VLAD-256	74.43	78.74	80.99	82.78	71.05	76.33	78.36	80.53
VLAD-512	71.97	76.99	79.70	81.02	71.61	75.16	78.10	79.45
FV-64	78.39	80.90	82.95	84.36	74.17	78.46	81.51	82.72
FV-128	77.99	80.94	82.78	84.31	74.22	78.62	80.71	82.47
FV-256	76.24	81.19	83.28	84.31	72.73	78.13	81.50	83.21
FV-512	76.04	79.95	81.59	83.12	75.24	78.38	80.16	80.90

TABLE V.3 – Performances des signatures réduites sur l'ensemble de teste avec différent taux de parcimonie (mAP en %).

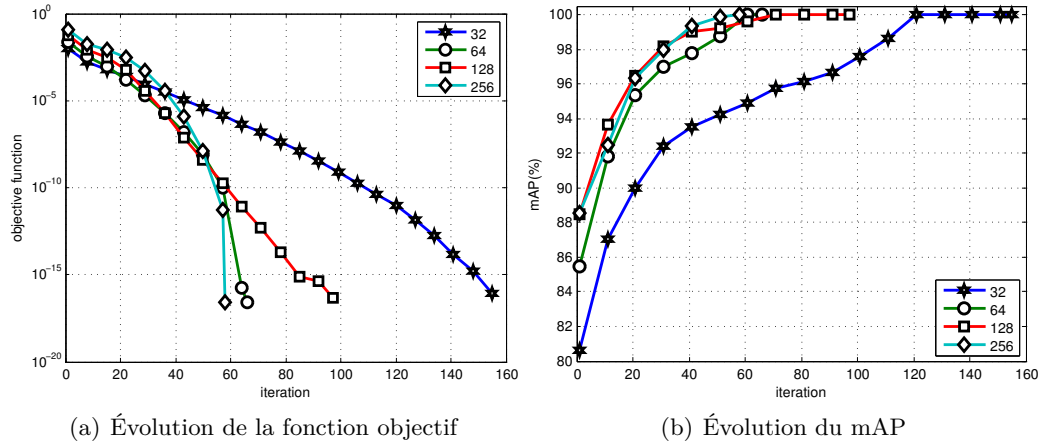


FIGURE V.1 – Évolution de la fonction objectif et du mAP sur l'ensemble d'entraînement sans la contrainte de parcimonie.

d'augmenter le mAP jusqu'à 100%, à condition que la contrainte de parcimonie ne fasse pas chuter la vitesse de convergence au delà d'un temps raisonnable.

V.3.1.d Performances en recherche par similarité

Maintenant, nous nous intéressons à la capacité de généralisation de notre méthode. Nous entraînons les projecteurs sur l'ensemble d'entraînement et en évaluant ces performances (mAP) sur l'ensemble de test.

La Figure V.3 montre l'évolution du mAP sur l'ensemble d'entraînement et de test avec un taux de parcimonie de $\tau = 99\%$ pour différentes valeurs de N . Nous voyons que notre algorithme augmente le mAP sur l'ensemble de test. Cette augmentation est encore plus grande lorsque le nombre de projecteurs est petit (Par exemple, 22% augmentation de mAP pour $R = 32$). Notons que le mAP est rapidement stabilisé sur l'ensemble de test, même si l'algorithme n'a pas totalement convergé.

Le Tableau V.3 montre le mAP obtenu sur l'ensemble de test avec les signatures réduites. Les lignes sont les différentes signatures. De la deuxième à la cinquième colonne sont reportés les mAP avec un taux de parcimonie de $\tau = 90\%$ pour différentes valeurs

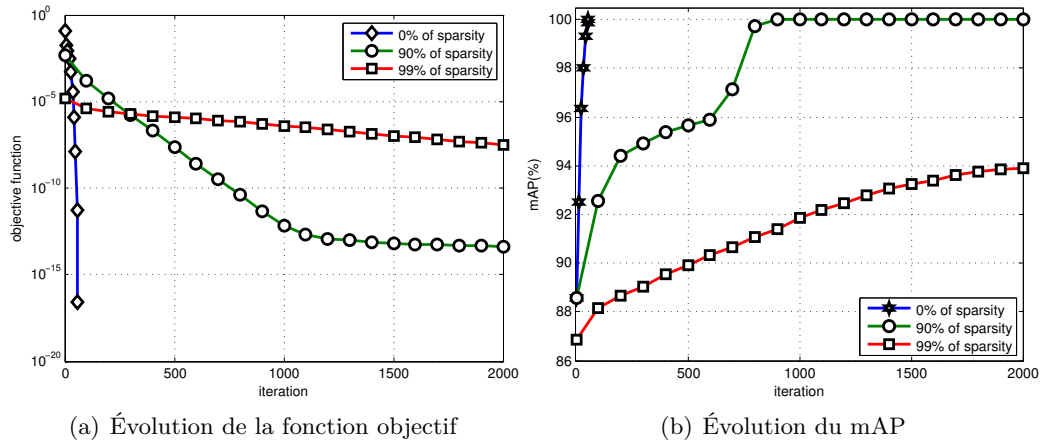


FIGURE V.2 – Évolution de la fonction objectif et du mAP sur l'ensemble d'entraînement avec la contrainte de parcimonie et plusieurs taux de parcimonie.

de N , et pour la sixième à la neuvième colonnes sont reportés les mAP pour un taux de parcimonie de $\tau = 99\%$ pour différentes valeurs de N . Nous pouvons voir que le mAP augmente avec N .

En comparant avec le Tableau V.2, nous constatons que, dans les cas où le nombre de paramètres d'apprentissage est petit (faible nombre de projecteurs, faible dimension de la signature d'origine ou fort taux de parcimonie), l'apprentissage permet d'obtenir de forts gains de performance. Nous voyons que quand le nombre de paramètres d'apprentissage devient trop grand, nous avons une perte de performance des signatures. Le problème n'est plus suffisamment contraint, et nous sur-apprenons la base d'entraînement.

V.4 Conclusions

Nous avons présenté dans ce chapitre une nouvelle méthode d'apprentissage supervisé pour obtenir des projecteurs linéaires pour la réduction de dimension des signatures dans le cadre de recherche de quasi-copies. Notre méthode est basée sur l'apprentissage de fonctions de similarité dans le but de respecter les contraintes de rang induites par la vérité terrain d'une base d'images en recherche de quasi-copie. Notre méthode a trois points importants : l'utilisation d'images pivots permettant de réduire considérablement le nombre de contraintes à prendre en compte et donc permettre l'utilisation de grandes bases d'apprentissage ; l'utilisation d'un ensemble de méthodes pour accélérer la convergence de l'algorithme d'apprentissage ; l'introduction d'une contrainte de parcimonie pour obtenir des projecteurs creux.

Les résultats expérimentaux ont montré que l'algorithme proposé permet bien d'apprendre à respecter l'ensemble des contraintes de rang et ainsi obtenir les meilleures performances possibles en recherche de quasi-copie sur la base d'entraînement. Nous avons constaté que l'introduction de la contrainte de parcimonie diminue fortement la vitesse de convergence de l'algorithme mais que les projecteurs obtenus permettent toujours d'améliorer les performances sur la base d'entraînement. Nous avons également constaté que les projecteurs obtenus permettent également d'améliorer les performances sur une base de test.

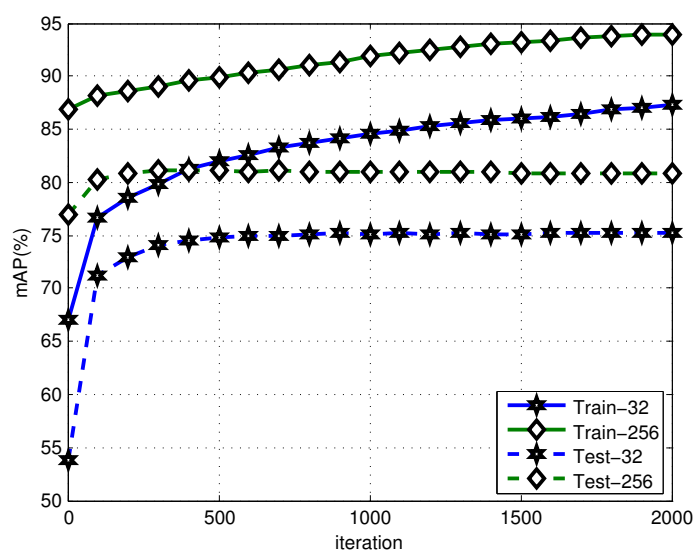


FIGURE V.3 – Évolution du mAP sur l'ensemble d'entraînement et l'ensemble de teste avec la contrainte de parcimonie.

Conclusions générales

Chapitre **VI**

Dans cette thèse, nous nous sommes intéressés aux méthodes d'indexation automatique par le contenu pour aider à l'indexation de très grands volumes de documents. Nous nous sommes particulièrement intéressés aux méthodes basées sur l'utilisation de signatures d'image. Nous avons deux objectifs : améliorer le plus possible le pouvoir de discrimination des signatures ; réduire le plus possible les coûts induit par leur extractions et leur utilisations.

VI.1 Principales contributions

Les travaux dans cette thèse sont concentrés sur trois problèmes clefs : la performance des signatures, les coûts d'utilisation des signatures (majoritairement dus à la dimension des signatures) et les coûts d'extractions.

VI.1.1 Amélioration des performances de la signature

Nous avons amélioré la signature VLAT basée sur l'approximation du noyau sur sac gaussien. Pour cela, nous avons montré que la signature VLAT peut être vue comme une signature à déviation de modèle. Nous avons explicité le modèle utilisé dans la signature VLAT ainsi que la fonction de mesure de déviation. Nous avons proposé une amélioration de ce modèle basée sur l'utilisation de l'ACP dans les différentes parties de l'espace des descripteurs. Cette amélioration permet de réduire la dimension de la signature tout en conservant ses performances.

VI.1.2 Réduction de la dimension des signatures

Pour réduire le coût de stockage des signatures ainsi que le coût calculatoire lors de leur utilisation, nous avons proposé deux nouvelles méthodes pour fortement réduire leur dimension. Nos méthodes sont basées sur la projection linéaire des signatures dans un espace de petite dimension.

La première méthode est basée sur un apprentissage non supervisé des projecteurs. Les projecteurs sont construits de telle sorte que la similarité entre les signatures d'origine soit conservée dans l'espace réduit.

La deuxième méthode est basée sur un apprentissage supervisé de fonction de similarité pour obtenir les projecteurs. Le but de cette méthode est de construire des projecteurs tel que les similarités entre les signatures dans l'espace réduit respectent un ensemble de contraintes sur le rang des images pertinentes. Ces contraintes de rang sont obtenues à l'aide d'une vérité terrain de recherche de quasi-copie.

VI.1.3 Réduction des coûts de redimensionnement

La réduction de la dimension des signatures est l'étape la plus coûteuse dans la chaîne d'extraction des signatures, aussi bien en terme de coût de stockage qu'en terme de coût calculatoire. Ces coûts sont dus à la grande dimension des projecteurs de réduction de dimension. Pour réduire ces coûts, nous avons proposé d'utiliser des projecteurs creux. Nous avons alors proposé deux méthodes pour obtenir ces projecteurs.

La première méthode est une extension de la méthode pour obtenir des projecteurs pleins de manière non supervisée. Nous construisons une approximation des projecteurs plein par seuillage des petites valeurs. Puis nous construisons une matrice de correction permettant de corriger les erreurs introduites par l'approximation des projecteurs. La projection dans l'espace réduit s'effectue alors en deux étapes : une projection peu coûteuse des signatures de grande dimension dans un espace de petite dimension à l'aide de projecteurs creux ; puis une projection peu coûteuse sur la matrice de correction dans l'espace de petite dimension.

La deuxième méthode est une extension de la méthode pour obtenir les projecteurs de manière supervisée. Nous avons introduit dans le problème d'optimisation une contrainte de parcimonie sur les projecteurs. Pour résoudre ce nouveau problème, nous avons ajouté à notre algorithme de résolution une étape de projection sur la contrainte de parcimonie.

VI.2 Perspectives

Dans ce manuscrit, nous nous sommes focalisés sur la chaîne d'extraction de signatures d'images basées sur l'agrégation de descripteurs locaux. Nous avons proposé plusieurs améliorations de cette chaîne dans le but d'améliorer ses performances ainsi que de réduire ses coûts. Plusieurs perspectives s'offrent à nous pour poursuivre les idées proposées dans ce document.

VI.2.1 Perspectives à court terme

Nous avons montré que l'utilisation de projecteurs creux permet de réduire fortement les coûts entraînés par l'étape de réduction de dimension des signatures tout en conservant de très bonnes performances par rapport aux signatures non réduites. Nous devons évaluer l'influence de l'ajout des contraintes de parcimonie sur les performances obtenues avec des bases de grande dimension.

Une évaluation plus approfondie de nos méthodes doit être effectuée en classification automatique sur des bases de grande dimension comme les bases du challenge ImageNET. Il serait très intéressant d'évaluer l'influence des projecteurs obtenus avec la méthode de

reconstruction du rang en recherche de quasi-copie sur des problèmes de classification automatique.

VI.2.2 Perspectives à moyen terme

Dans la chaîne d'extraction des signatures, de nombreux paramètres sont encore choisis empiriquement en fonction des performances obtenues par la signature. Il serait intéressant de construire des critères simples permettant d'apprendre ces paramètres.

Le centrage des données suivie d'une normalisation ℓ_2 des données sont deux étapes couramment utilisées dans les chaînes d'extraction de signature. La normalisation des données permet d'obtenir une propriété très intéressante quand on utilise un produit scalaire comme fonction de similarité : la similarité entre deux données identiques est toujours supérieure à la similarité entre deux données différentes. En effet, la normalisation des données correspond à la projection des données sur l'hypersphère de rayon 1 et l'utilisation du produit scalaire permet alors de mesurer le cosinus de l'angle entre les deux points sur l'hypersphère et son centre. Le centrage des données permet alors de placer le centre de l'hypersphère au centre des données pour obtenir une bonne répartition des données sur l'hypersphère et ainsi augmenter la dynamique des scores de similarité. Nous pouvons alors nous poser la question suivante : Le centrage par la moyenne permet-il d'obtenir la meilleure répartition des données sur l'hypersphère ou la meilleure dynamique des scores de similarité ? Il serait alors intéressant de trouver le centrage qui distribue le plus uniformément les données sur l'hypersphère ou qui donne la plus forte variance des scores de similarité.

La normalisation en loi de puissance est aussi une étape couramment utilisée dans les chaînes d'extraction de signature. Cette étape consiste à mettre à la puissance α la valeur absolue de chaque composante des données puis de reporter le signe de la composante. Dans la plus part des cas, le choix de α est fait de manière empirique ou par une méthode de validation croisée. Nous pouvons alors nous poser les questions suivantes : Peut-on construire un critère sur les données pour choisir le coefficient de la normalisation ? Obtient-on un gain de performance si nous utilisons un coefficient différent pour chaque composante de la donnée ? Nous proposons alors de trouver les coefficients de normalisation qui maximisent la variance des scores de similarité.

Il serait également intéressant d'étendre notre méthode de calcul des projecteurs de réduction dimension supervisée pour construire une matrice de correction des signatures réduites par les projecteurs creux obtenus par seuillage.

VI.2.3 Perspectives à long terme

Dans la chaîne d'extraction que nous avons présenté, nous effectuons un apprentissage successif des étapes d'extraction en partant des images jusqu'aux signatures. Le modèle alors obtenu dans la chaîne d'extraction a une forte attache aux données. Les récents travaux d'extraction de signature d'image basés sur les architectures *deep learning* effectuent leur apprentissage en deux étapes. La première étape consiste à effectuer un apprentissage successif des étapes d'extraction en partant des images jusqu'aux signatures. Puis la deuxième étape consiste à effectuer un apprentissage successif des étapes en partant d'une vérité terrain vers les images. Cela permet de modifier le modèle de la chaîne d'extraction pour qu'il modélise au mieux les informations utiles à la tâche d'indexation. Les signatures permettent alors d'obtenir de très bonnes performances en indexation par le contenu. Il

serait alors très intéressant d'ajouter une étape d'apprentissage à la chaîne d'extraction que nous avons proposé, pour obtenir un modèle qui modélise en priorité les informations utiles pour les tâches d'indexations.

Enfin, pour construire les signatures d'image, nous avons utilisé une extraction dense des descripteurs locaux de l'images. En effet, les expériences ont montré que l'extraction dense des descripteurs locaux permet d'obtenir des signatures plus discriminante que les méthodes basées sur une sélection des descripteurs locaux avec un à priori (point d'intérêt). Il serait alors très intéressant de construire une méthode permettant d'effectuer une sélection à posteriori des descripteurs locaux afin de supprimer les descripteurs qui n'apportent pas d'information à la signature.

Appendices



Les bases d'images

Dans cette section, nous présentons les différentes bases d'images que nous utilisons pour entraîner et évaluer nos méthodes. Les bases que nous utilisons sont des bases généralistes couramment utilisées pour évaluer les performances des méthodes de recherche par similarité ou des méthodes de classification automatique. Les bases d'images dites généralistes sont constituées de photographies de scènes urbaines et naturelles principalement prises par des amateurs. Les images des bases généralistes sont généralement soumises à aucune contrainte, aussi bien sur la qualité de l'image (résolution, compression, illumination, ...), que sur la qualité de représentation des concepts sémantiques (position inconnue, échelle inconnue, multiples concepts sémantiques représentés dans la même image, ...).

A.1 Bases d'images non annotées

A.1.1 Flickr60k

La base d'image Flickr60k est constituée de 67714 images extraites du site Web de partage de photographies Flickr¹ [Jégou et al., 2008]. C'est une base généraliste qui offre un échantillonnage représentatif des images partagées sur les sites de partage de photos. Elle est couramment utilisée comme base d'entraînement pour les étapes non-supervisées. Par exemple, elle est utilisée pour apprendre les modèles (par exemple pour le partitionnement ou mélanges de Gaussiennes) ou encore pour apprendre les paramètres des méthodes non-supervisées pour la réduction de dimension.

A.1.2 Flickr1M

La base d'image Flickr1M est constituée d'un million d'images extraites du site Web de partage de photographies Flickr [Jégou et al., 2008]. Flickr1M et Flickr60k sont deux ensembles d'images disjoints. Elle est couramment utilisée pour créer des distracteurs employés pour tester la robustesse des méthodes de recherche par similarité.

1. <https://www.flickr.com/>

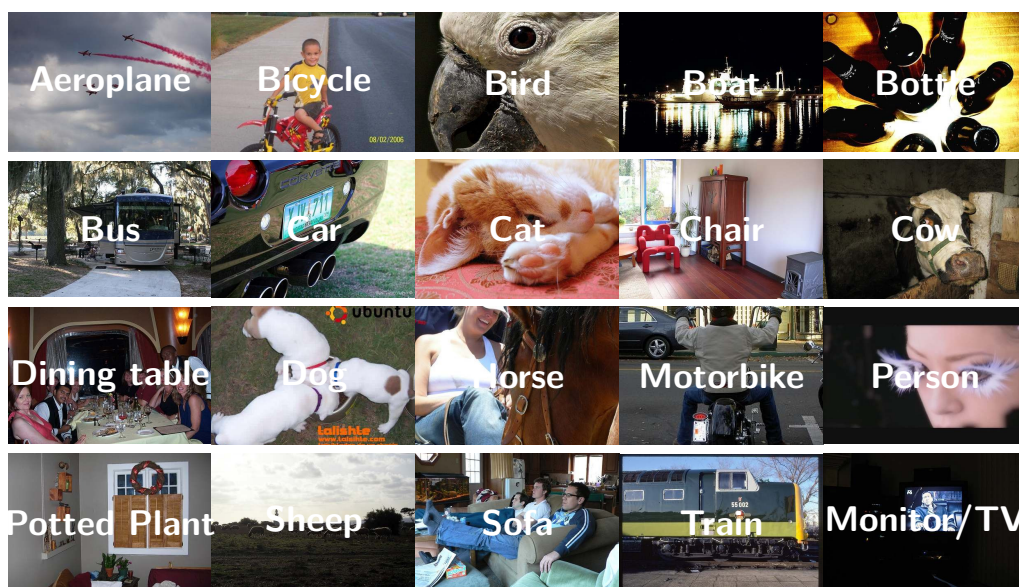


FIGURE A.1 – Exemple d'image de la base VOC 2007

A.2 Bases de classification

A.2.1 Base VOC2007

La base d'images VOC2007 est une base généraliste de 9963 images. Cette base a été utilisée pour le challenge "VOC" du réseau d'excellence PASCAL en 2007² [Everingham et al.,]. Ce challenge consiste à reconnaître automatiquement des objets dans un environnement réel. L'apprentissage des classifieurs statistiques est effectué sur un sous-ensemble d'entraînement. Les 20 classes (Figure A.1) d'objets sont : personne, oiseaux, chat, vache, chien, cheval, mouton, avion, vélo, bateau, bus, voiture, moto, train, bouteille, chaise, table à manger, plante en pot, sofa, TV / moniteur. Parmi les 9963 images, 24640 objets sont annotés. La base est divisée en deux parties : l'ensemble d'entraînement (50% des images) permettant d'entraîner les classifieurs ; et l'ensemble de test permettant de tester les classifieurs.

Pour évaluer la performance d'une méthode de classification, nous utilisons la mesure proposée par le challenge VOC en 2007. La mesure proposée est basée sur la courbe de précision/rappel du classifieur pour chaque classe. Pour chaque classe, la précision moyenne ou *Average Precision* (AP) est calculée. Le calcul est effectué sur la courbe de précision/rappel approximée à 11 points. Pour évaluer la performance d'une méthode de classification sur l'ensemble des catégories de la base, nous utilisons la moyenne sur l'ensemble des catégories des précisions moyennes ou *mean Average Precision* (mAP).

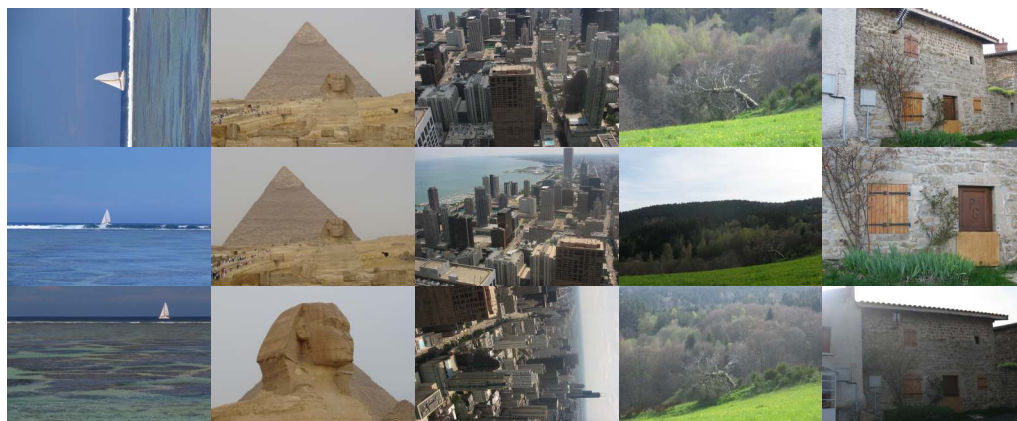


FIGURE A.2 – Exemple d'image de la base Holidays

A.3 Bases de copies et quasi-copies d'images

A.3.1 Base Holidays

La base d'images Holidays est une base généraliste de 1491 images proposée dans le contexte du projets ANR RAFFUT³ [Jégou et al., 2008]. Elle permet d'évaluer les méthodes de recherche d'images par similarité (Figure A.2). Les images sont regroupées en 500 catégories représentant la même scène prise plusieurs fois (au moins 2 fois) avec différentes orientations, points de vue, éclairage, flou, etc. Le challenge consiste à retrouver toutes les quasi-copies d'une image requête. Pour chaque requête, l'ensemble des images est trié par ordre décroissant de similarité visuelle par rapport à la requête.

Pour évaluer la performance d'une méthode de recherche par similarité, nous utilisons le protocole de la base. Nous calculons pour chaque requête, la précision moyenne de la fonction de similarité. Ce calcul ne prend pas en compte la requête dans la courbe de précision/rappel. De plus, seuls les points correspondants aux quasi-copies sont utilisés pour calculer la précision moyenne (AP), aucune interpolation n'est effectuée entre ces points. Enfin, nous calculons la moyenne des précisions moyennes sur l'ensemble des requêtes ou *mean Average Precision* (mAP).

A.3.2 Base Oxford

La base d'image Oxford est une base généraliste de 5062 images proposées par le *Visual Geometry Group* de l'Université d'Oxford⁴ [Philbin et al., 2007]. Elle est constituée d'images de monuments de la ville d'Oxford extraites du site Web de partage de photographies Flickr. Elle permet d'évaluer les méthodes de recherche d'image par similarité (Figure A.3). Les images sont regroupées en 11 catégories représentant plusieurs monuments de la ville d'Oxford. Pour chaque catégorie, 5 images requêtes sont définies, chacune représentant une vue différente du monument. Pour chaque requête, le but est de retrouver les images représentant la même vue du monument que la requête.

Pour évaluer la performance d'une méthode de recherche par similarité, nous utilisons

2. <http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/>

3. <https://lear.inrialpes.fr/~jegou/data.php>

4. <http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/>



FIGURE A.3 – Exemple d'image de la base Oxford



FIGURE A.4 – Exemple d'image de la base Paris

le protocole de la base. Nous calculons pour chaque requête la précision moyenne de la fonction de similarité. Pour chaque image requête, un ensemble d'images indésirables est défini, ces images ne sont pas prises en compte lors du calcul de la précision moyenne. Enfin nous calculons la moyenne sur l'ensemble des requêtes des précisions moyennes ou *mean Average Precision* (mAP).

A.3.3 Base Paris

La base d'images Paris est une base généraliste de 6412 images proposées par le *Visual Geometry Group* de l'Université d'Oxford⁵ [Philbin et al., 2008]. Elle est constituée d'images de monuments de la ville Paris extraite du site Web de partage de photographies Flickr. Elle permet d'évaluer les méthodes de recherche d'images par similarité (Figure A.4). Les images sont regroupées en 12 catégories représentant plusieurs monuments de la ville de Paris. Pour chaque catégorie, un ensemble d'images requêtes sont définies, chacune représentant une vue différente du monument. Pour chaque requête, le but est de retrouver les images représentant la même vue du monument que l'image requête.

Pour évaluer la performance d'une méthode de recherche par similarité, nous utilisons

5. <http://www.robots.ox.ac.uk/vgg/data/parisbuildings/>

le protocole de la base qui est identique au protocole de la base d'images Oxford.

Les mesures de performance

Annexe B

Pour comparer les différentes méthodes d'indexation, il existe de nombreux indicateurs permettant de mesurer et comparer leurs performances.

Pour évaluer les performances d'une méthode, il est nécessaire d'avoir une vérité terrain. Cette vérité terrain est la réponse idéale que l'on attend de la méthode d'indexation automatique. Par exemple, en recherche de quasi-copies, la vérité terrain est constituée d'un ensemble de requêtes et pour chaque requête de l'ensemble des documents qui sont des quasi-copies du document requête. Autre exemple, en classification automatique, la vérité terrain est constituée d'un ensemble de classes. Pour chaque classe nous avons deux ensembles de documents : un ensemble de documents appartenant à la classe et un ensemble de documents n'appartenant pas à la classe.

Les méthodes d'évaluation utilisent alors la vérité terrain pour évaluer les erreurs commises par la méthode d'indexation et pour leur attribuer un score de performance. Il est important de noter que le score de performance obtenu par une méthode d'indexation dépend directement de la vérité terrain utilisée. Il est impossible de comparer les scores obtenus par deux méthodes d'indexation si ces scores n'ont pas été calculés avec la même vérité terrain.

Dans cette thèse, nous utilisons le mAP (*mean Average Precision*) comme score de performance. Le mAP est compris entre 0% et 100%. Quand le mAP est égal à 100%, la méthode d'indexation n'a commis aucune erreur. Pour calculer le mAP, la méthode d'indexation doit être capable de répondre à la question suivante : parmi l'ensemble de documents $\mathcal{D}$, quel est l'ensemble de i documents $\mathcal{R}_i \in \mathcal{D}$ qui correspond le mieux à ce que l'on cherche ? Par exemple, quels sont les i documents qui sont le plus similaires au document requête ?

Le mAP est basé sur deux critères de mesure : le rappel et la précision. Le rappel R_i correspond à la proportion de documents corrects dans $\mathcal{R}_i$ par rapport au nombre total de documents corrects dans $\mathcal{D}$:

$$R_{qi} = \frac{|\mathcal{R}_i \cap \mathcal{P}_q|}{|\mathcal{P}_q|}, \quad (\text{B.1})$$

$i = \mathcal{R}_i $	1	2	3	4	5	6	7	...
$ \mathcal{R}_i \cup \mathcal{P}_q $	1	2	2	2	3	4	4	...
P_{qi}	1	1	2/3	2/4	3/5	4/6	4/7	...
R_{qi}	1/10	2/10	2/10	2/10	3/10	4/10	4/10	...
$R_{qi} - R_{q(i-1)}$	1/10	1/10	0	0	1/10	1/10	0	...

TABLE B.1 – Exemple de calcul du rappel et de la précision avec $|\mathcal{P}_q| = 10$.

Base	VOC2007	Holidays	Oxford	Paris
mAP	77.2%	84.3%	68.0%	79.5%

TABLE B.2 – Meilleurs performance obtenu sur les bases de l'état de l'art [Sharif Razavian et al., 2014].

avec $|\cdot|$ le cardinal de l'ensemble et $\mathcal{P}_q$ l'ensemble des documents corrects de la requête q . La précision P_i correspond à la proportion de documents correctes dans $\mathcal{R}_i$ par rapport au nombre de documents dans $\mathcal{R}_i$:

$$P_{qi} = \frac{|\mathcal{R}_i \cap \mathcal{P}_q|}{|\mathcal{R}_i|}. \quad (\text{B.2})$$

Pour chaque requête q , ces deux critère de mesure sont utilisés pour calculer la précision moyenne que l'on note par : AP *Average Precision*. Nous calculons la précision moyenne comme suit :

$$AP_q = \sum_{k=1}^{\mathcal{D}} P_{qk} (R_{qk} - R_{q(k-1)}). \quad (\text{B.3})$$

Le Tableau B.1 montre un exemple d'évolution du rappel et de la précision. Le calcul de la précision moyenne est obtenu en sommant les valeurs en gras du tableau et en divisant le total par $|\mathcal{P}_q|$, le nombre de documents corrects dans $\mathcal{D}$.

Le mAP est alors obtenu en calculant la moyenne des précisions moyennes pour chaque requête de la vérité terrain :

$$mAP = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} AP_q. \quad (\text{B.4})$$

Pour donner une idée des ordres de grandeur du mAP obtenus sur des vérités terrain, nous pouvons nous intéresser au mAP obtenu dans le cas d'une méthode qui répond aléatoirement. Pour la base d'images Holidays utilisée pour évaluer les méthodes de recherche de quasi-copies le mAP alors obtenu est de 0.3%. Pour la base d'images VOC2007 utilisée pour évaluer les méthodes de classification automatique avec une méthode qui répond aléatoirement, nous obtenons un mAP de 7%.

Pour donnée une idée des ordres de grandeurs du mAP obtenus, nous reportons dans la Tableau B.2 les meilleurs performance obtenu à ce jour. Nous voyons que sur base Holidays le meilleur mAP est de l'ordre de 85%, ce qui est une très bonne performance en comparaison au mAP obtenu par une méthode aléatoire.

Matrices et vecteurs creux

Les matrices creuses et vecteurs creux sont majoritairement constitués des valeurs nulles. L'utilisation de ce type de matrices/vecteurs est très intéressante pour réduire leur coûts de stockage et les coûts calculatoires des opérations matricielles/vectorielles. La parcimonie d'une matrice ou d'un vecteur est définie par le taux de parcimonie :

$$\tau(\mathbf{P}) = \frac{\text{Nombre de zéros dans } \mathbf{P}}{\text{Nombre de valeurs dans } \mathbf{P}}. \quad (\text{C.1})$$

Pour l'utilisation et le stockage des matrices creuses, nous considérons toute matrice creuse comme une concaténation de vecteurs colonnes creux :

$$\mathbf{M} = [\mathbf{m}_1 \cdots \mathbf{m}_m] \quad (\text{C.2})$$

Nous utilisons cette convention car l'opération matricielle que nous effectuons le plus souvent est la projection d'un ensemble de vecteurs sur un autre $\mathbf{C} = \mathbf{A}^\top \mathbf{B}$. Pour calculer chaque composante c_{ij} de la matrice $\mathbf{C}$, nous accédons à un seul vecteur à la fois dans chacune des matrices $\mathbf{A}$ et $\mathbf{B}$.

C.1 Stockage des vecteurs creux

Pour stocker un vecteur creux, nous considérons la méthode de stockage suivante : pour chaque valeur non-nulle du vecteur, nous stockons un couple $\{i_j, v_j\}$ correspondant à l'indice i_j de la $j^{\text{ème}}$ valeur non-nulle dans le vecteur et v_j correspondant à la valeur. Nous stockons en mémoire des couples (index, valeur) les uns après les autres :

$$[\{i_1, v_1\}, \{i_2, v_2\}, \cdots, \{i_{P-1}, v_{P-1}\}, \{i_P, v_P\}] \quad (\text{C.3})$$

Nous considérons que le coût de stockage d'un indice est égal au coût de stockage d'une valeur (par exemple, l'indice est stocké sur un entier (4 octets) et la valeur sur un flottant (4 octets)).

Avec cette convention, le stockage d'un vecteur avec un taux de parcimonie de 0% (vecteur dense) a un coût deux fois supérieur à la méthode de stockage standard (toutes les valeurs les une à la suite des autres).

Le ratio du coût de stockage de cette convention et du coût de stockage usuel est $2(1 - \tau(\mathbf{P}))$. Donc, le taux de parcimonie doit être supérieur à 50% pour obtenir un gain de stockage.

Pour des vecteurs creux possédant une structure particulière dans la distribution des valeurs non-nulles, des méthodes de stockage plus performantes peuvent être envisagées. Par exemple, pour les vecteurs creux par bloc, un stockage des valeurs avec un codage par plages (*Run-length encoding*) permet de fortement réduire le coût de stockage en comparaison avec la méthode de stockage classique.

C.2 Opérations avec des vecteurs creux

L'opération que nous utilisons majoritairement avec les vecteurs creux est le produit scalaire. Le coût calculatoire du produit scalaire est fortement réduit avec des vecteurs creux car il n'est pas nécessaire d'effectuer les opérations sur les valeurs nulles. Par exemple, le produit scalaire entre un vecteur dense $\mathbf{x}$ et un vecteur creux $\tilde{\mathbf{y}}$ s'écrit

$$s(\mathbf{x}, \tilde{\mathbf{y}}) = \sum_{\{i_i, v_i\} \in \tilde{\mathbf{y}}} v_i \mathbf{x}_{i_i} \quad (\text{C.4})$$

Le ratio entre le coût calculatoire avec un vecteur parcimonieux et le coût calculatoire avec un vecteur dense est de $1 - \tau(\mathbf{P})$ avec $\tau(\mathbf{P})$ le taux de parcimonie de $\mathbf{P}$.

Bibliographie

- [Ahonen et al., 2006] Ahonen, T., Hadid, A., and Pietikainen, M. (2006). Face description with local binary patterns : Application to face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(12) :2037–2041.
- [Arandjelovic and Zisserman, 2012] Arandjelovic, R. and Zisserman, A. (2012). Three things everyone should know to improve object retrieval. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2911–2918. IEEE.
- [Avila et al., 2011] Avila, S., Thome, N., Cord, M., Valle, E., and de A Araujo, A. (2011). Bossa : Extended bow formalism for image classification. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 2909–2912. IEEE.
- [Avriel and Wilde, 1966] Avriel, M. and Wilde, D. J. (1966). Optimality proof for the symmetric Fibonacci search technique. *Fibonacci Quarterly*, 4(3) :265–269.
- [Baeza-Yates et al., 1999] Baeza-Yates, R., Ribeiro-Neto, B., et al. (1999). *Modern information retrieval*, volume 463. ACM press New York.
- [Bai et al., 2009] Bai, B., Weston, J., Grangier, D., Collobert, R., Sadamasa, K., Qi, Y., Chapelle, O., and Weinberger, K. (2009). Supervised semantic indexing. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 187–196. ACM.
- [Bay et al., 2006] Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf : Speeded up robust features. In *Computer Vision—ECCV 2006*, pages 404–417. Springer.
- [Bilmes et al., 1998] Bilmes, J. A. et al. (1998). A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *International Computer Science Institute*, 4(510) :126.
- [Bishop et al., 1995] Bishop, C. M. et al. (1995). *Neural networks for pattern recognition*. Clarendon press Oxford.
- [Bottou, 2004] Bottou, L. (2004). Stochastic learning. In *Advanced lectures on machine learning*, pages 146–168. Springer.
- [Chakravarti and Meng, 2009] Chakravarti, R. and Meng, X. (2009). A study of color histogram based image retrieval. In *ITNG*, pages 1323–1328.
- [Chatfield et al., 2011] Chatfield, K., Lempitsky, V., Vedaldi, A., and Zisserman, A. (2011). The devil is in the details : an evaluation of recent feature encoding methods. In *Proceedings of the British Machine Vision Conference (BMVC)*.
- [Chechik et al., 2010] Chechik, G., Sharma, V., Shalit, U., and Bengio, S. (2010). Large scale online learning of image similarity through ranking. *The Journal of Machine Learning Research*, 11 :1109–1135.
- [Cover and Thomas, 2012] Cover, T. M. and Thomas, J. A. (2012). *Elements of information theory*. John Wiley & Sons.
- [Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE.

-
- [Davis et al., 2007] Davis, J. V., Kulis, B., Jain, P., Sra, S., and Dhillon, I. S. (2007). Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216. ACM.
- [Delhumeau et al., 2013] Delhumeau, J., Gosselin, P.-H., Jégou, H., and Pérez, P. (2013). Revisiting the vlad image representation. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 653–656. ACM.
- [do Valle Jr, 2008] do Valle Jr, E. A. (2008). *Local-descriptor matching for image identification systems*. PhD thesis, Université de Cergy-Pontoise.
- [Douze et al., 2011] Douze, M., Ramisa, A., and Schmid, C. (2011). Combining attributes and fisher vectors for efficient image retrieval. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 745–752. IEEE.
- [Dudoit et al., 2002] Dudoit, S., Fridlyand, J., and Speed, T. P. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American statistical association*, 97(457) :77–87.
- [Everingham et al.,] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html>.
- [Everingham et al., 2006] Everingham, M., Zisserman, A., Williams, C., and Van Gool, L. (2006). The pascal visual object classes challenge 2006 (voc 2006) results.
- [Gerald and Wheatley, 1994] Gerald, C. F. and Wheatley, P. O. (1994). *Applied numerical analysis*. Addison-Wesley.
- [Gordoa et al., 2012] Gordoa, A., Rodríguez-Serrano, J. A., Perronnin, F., and Valveny, E. (2012). Leveraging category-level labels for instance-level image retrieval. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3045–3052. IEEE.
- [Gorisse, 2010] Gorisse, D. (2010). *Passage à l’échelle des méthodes de recherche sémantique dans les grandes bases d’images*. PhD thesis, Université de Cergy Pontoise.
- [Gorisse et al., 2008] Gorisse, D., Cord, M., Precioso, F., and Philipp-Foliguet, S. (2008). Fast approximate kernel-based similarity search for image retrieval task. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE.
- [Guillaumin et al., 2009] Guillaumin, M., Verbeek, J., and Schmid, C. (2009). Is that you ? metric learning approaches for face identification. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 498–505. IEEE.
- [Han and Ma, 2002] Han, J. and Ma, K.-K. (2002). Fuzzy color histogram and its use in color image retrieval. *Image Processing, IEEE Transactions on*, 11(8) :944–952.
- [Harris and Stephens, 1988] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK.
- [Hays and Efros, 2008] Hays, J. and Efros, A. A. (2008). Im2gps : estimating geographic information from a single image. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE.
- [Horn, 1990] Horn, R. A. (1990). The hadamard product. In *Proc. Symp. Appl. Math*, volume 40, pages 87–169.
- [Horn and Johnson, 2012] Horn, R. A. and Johnson, C. R. (2012). *Matrix analysis*. Cambridge university press.

- [Hotelling, 1933] Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6) :417.
- [Hotelling, 1936] Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, pages 321–377.
- [Hyvärinen et al., 2004] Hyvärinen, A., Karhunen, J., and Oja, E. (2004). *Independent component analysis*, volume 46. John Wiley & Sons.
- [Itti et al., 1998] Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11) :1254–1259.
- [Jain, 2010] Jain, A. K. (2010). Data clustering : 50 years beyond k-means. *Pattern Recognition Letters*, 31(8) :651–666.
- [Jain et al., 2012] Jain, M., Benmokhtar, R., Jégou, H., and Gros, P. (2012). Hamming embedding similarity-based image classification. In *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*, page 19. ACM.
- [Jégou et al., 2008] Jégou, H., Douze, M., and Schmid, C. (2008). Hamming embedding and weak geometric consistency for large scale image search. In *Computer Vision–ECCV 2008*, pages 304–317. Springer.
- [Jégou et al., 2011] Jégou, H., Douze, M., and Schmid, C. (2011). Product quantization for nearest neighbor search. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(1) :117–128.
- [Jégou et al., 2010] Jégou, H., Douze, M., Schmid, C., and Pérez, P. (2010). Aggregating local descriptors into a compact image representation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3304–3311. IEEE.
- [Jégou et al., 2012] Jégou, H., Perronnin, F., Douze, M., Sánchez, J., Pérez, P., and Schmid, C. (2012). Aggregating local image descriptors into compact codes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(9) :1704–1716.
- [Jolliffe, 2005] Jolliffe, I. (2005). *Principal component analysis*. Wiley Online Library.
- [Ke and Sukthankar, 2004] Ke, Y. and Sukthankar, R. (2004). Pca-sift : A more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–506. IEEE.
- [Kedem et al., 2012] Kedem, D., Tyree, S., Sha, F., Lanckriet, G. R., and Weinberger, K. Q. (2012). Non-linear metric learning. In *Advances in Neural Information Processing Systems*, pages 2573–2581.
- [Kihl et al., 2013] Kihl, O., Picard, D., and Gosselin, P.-H. (2013). A unified formalism for video descriptors. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 2416–2419. IEEE.
- [Kitchen and Rosenfeld, 1982] Kitchen, L. and Rosenfeld, A. (1982). Gray-level corner detection. *Pattern recognition letters*, 1(2) :95–102.
- [Koch and Ullman, 1987] Koch, C. and Ullman, S. (1987). Shifts in selective visual attention : towards the underlying neural circuitry. In *Matters of Intelligence*, pages 115–141. Springer.

-
- [Law et al., 2013] Law, M. T., Thome, N., and Cord, M. (2013). Quadruplet-wise image similarity learning. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 249–256. IEEE.
- [Law et al., 2014] Law, M. T., Thome, N., and Cord, M. (2014). Fantope regularization in metric learning.
- [Li and Wang, 2003] Li, J. and Wang, J. Z. (2003). Automatic linguistic indexing of pictures by a statistical modeling approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(9) :1075–1088.
- [Lindeberg, 1998] Lindeberg, T. (1998). Feature detection with automatic scale selection. *International journal of computer vision*, 30(2) :79–116.
- [Liu et al., 2010] Liu, G.-H., Zhang, L., Hou, Y.-K., Li, Z.-Y., and Yang, J.-Y. (2010). Image retrieval based on multi-texton histogram. *Pattern Recognition*, 43(7) :2380–2389.
- [Lloyd, 1982] Lloyd, S. (1982). Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2) :129–137.
- [Lowe, 1999] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee.
- [Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2) :91–110.
- [Mahalanobis, 1936] Mahalanobis, P. C. (1936). On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, 2 :49–55.
- [Mao and Jain, 1992] Mao, J. and Jain, A. K. (1992). Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Pattern recognition*, 25(2) :173–188.
- [Martin et al., 2001] Martin, D., Fowlkes, C., Tal, D., and Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 416–423. IEEE.
- [Mignon and Jurie, 2012] Mignon, A. and Jurie, F. (2012). Pcca : A new approach for distance learning from sparse pairwise constraints. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2666–2672. IEEE.
- [Mikolajczyk and Schmid, 2001] Mikolajczyk, K. and Schmid, C. (2001). Indexing based on scale invariant interest points. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 525–531. IEEE.
- [Mikolajczyk and Schmid, 2004] Mikolajczyk, K. and Schmid, C. (2004). Scale & affine invariant interest point detectors. *International journal of computer vision*, 60(1) :63–86.
- [Mikolajczyk and Schmid, 2005] Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10) :1615–1630.
- [Mikolajczyk et al., 2005] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Van Gool, L. (2005). A comparison of affine region detectors. *International journal of computer vision*, 65(1-2) :43–72.

- [Moravec, 1980] Moravec, H. P. (1980). Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical report, DTIC Document.
- [Mouragnon et al., 2006] Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F., and Sayd, P. (2006). Real time localization and 3d reconstruction. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 363–370. IEEE.
- [Natarajan, 1995] Natarajan, B. K. (1995). Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2) :227–234.
- [Negrel et al., 2013] Negrel, R., Picard, D., and Gosselin, P. (2013). Web-scale image retrieval using compact tensor aggregation of visual descriptors. *MultiMedia, IEEE*, 20(3) :24–33.
- [Negrel et al., 2012a] Negrel, R., Picard, D., and Gosselin, P.-H. (2012a). Compact tensor based image representation for similarity search. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pages 2425–2428. IEEE.
- [Negrel et al., 2014a] Negrel, R., Picard, D., and Gosselin, P.-H. (2014a). Dimensionality reduction of visual features using sparse projectors for content-based image retrieval. In *Image Processing (ICIP), 2014 21th IEEE International Conference on*. IEEE.
- [Negrel et al., 2014b] Negrel, R., Picard, D., and Gosselin, P.-H. (2014b). Evaluation of second-order visual features for land-use classification. In *Content-Based Multimedia Indexing (CBMI), 2014 12th International Workshop on*, pages 1–5. IEEE.
- [Negrel et al., 2012b] Negrel, R., Picard, D., Gosselin, P.-H., et al. (2012b). Using spatial pyramids with compacted vlat for image categorization. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR), 2012, Tsukuba, Japan, November 11–15, 2012*, pages 2460–2463.
- [Negrel et al., 2014c] Negrel, R., Picard, D., Gosselin, P.-H., et al. (2014c). Efficient dimension reduction of global signature with sparse projectors for image near duplicate retrieval. In *Proceedings of the 22st International Conference on Pattern Recognition (ICPR), 2014, Stockholm, Sweden, August 24–28, 2014*.
- [Nistér et al., 2004] Nistér, D., Naroditsky, O., and Bergen, J. (2004). Visual odometry. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–652. IEEE.
- [Pearson, 1901] Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11) :559–572.
- [Peng et al., 2014] Peng, X., Wang, L., Qiao, Y., and Peng, Q. (2014). Boosting vlad with supervised dictionary learning and high-order statistics. In *Computer Vision–ECCV 2014*, pages 660–674. Springer.
- [Perronnin and Dance, 2007] Perronnin, F. and Dance, C. (2007). Fisher kernels on visual vocabularies for image categorization. In *Computer Vision and Pattern Recognition, 2007. CVPR’07. IEEE Conference on*, pages 1–8. IEEE.
- [Perronnin et al., 2010a] Perronnin, F., Liu, Y., Sánchez, J., and Poirier, H. (2010a). Large-scale image retrieval with compressed fisher vectors. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3384–3391. IEEE.

- [Perronnin et al., 2010b] Perronnin, F., Sánchez, J., and Mensink, T. (2010b). Improving the fisher kernel for large-scale image classification. In *Computer Vision—ECCV 2010*, pages 143–156. Springer.
- [Philbin et al., 2007] Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. In *Computer Vision and Pattern Recognition, 2007. CVPR’07. IEEE Conference on*, pages 1–8. IEEE.
- [Philbin et al., 2008] Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2008). Lost in quantization : Improving particular object retrieval in large scale image databases. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE.
- [Picard and Gosselin, 2011] Picard, D. and Gosselin, P.-H. (2011). Improving image similarity with vectors of locally aggregated tensors. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 669–672. IEEE.
- [Picard and Gosselin, 2013] Picard, D. and Gosselin, P.-H. (2013). Efficient image signatures and similarities using tensor products of local descriptors. *Computer Vision and Image Understanding*, 117(6) :680–687.
- [Precioso et al., 2011] Precioso, F., Cord, M., Gorisse, D., and Thome, N. (2011). Efficient bag-of-feature kernel representation for image similarity search. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 109–112. IEEE.
- [Safadi and Quénot, 2013] Safadi, B. and Quénot, G. (2013). Descriptor optimization for multimedia indexing and retrieval. In *Content-Based Multimedia Indexing (CBMI), 2013 11th International Workshop on*, pages 65–71. IEEE.
- [Saul et al., 2006] Saul, L. K., Weinberger, K. Q., Ham, J. H., Sha, F., and Lee, D. D. (2006). Spectral methods for dimensionality reduction. *Semisupervised learning*, pages 293–308.
- [Schölkopf et al., 1997] Schölkopf, B., Smola, A., and Müller, K.-R. (1997). Kernel principal component analysis. In *Artificial Neural Networks—ICANN’97*, pages 583–588. Springer.
- [Schölkopf and Smola, 2002] Schölkopf, B. and Smola, A. J. (2002). *Learning with kernels : support vector machines, regularization, optimization, and beyond*. MIT press.
- [Shalev-Shwartz et al., 2007] Shalev-Shwartz, S., Singer, Y., and Srebro, N. (2007). Pegasos : Primal estimated sub-gradient solver for svm. In *ICML, 2007 24th International Conference on*, pages 807–814. ACM.
- [Sharif Razavian et al., 2014] Sharif Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S. (2014). Cnn features off-the-shelf : an astounding baseline for recognition. *arXiv preprint arXiv :1403.6382*.
- [Shawe-Taylor and Cristianini, 2004] Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge university press.
- [Shi and Tomasi, 1994] Shi, J. and Tomasi, C. (1994). Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR’94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE.
- [Sivic and Zisserman, 2003] Sivic, J. and Zisserman, A. (2003). Video google : A text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470–1477. IEEE.

- [Tamura et al., 1978] Tamura, H., Mori, S., and Yamawaki, T. (1978). Textural features corresponding to visual perception. *Systems, Man and Cybernetics, IEEE Transactions on*, 8(6) :460–473.
- [Teng et al., 2005] Teng, L., Li, H., Fu, X., Chen, W., and Shen, I.-F. (2005). Dimension reduction of microarray data based on local tangent space alignment. In *Cognitive Informatics, 2005.(ICCI 2005). Fourth IEEE Conference on*, pages 154–159. IEEE.
- [Tong and Koller, 2002] Tong, S. and Koller, D. (2002). Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 2 :45–66.
- [Torgerson, 1958] Torgerson, W. S. (1958). Theory and methods of scaling.
- [Van De Sande et al., 2010] Van De Sande, K. E., Gevers, T., and Snoek, C. G. (2010). Evaluating color descriptors for object and scene recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9) :1582–1596.
- [Van De Weijer et al., 2006] Van De Weijer, J., Gevers, T., and Bagdanov, A. D. (2006). Boosting color saliency in image feature detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(1) :150–156.
- [Van der Maaten et al., 2009] Van der Maaten, L., Postma, E., and Van den Herik, H. (2009). Dimensionality reduction : A comparative review. *Technical Report TiCC TR 2009-005*.
- [van Gemert et al., 2008] van Gemert, J. C., Geusebroek, J.-M., Veenman, C. J., and Smeulders, A. W. (2008). Kernel codebooks for scene categorization. In *Computer Vision—ECCV 2008*, pages 696–709. Springer.
- [Van Gemert et al., 2010] Van Gemert, J. C., Snoek, C. G., Veenman, C. J., Smeulders, A. W., and Geusebroek, J.-M. (2010). Comparing compact codebooks for visual categorization. *Computer Vision and Image Understanding*, 114(4) :450–462.
- [van Gemert et al., 2010] van Gemert, J. C., Veenman, C. J., Smeulders, A. W., and Geusebroek, J.-M. (2010). Visual word ambiguity. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(7) :1271–1283.
- [Wang et al., 2010] Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., and Gong, Y. (2010). Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3360–3367. IEEE.
- [Wang et al., 2006] Wang, M., Hua, X.-S., Song, Y., Yuan, X., Li, S., and Zhang, H.-J. (2006). Automatic video annotation by semi-supervised learning with kernel density estimation. In *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 967–976. ACM.
- [Wang et al., 2009] Wang, X., Han, T. X., and Yan, S. (2009). An hog-lbp human detector with partial occlusion handling. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 32–39. IEEE.
- [Weinberger et al., 2005] Weinberger, K. Q., Blitzer, J., and Saul, L. K. (2005). Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480.
- [Weinberger and Saul, 2009] Weinberger, K. Q. and Saul, L. K. (2009). Distance metric learning for large margin nearest neighbor classification. *The Journal of Machine Learning Research*, 10 :207–244.

- [Won et al., 2002] Won, C. S., Park, D. K., and Park, S.-J. (2002). Efficient use of mpeg-7 edge histogram descriptor. *Etri Journal*, 24(1) :23–30.
- [Xing et al., 2002] Xing, E. P., Jordan, M. I., Russell, S., and Ng, A. Y. (2002). Distance metric learning with application to clustering with side-information. In *Advances in neural information processing systems*, pages 505–512.
- [Yang et al., 2009] Yang, J., Yu, K., Gong, Y., and Huang, T. (2009). Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1794–1801. IEEE.
- [Zhu et al., 2006] Zhu, Q., Yeh, M.-C., Cheng, K.-T., and Avidan, S. (2006). Fast human detection using a cascade of histograms of oriented gradients. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1491–1498. IEEE.

Représentations Optimales pour la Recherche dans les Bases d'Images Patrimoniales

Résumé : Depuis plusieurs décennies, le développement des technologies de numérisation et de stockage ont permis la mise en œuvre de nombreux projets de numérisation du patrimoine culturel. L'approvisionnement massif et continu de ces bases de données numériques du patrimoine culturel entraîne de nombreux problèmes d'indexation. En effet, il n'est plus possible d'effectuer une indexation manuelle de toutes les données. Pour indexer et rendre accessible facilement les données, des méthodes d'indexation automatique et d'aide à l'indexation se sont développées depuis plusieurs années. Cependant, les méthodes d'indexation automatique pour les documents non-textuels (image, vidéo, son, modèle 3D, ...) sont encore complexes à mettre en œuvre pour de grands volumes de données.

Dans cette thèse, nous nous intéressons en particulier à l'indexation automatique d'images. Pour effectuer des tâches d'indexation automatique ou d'aide à l'indexation, il est nécessaire de construire une méthode permettant d'évaluer la similarité entre deux images. Nos travaux sont basés sur les méthodes à signatures d'image ; ces méthodes consistent à résumer le contenu visuel de chaque image dans une signature (vecteur unique), puis d'utiliser ces signatures pour calculer la similarité entre deux images. Pour extraire les signatures, nous utilisons la chaîne d'extraction suivante : en premier, nous extrayons de l'image un grand nombre de descripteurs locaux ; puis nous résumons l'ensemble de ces descripteurs dans une signature de grande dimension ; enfin nous réduisons fortement la dimension de la signature. Les signatures de l'état de l'art basées sur cette chaîne d'extraction permettent d'obtenir de très bonnes performances en indexation automatique et en aide à l'indexation. Cependant, les méthodes de l'état de l'art ont généralement de forts coûts mémoires et calculatoires qui rendent impossible leur mise en œuvre sur des grands volumes de données.

Dans cette thèse, notre objectif est double : d'une part nous voulons améliorer les signatures d'images pour obtenir de très bonnes performances dans les problèmes d'indexation automatique ; d'autre part, nous voulons réduire les coûts de la chaîne de traitement, pour permettre le passage à l'échelle. Nous proposons des améliorations d'une signature d'image de l'état de l'art nommée VLAT (*Vectors of Locally Aggregated Tensors*). Ces améliorations permettent de rendre la signature plus discriminante tout en réduisant sa dimension.

Pour réduire la dimension des signatures, nous effectuons une projection linéaire de la signature dans un espace de petite dimension. Nous proposons deux méthodes pour obtenir des projecteurs de réduction de dimension tout en conservant les performances des signatures d'origine. Notre première méthode consiste à calculer les projecteurs qui permettent d'approximer le mieux possible les scores de similarités entre les signatures d'origine. La deuxième méthode est basée sur le problème de recherche de quasi-copies ; nous calculons les projecteurs qui permettent de respecter un ensemble de contraintes sur le rang des images dans la recherche par rapport à l'image requête. L'étape la plus coûteuse de la chaîne d'extraction est la réduction de dimension de la signature à cause de la grande dimension des projecteurs.

Pour les réduire, nous proposons d'utiliser des projecteurs creux en introduisant une contrainte de parcimonie dans nos méthodes de calcul des projecteurs. Comme il est généralement complexe de résoudre un problème d'optimisation avec une contrainte de parcimonie stricte, nous proposons pour chacun des problèmes une méthode pour obtenir une approximation des projecteurs creux recherchés.

L'ensemble de ces travaux font l'objet d'expériences montrant l'intérêt pratique des méthodes proposées par comparaison avec les méthodes de l'état de l'art.

Mots-clés : Patrimoine, Recherche par similarité, Classification d'image.

Optimal representation for searching the image databases heritage

Abstract : In the last decades, the development of scanning and storing technologies resulted in the development of many projects of cultural heritage digitization. The massive and continuous flow of numerical data in cultural heritage databases causes many problems for indexing. Indeed, it is no longer possible to perform a manual indexing of all data. To index and ease the access to data, many methods of automatic and semi-automatic indexing have been proposed in the last years. The current available methods for automatic indexing of non-textual documents (images, video, sound, 3D model, ...) are still too complex to implement for large volumes of data.

In this thesis, we focus on the automatic indexing of images. To perform automatic or semi-automatic indexing, it is necessary to build an automatic method for evaluating the similarity between two images. Our work is based on image signature methods ; these methods involve summarising the visual content of each image in a signature (single vector), and then using these signatures to compute the similarity between two images. To extract the signatures, we use the following pipeline : First, we extract a large number of local descriptors of the image ; Then we summarize all these descriptors in a large signature ; Finally, we strongly reduce the dimensionality of the resulting signature. The state of the art signatures based on this pipeline provide very good performance in automatic indexing. However, these methods generally incur high storage and computational costs that make their implementation impossible on large volumes of data.

In this thesis, our goal is twofold : First, we wish to improve the image signatures to achieve very good performance in automatic indexing problems ; Second, we want to reduce the cost of the processing chain to enable scalability. We propose to improve an image signature of the state of the art named VLAT (*Vectors of Locally Aggregated Tensors*). Our improvements increase the discriminative power of the signature.

To reduce the size of the signatures, we perform linear projections of the signatures in a lower dimensional space. We propose two methods to compute the projectors while maintaining the performance of the original signatures. Our first approach is to compute the projectors that best approximate the similarities between the original signatures. The second method is based on the retrieval of quasi-copies ; We compute the projectors that meet the constraints on the rank of retrieved images with respect to the query image. The most expensive step of the extraction pipeline is the dimensionality reduction step ; these costs are due to the large dimensionality of the projectors.

To reduce these costs, we propose to use sparse projectors by introducing a sparsity constraint in our methods. Since it is generally complex to solve an optimization problem with a strict sparsity constraint, we propose for each problem a method for approximating sparse projectors.

This thesis work is the subject of experiments showing the practical value of the proposed methods in comparison with existing methods

Keywords : Heritage, Similarity search, Image classification.