



MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE  
LA RECHERCHE SCIENTIFIQUE  
UNIVERSITÉ ABDELHAMID IBN BADIS - MOSTAGANEM

**Faculté des Sciences Exactes et de l'Informatique**  
**Département de Mathématiques et d'Informatique**  
**Filière : Informatique**

MEMOIRE DE FIN D'ETUDES  
Pour l'Obtention du Diplôme de Master en Informatique  
Option : **Ingénierie des Systèmes d'Information**

THEME :

**UTILISATION DU CONTEXTE POUR LA DETECTION  
DE MULTI-CONCEPTS DANS LES IMAGES FIXES**

**Etudiante :**

✓ **Lattar Hafsa**

**Encadrant :**

✓ **Dr.Hamadi Abdelkader**

**Année Universitaire 2016/2017**

## **Résumé**

Rechercher la cooccurrence d'un ensemble de concepts visuels (multi-concept) dans des images/vidéos non annotées est une étape importante pour répondre à des requêtes complexes des utilisateurs. En effet, ces requêtes sont souvent exprimées via un ensemble de termes sémantiques. D'autre part, considérer qu'un document multimédia est indexé par plusieurs concepts est utile : une simple combinaison d'un ensemble de concepts peut représenter d'autres sémantiques pouvant être complexes. Par exemple, la combinaison des concepts « neige », « montagne » et « personne(s) en train de se déplacer » pourrait être liée à une scène de « skieur » ou « une compétition de ski ». Cette problématique a été peu traitée dans l'état de l'art comparée à la détection de concepts singuliers dans les images. D'autre part, l'utilisation du contexte (les relations inter-concepts) a prouvé son efficacité dans l'indexation sémantique des documents multimédia.

Dans ce travail on a étudié, testé et évalué nos approches qui consistent à considérer le contexte pour la détection de multi-concepts dans les images fixes. Les expérimentations ont été évaluées sur le corpus standard international pascal voc pour la détection de bi-concepts and tri-concepts. Les approches testées ont donné lieu à des résultats intéressants comparables ou même meilleures que ceux de l'état de l'art.

**Mots-clés** : Indexation sémantique, multimédia, contexte, apprentissage supervisé, concepts multiples, bi-concepts, tri-concepts, Pascal Voc.

## **Abstract**

Searching for the co-occurrence of a set of visual concepts (multi-concepts) in non-annotated images/ videos is an important step to answering complex user queries. Indeed, these queries are often expressed via a set of semantic terms. On the other hand, considering that a multimedia document is indexed by several concepts is useful: a simple combination of a set of concepts can represent other semantics that that can be complex. For example, the combination of concepts “snow”, “mountain” and “person(s) moving” could be linked to a “skier” or “ski competition scene”. This problem has been little dealt in the state of the art compared to the detection of singular concepts in the images. On the other hand, the use of context (inter-concept relationships) has proved its effectiveness in the semantic indexing of multimedia documents.

In this work we have studied, tested and evaluated our approaches which consider the context for the detection of multi-concepts in the still images. The experiments were evaluated on the international standard corpus pascal voc for bi-concepts and tri-concepts. The tested approaches achieved very interesting results comparable or even better than those of the state of the art.

**Keywords**: Semantic indexing, multimedia, context, supervised learning, Multiple concepts, bi-concepts, tri-concepts, Pascal Voc.

## **Dédicace**

Je dédie ce mémoire

À mes chers parents ma mère et mon père

À ma sœur et mes frères

À mes amies et mes camarades

Sans oublier tous mes professeurs que se soit

Primaire, du moyen, du secondaire ou de l'enseignement  
supérieur.

## Remerciement

Je souhaite manifester mes sincères remerciements à Dieu, le tout puissant, pour ses faveurs et ses grâces de m' avoir donné le courage et la patience pour achever ce modeste travail

Je remercie naturellement ma famille pour son aide, sa générosité et son soutien moral qui ont été pour moi une source de courage et de confiance.

Je tiens à présenter de tout mon cœur mes remerciements et mes reconnaissances à mon honorable encadreur «Mr. *Hamadi Abdelkader* » pour son aide, ses conseils précieux, sa gentillesse, son encouragement, sa disponibilité dans ce projet et sa confiance qui m'a permis de ne jamais faiblir et de poursuivre toujours plus loin mon travail.

Je remercie aussi vivement les honorables membres du jury qui ont accepté d'évaluer ce travail.

Enfin, un grand merci à tous mes enseignants du début à la fin de mes études, et à tous mes amis.

## Sommaire

Liste des Tableaux.....	i
Table des Figures .....	i
Liste des Abréviations.....	ii
Introduction générale .....	1
Système de détection de concepts .....	1
1. Introduction.....	3
2. Système de détection de concepts dans les images .....	3
3. Description des images.....	4
3.1. Descripteurs de bas niveau .....	4
3.1.1. Descripteurs de couleurs .....	4
3.1.2. Descripteurs de texture.....	5
3.1.3. Descripteurs de la forme .....	5
3.1.4. Descripteurs de points d'intérêt.....	5
3.2. Descripteurs de haut niveau .....	6
3.3 Optimisation des descripteurs .....	6
4. Modélisation/prédiction .....	6
5. Fusion.....	7
6. Ré-ordonnancement .....	8
7. Les ontologies .....	8
8. Utilisation du contexte pour la détection de concepts.....	8
9. Protocole d'évaluation des systèmes de détection de concepts.....	8
9.1. Mesures d'évaluation .....	8
9.2. Compagnes d'évaluation.....	9
10. Conclusion.....	10
Système de détection de multi-concepts .....	9
1. Introduction.....	11
2. La tâche de détection de multi-concepts .....	11
3. Description des approches de détection de multi-concepts.....	12
4. Discussion .....	14
5. Conclusion.....	15
Utilisation du contexte pour la détection de concepts dans les images .....	14
1. Introduction.....	16
2. Qu'est ce que le contexte ?.....	16
3. Pourquoi avoir besoin du contexte ? .....	16
4. Quelques types de contexte .....	17
5. Comment utiliser le contexte dans un système de détection de concepts ?.....	17

6. Quelques approches retenues utilisant le contexte sémantique .....	19
7. Conclusion.....	22
Contributions et expérimentations .....	22
1. Introduction.....	23
2. Description des approches proposées.....	23
2.1 Approche 1 : Facteur de boosting.....	23
2.2. Approche 2 : Rétroaction conceptuelle .....	23
3. Expérimentation .....	24
3.1. Données utilisées.....	24
3.2. Méthode d'apprentissage utilisée.....	25
3.3. Détecteurs de concepts .....	25
3.4. Fusion.....	26
3.5. Détecteurs de multi-concepts .....	26
4. Processus d'expérimentation.....	26
4.1. Approche 1 : Facteur de boosting .....	26
4.2 Approche 2 : rétroaction conceptuelle.....	27
5. Résultats et discussions : .....	27
L'approche « facteur de boosting » : .....	27
L'approche rétroaction conceptuelle .....	30
6. Synthèse .....	32
7. Conclusion.....	32
Conception et développement du système .....	22
1. Introduction.....	33
2. Conception .....	33
2.1. Diagramme de cas d'utilisation.....	34
2.2. Diagramme de classes .....	34
3. Implémentation.....	35
3.1. Environnement matériel et logiciel .....	35
3.2. Les interfaces graphiques principales.....	36
4. Conclusion.....	40
Conclusion générale .....	41
Bibliographie.....	42

## Liste des Tableaux

<b>Tableau 4. 1</b> : La collection voc 2012. ....	24
<b>Tableau 4. 2</b> : le nombre de chaque type de concept. ....	24
<b>Tableau 4. 3</b> : le nombre de chaque type de catégorie. ....	25
<b>Tableau 4. 4</b> : Descripteurs utilisés. ....	25
<b>Tableau 4. 5</b> : Résultats (MAP (gain %) de notre approche et une comparaison avec les résultats du système de base pour les concepts singuliers. ....	27
<b>Tableau 4. 6</b> : Résultats (MAP (gain %) de notre approche et une comparaison avec les résultats du système de base pour les concepts doubles. ....	28
<b>Tableau 4. 7</b> : Résultats (MAP (gain %) de notre approche et une comparaison avec les résultats du système de base pour les concepts double. ....	28
<b>Tableau 4. 8</b> : Résultats (MAP (gain %) de notre approche et une comparaison avec les résultats du système de base pour les concepts triples. ....	29
<b>Tableau 4. 9</b> : Résultats (MAP (gain %) de notre approche et une comparaison avec les résultats du système de base pour les concepts triples. ....	29
<b>Tableau 4. 10</b> : Résultats (MAP (gain %) de notre approche et une comparaison avec les résultats du système de base pour les concepts singuliers. ....	30
<b>Tableau 4. 11</b> : Résultats (MAP (gain %) de notre approche et une comparaison avec les résultats du système de base pour les concepts paires. ....	30
<b>Tableau 4. 12</b> : Résultats (MAP (gain %) de notre approche et une comparaison avec les résultats du système de base pour les concepts paires. ....	31
<b>Tableau 4. 13</b> : Résultats (MAP (gain %) de notre approche et une comparaison avec les résultats du système de base pour les concepts triples. ....	31
<b>Tableau 4. 14</b> : Résultats (MAP (gain %) de notre approche et une comparaison avec les résultats du système de base pour les concepts triples. ....	32

## Table des Figures

<b>Figure 1.1</b> : Architecture d'un système de détection de concepts dans les images [1]. ....	3
<b>Figure 1.2</b> : Exemple de l'histogramme de couleur [8]. ....	5
<b>Figure 1.3</b> : Principe de calcul du descripteur SIFT [9]. ....	5
<b>Figure 1. 4</b> : Courbe de rappel-Précision. ....	9
<b>Figure 2.1:</b> Apprentissage direct d'un détecteur de multi-concept [3]. ....	13
<b>Figure 2.2</b> : Fusion de détecteurs de concepts individuels [3]. ....	14
<b>Figure 2.3</b> : Résultats (AP) de la détection de bi-concept (AP) par différentes approches sur la collection Pascal VOC 2012[1]. ....	14
<b>Figure 2.4</b> : Résultats (AP) de la détection de tri-concept (AP) par différentes approches sur la collection Pascal VOC 2012[1]. ....	15
<b>Figure 3.1</b> : Utilité du contexte pour la détection de concepts visuels [1]. ....	16
<b>Figure 3.2</b> : Différentes possibilités d'utilisation du contexte dans un système d'indexation de documents multimédia [1] ....	18
<b>Figure 3.3</b> : Exemple d'apprentissage basé sur l'ontologie [6]. ....	19
<b>Figure 3.4:</b> Sélection des ancêtres et descendants du concept cible dans la hiérarchie de concepts (ex. Car) [5].	21
<b>Figure 3.5:</b> Système d'indexation sémantique avec rétroaction conceptuelle[1]. ....	22
<b>Figure 4. 1</b> : l'approche facteur de boosting. ....	23
<b>Figure 4. 2</b> : L'approche rétroaction conceptuelle. ....	24
<b>Figure 5. 1</b> Interface principale de StarUML version 2.8.0. ....	33
<b>Figure 5. 2</b> Diagramme de cas d'utilisation ....	34
<b>Figure 5.3</b> Diagramme de classes. ....	35
<b>Figure 5. 4:</b> écran de démarrage. ....	36
<b>Figure 5. 5:</b> Interface principe. ....	37
<b>Figure 5. 6:</b> interface de détection de concepts "approche classique". ....	37
<b>Figure 5. 7:</b> interface de détection de concepts "approche du contexte". ....	38
<b>Figure 5. 8:</b> Interface d'évaluation des résultats. ....	38
<b>Figure 5. 9:</b> Interface de visualisation des résultats. ....	39

<b>Figure 5. 10 :</b> interface des paramètres d'apprentissage.....	39
<b>Figure 5. 11:</b> interfaces des paramètres de Facteur de boosting.....	40

## **Liste des Abréviations**

**VOC :** Visual Object Classes.

**SVM:** Support Vector Machine.

**CSS :** Curvative Scale Space descriptors.

**SIFT :** Scale Invariant Feature Transform.

**MSVM:** Multi-SVM.

**AP:** Average Precision.

**infAP :** Inferred Average Precision.



## Introduction générale

Grâce aux avancées récentes de la technologie ces dernières années, en particulier dans le domaine du multimédia et de l'informatique, la quantité d'information croît d'une manière exponentielle, spécialement les images. Alors, *Comment s'y retrouver dans ces montagnes d'images? Comment pouvoir rechercher simplement dans une banque d'images ?* Si pour nous les humains, le processus de comparaison d'images et donc, la recherche, semble facile, nous sommes limités lorsqu'il s'agit de traiter d'énormes quantités d'information. Par conséquent, le besoin d'automatiser cette tâche s'est imposé. Or, pour pouvoir rechercher automatiquement des images, il faut que ces dernières soient annotées par des sémantiques (des concepts). Il est clair qu'annoter des images manuellement s'avère très coûteux. Donc, la nécessité d'un système d'indexation automatique des bases d'images n'est donc plus à démontrer.

L'indexation est une pratique indispensable pour retrouver rapidement des documents. Jusqu'à une époque récente, elle semblait réservée à l'intelligence humaine. Car indexer ne consiste pas à créer des indexes, mais à affecter aux documents des indices, des marques significatives (concepts) à leurs contenus, à la suite d'une série d'opérations mentales complexes. Cette Indexation est appelée indexation sémantique. Cette dernière est généralement réalisée par la détection de concepts visuels via des approches d'apprentissage automatique.

Historiquement, le premier type de système proposé est le système d'indexation et de recherche de documents multimédia par le contenu, un tel système permet aux utilisateurs de formuler des requêtes en présentant un exemple du type de l'image recherchée. Le système identifie alors, parmi la collection d'images, celles qui correspondent le plus à l'image requête, et les affiche. Malgré leur utilité, ce genre de systèmes ne répond pas à tous les besoins des utilisateurs à cause de l'impossibilité d'exprimer tous les besoins via des exemples. Par exemple, avec une image d'un bâtiment, exprime-t-on que l'on recherche ce bâtiment ou un bâtiment similaire. De plus, quand on recherche une image, c'est généralement parce qu'on n'en possède pas des exemples. Nous rappelons que ces systèmes ignorent totalement la sémantique et fonctionnent de manière aveugle, en comparant des caractéristiques de bas niveau [1].

La manière la plus simple pour un utilisateur, surtout un novice ou débutant, est de formuler ses attentes à travers des termes textuels. Pour ce faire, un système automatique doit pouvoir faire une correspondance entre du texte compréhensible par l'humain et un contenu brute. Autrement dit, il devient nécessaire de passer à une analyse sémantique. Cette solution, simple pour l'utilisateur est très compliquée à réaliser. En effet, cela est dû au fait qu'elle ouvre la possibilité d'exprimer des requêtes sémantiques complexes, auxquelles il s'avère très difficile de répondre via les méthodes existantes dans l'état de l'art [1].

La recherche sémantique est une tâche très difficile à cause des problèmes du fossé sémantique [2], de données (disponibilité, crédibilité, description), de classes déséquilibrées, de la difficulté de décrire des images (Descripteurs pas toujours efficaces pour décrire toutes les images). Rajoutons à cela, les problèmes liés aux méthodes d'apprentissage.

Malgré la difficulté de la tâche de la recherche sémantique, la communauté a connu une amélioration des résultats ces dernières années. La plupart des efforts ont été focalisés sur la

détection de concepts singuliers dans les images mais indexer l'image par un concept singulier ne suffit pas pour répondre aux requêtes complexes des utilisateurs. Cette observation ouvre la possibilité de voir la problématique de détection d'un ensemble de concepts simultanément (multi-concepts) pour améliorer la performance des systèmes : Par exemple, la détection de « montagne, neige et skieur » à la fois. Si la tâche de détection de concepts singuliers est difficile, spécialement pour les concepts rares ou difficiles à décrire d'une manière efficace, cette difficulté est accentuée pour le cas des multi-concepts.

Le très peu de travaux [3] [4] qui ont abordé cette problématique dans l'état de l'art confirment la difficulté de la tâche avec les résultats médiocres obtenus.

Pour le cas des concepts singuliers, les chercheurs ont constaté que la détection d'un concept singulier seul n'est pas optimale car un même concept peut apparaître dans des contextes très divers et son apparence peut être très différente en fonction de ces contextes, pour cela, les chercheurs ont exploité le contexte pour améliorer les performances initiales de détection d'un concept cible et les résultats obtenus étaient bons [5] [6] [1].

En s'inspirant de ces résultats et en considérant un multi-concept comme un concept, cette même idée pourrait fonctionner aussi bien pour le cas des multi-concepts. Nous nous intéressons dans notre travail à l'utilisation du contexte pour améliorer la détection de multi-concepts dans les images fixes. Nous allons valider nos approches en utilisant les bi-concepts et les tri-concepts dans le contexte de la campagne d'évaluation Pascal Voc, en utilisant le corpus Pascal Voc 2012.

Le reste du rapport est divisé en cinq chapitres suivis d'une conclusion générale. Les chapitres 1 et 2 sont dédiés à la description de l'architecture d'un système de détection de concepts singuliers et de multi-concepts, respectivement. Dans le chapitre 3, nous présenterons la notion du contexte et quelques travaux exploitant le contexte sémantique pour améliorer la détection de concepts dans les images et les vidéos. Le chapitre 4 est dédié à la présentation de nos approches qui consistent à utiliser le contexte pour améliorer la détection des multi-concepts et à la description des expérimentations réalisées et des résultats d'évaluation obtenus.

# **Chapitre 1**

## **Systeme de detection de concepts**

## 1. Introduction

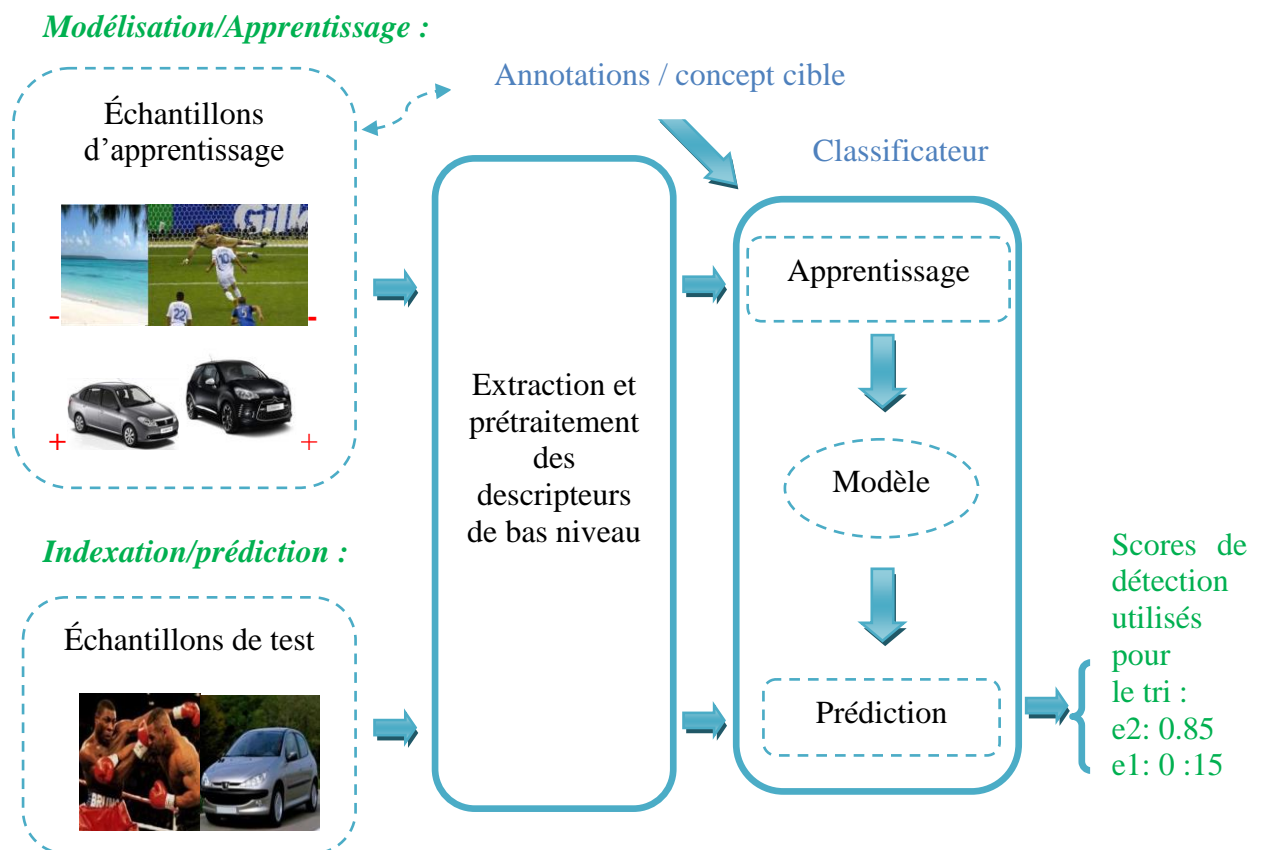
Dans ce chapitre, nous présentons une description d'un système de base de détection de sémantiques (concepts) dans les images. Nous détaillerons les différentes étapes par lesquelles passe un tel système. Nous allons considérer par la suite uniquement le cas des images, mais le système reste valable pour le cas des vidéos.

## 2. Système de détection de concepts dans les images

Un système de détection de concepts dans les images inclut deux étapes essentielles :

- ✓ La modélisation (ou l'apprentissage).
- ✓ La prédiction (ou l'indexation).

La figure 1.1 présente un schéma général décrivant un système basique de détection de concepts dans les images.



**Figure 1.1 :** Architecture d'un système de détection de concepts dans les images [1].

Dans l'étape de *modélisation*, le système cherche à étudier la corrélation entre les descriptions des exemples et leurs annotations dans le but de créer un modèle de classification. Le modèle est ensuite utilisé pour assigner des scores de classification à des exemples étant donnés leurs descripteurs de bas niveau. Ces scores peuvent être interprétés

comme des probabilités que les exemples en question contiennent le concept cible. Avant que le modèle soit appris, un ou plusieurs descripteurs sont extraits de chaque échantillon d'apprentissage. Une étape de prétraitement et d'optimisation des descripteurs peut être prévue. L'étape d'optimisation peut concerner la normalisation (ex. Transformation de loi de puissance [7] ) ou la réduction de dimensionnalité (ex. L'analyse en composantes principales) des vecteurs représentant les descripteurs. Cette étape peut influencer l'étape d'apprentissage et donc, la performance du système. Une fois les descripteurs prêts, on peut réaliser une fusion de ces descripteurs. On parle de fusion précoce (*voir la section 5*). L'apprentissage est donc lancé à ce stade en utilisant un algorithme d'apprentissage supervisé (ex. SVM, Réseaux de neurones, arbres de décision, etc.). L'algorithme d'apprentissage utilise les descripteurs ainsi que les annotations manuelles des échantillons d'apprentissage pour apprendre une fonction de classification.

Dans l'étape de *prédiction*, le modèle appris dans la phase d'apprentissage est utilisé pour prédire l'occurrence (ou non-occurrence) du concept cible dans des échantillons non vus (l'ensemble de test). Les mêmes descripteurs que ceux utilisés lors de l'étape d'apprentissage, sont extraits des échantillons de test et ces derniers subissent aussi les mêmes approches d'optimisation.

Ensuite, des scores de prédiction sont calculés. Ces scores reflètent la probabilité que les exemples de test contiennent le concept à détecter. Les échantillons de test sont ensuite ordonnés par ordre décroissant par rapport à leurs scores de prédiction. L'élément ayant plus de chance (ou de probabilité) de contenir le concept cible se retrouvera en haut de la liste. La liste ordonnée est ensuite retournée comme résultat du processus de détection de concept (ou d'indexation). Nous notons ici, que l'étape d'ordonnement ne sert que pour la visualisation ou le calcul d'une mesure d'évaluation (voir la section 9.1) qui nécessitent un ordonnancement des documents en fonction de leur susceptibilité de contenir le concept cible [1].

### 3. Description des images

Il existe dans l'état de l'art plusieurs descripteurs pour décrire une image. Ces descripteurs peuvent être catégorisés selon le niveau d'extraction. En l'occurrence :

- ✓ Les descripteurs globaux qui sont calculés à partir de l'image entière ;
- ✓ Les descripteurs locaux qui sont calculés sur une partie de l'image considérée ;

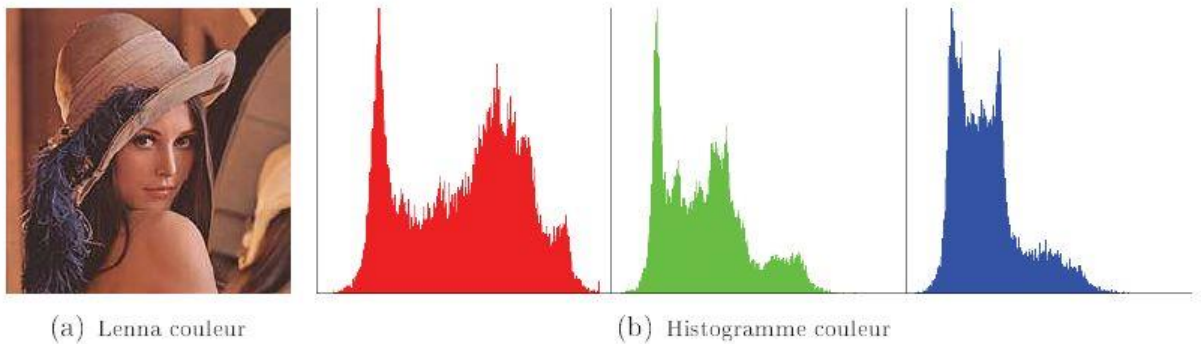
On peut également catégoriser les descripteurs selon plusieurs autres critères, notamment la modalité ou le niveau d'extraction, comme présenté ci-après.

#### 3.1. Descripteurs de bas niveau

##### 3.1.1. Descripteurs de couleurs

La couleur est une caractéristique riche en informations et très utilisée pour la représentation des images. Il existe plusieurs descripteurs de couleurs parmi eux : l'histogramme, le corrélogramme, le descripteur des couleurs dominantes, etc.

L'histogramme de couleurs qui est montré dans la figure 1.2 est parmi les descripteurs les plus simples. Sa performance est cependant, moins bonne comparée à certaines autres [8].



**Figure 1.2** : Exemple de l'histogramme de couleur [8].

### 3.1.2. Descripteurs de texture

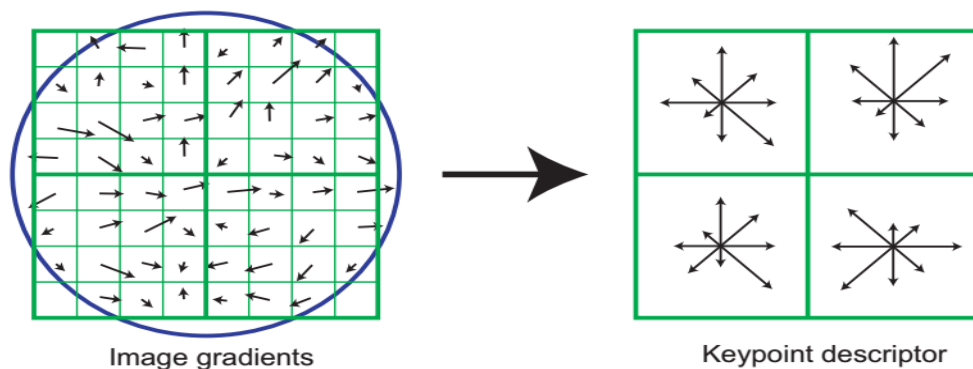
La texture représente également un descripteur bas niveau efficace utilisé dans le cadre de l'indexation et la recherche par le contenu. La texture peut être vue comme un ensemble de pixels spatialement agencés selon un certain nombre de relations spatiales. Il existe plusieurs méthodes d'extraction des caractéristiques de textures parmi eux : les filtres de Gabor, les méthodes statistiques, les méthodes fréquentielles, etc [8].

### 3.1.3. Descripteurs de la forme

Les descripteurs de formes permettent de représenter une information pertinente sur le contenu de l'image et précisément sur la forme. Il existe plusieurs descripteurs de formes comme: CSS (Curvative Scale Space descriptors), les filtres de convolution, les descripteurs de Fourier etc [8].

### 3.1.4. Descripteurs de points d'intérêt

Il est parfois indispensable d'extraire des descripteurs locaux, surtout pour détecter des sémantiques qui ne couvrent qu'une partie de l'image. Les descripteurs locaux peuvent être extraits des régions de l'image. Ces régions sont obtenues soit par segmentation de l'image entière ou par la recherche des points d'intérêt. Dans ce contexte, Lowe [9] propose des descripteurs appelés SIFT (Scale Invariant Feature Transform).



**Figure 1.3** : Principe de calcul du descripteur SIFT [9].

Les descripteurs SIFT sont obtenus en construisant un vecteur de grande dimension représentant les gradients dans une région locale de l'image. Les points d'intérêt de l'image sont calculés en utilisant un détecteur (ex. Harris). En considérant un point d'intérêt P, le voisinage de P est décomposé en 16 blocs de 4\*4 pixels (voir la figure 1.3). Dans chaque bloc, un histogramme d'orientation de gradients est formé, en discrétisant l'orientation en 8 bins (correspondant aux différentes orientations possibles : {0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°}). Le vecteur représentatif du descripteur aura dans ce cas 128 composantes (4\*4\*8=128). Les descripteurs SIFT se montrent invariants à l'échelle et à la rotation et robuste au bruit et au changement de l'illumination. Généralement, les SIFTs donnent les meilleurs résultats malgré leur grande dimension [1].

### 3.2. Descripteurs de haut niveau

Les descripteurs de haut niveau sont des descripteurs qui passent par une phase d'apprentissage. On distingue deux approches pour la génération de ces descripteurs, une pour les descripteurs sémantiques qui prend en considération de l'information sémantique (ex., Les relations entre concepts), et une autre approche qui utilise un apprentissage profond et exploite les résultats intermédiaires de cet apprentissage pour construire des descripteurs appris près à être utilisés dans un système de détection de concepts.

### 3.3 Optimisation des descripteurs

Avant d'utiliser un descripteur dans un système de détection de concepts, il est recommandé de le soumettre à une chaîne de prétraitements. Cette phase est appelée : optimisation des descripteurs. L'optimisation des descripteurs passe par deux étapes importantes : la normalisation des descripteurs et la réduction de dimensionnalité. La normalisation consiste à remettre sur la même échelle les valeurs du vecteur représentant le descripteur. La deuxième étape, comme son nom l'indique, est une méthode permettant de réduire le nombre de composantes formant le descripteur. Un descripteur optimisé a tendance à être plus efficace dans une approche de classification [1].

## 4. Modélisation/prédiction

Après l'extraction des descripteurs et leur optimisation. On utilise un apprenant pour générer un modèle permettant de prédire la détection du concept cible dans une image étant donné son descripteur. Généralement on opte pour un apprentissage supervisé par classification binaire. En plus des descripteurs normalisés, des annotations des images d'apprentissage sont utilisées (Deux classes : 1) Positive : l'image contient le concept et 2) négative : l'image ne contient pas le concept étudié). Plusieurs méthodes d'apprentissage peuvent être utilisées. Or, SVM demeure celle qui donne les meilleurs résultats dans le contexte de notre problématique. D'autres parts, il s'avère que les corpus utilisés pour la détection de concepts sont généralement déséquilibrés (Trop d'exemples négatifs et trop peu d'exemples positifs). Pour remédier à ce problème, une solution consiste à sous-échantillonner le corpus. Dans l'état de l'art, on trouve Multi-SVMs [10] qui donne nettement de meilleurs résultats que SVM classique.



**Multi-SVMs :**

*Safadi et al* [10] proposent une méthode qui consiste à combiner  $m$  classificateurs via une stratégie de « Bagging » où chacun d'entre eux utilise tous les échantillons d'apprentissage de la classe dominée (typiquement, la classe positive) et un ensemble d'échantillons de la classe dominante (typiquement, la classe négative) est tiré aléatoirement avec remise (bootstrap) :

$$m = (F_{neg} * N_{neg}) / (F_{pos} * N_{pos})$$

Où  $N_{pos}$  est le nombre d'exemples positifs,  $N_{neg}$  est le nombre d'échantillons négatifs,  $F_{neg}$  et  $F_{pos}$  sont des paramètres (entiers positifs non nuls) relatifs aux classes positive et négative, respectivement.  $F_{pos}$  gère la proportion des échantillons de la classe dominante qu'on veut utiliser, par rapport au nombre d'exemples de la classe dominé (ex. Deux fois plus d'exemples négatifs que positifs).  $F_{neg}$  quant à lui, permet de contrôler, à l'aide de  $F_{pos}$ , le nombre de classificateurs souhaité. L'ensemble d'apprentissage est divisé en  $m$  sous-ensemble, où chaque sous-ensemble contient tous les exemples positifs contenus dans le corpus de départ et  $(F_{pos} * N_{pos})$  exemples négatifs sont tirés aléatoirement avec remise. Ensuite chacun des  $m$  classificateurs est entraîné sur un sous-ensemble différent. On remarque que la contrainte  $(F_{neg} * N_{neg}) \geq (F_{pos} * N_{pos})$  doit être vérifiée. Finalement, les scores des  $m$  classificateurs sont fusionnés en utilisant n'importe quelle fonction possible, typiquement une moyenne. Plus la valeur de  $m$  est grande, meilleure est la performance finale. Il a été montré qu'utiliser SVM comme classificateur de base (Multi-SVMs) donne les meilleurs résultats dans le domaine de l'indexation des images. Nous allons utiliser dans notre travail cette méthode à base de SVM que nous appellerons MSVM.

**5. Fusion**

En général, décrire une image par un seul descripteur s'avère insuffisant. En effet, on peut décrire une image en se basant sur la couleur, la texture, ou en extrayant des points d'intérêts pour avoir une bonne description des images. Des expérimentations ont montré que la fusion de plusieurs descripteurs donne de meilleurs résultats pour la détection de concepts.

La fusion peut être appliquée à deux niveaux différents :

- ✓ Fusion de bas niveau, appelée « fusion précoce ».
- ✓ Fusion de haut niveau, appelée « fusion tardive »

La fusion précoce consiste à combiner un ou plusieurs descripteurs uni-modal/modaux pour générer une nouvelle représentation regroupant des informations issues de différentes modalités. Contrairement à la fusion précoce qui combine des informations de bas niveau, la fusion tardive fusionne des informations sémantiques, qui sont souvent des scores de classification/prédiction (ou des probabilités) renvoyés par des apprenants entraînés sur différents descripteurs [1].



## 6. Ré-ordonnement

Le ré-ordonnement (re-ranking en anglais) est une stratégie permettant d'améliorer la performance d'indexation en modifiant le score (ou la probabilité) des échantillons et ce, en utilisant les scores initiaux ainsi que d'autres sources d'information. Les nouveaux scores engendrent un nouveau classement des échantillons d'où le nom de « *reclassement* » ou « *ré-ordonnement* » ou « *re-scoring* » [1].

## 7. Les ontologies

Une ontologie est définie comme étant une spécification explicite d'une conceptualisation d'un domaine de connaissance, elle est généralement formée d'un ensemble de concepts abstraits qui sont organisés en relations hiérarchiques entre eux. Il y a une différence sémantique dans le type de relation hiérarchique qu'un homme peut facilement distinguer, mais celle-ci (la relation hiérarchique) doit être explicite pour la machine. Les concepts dans une ontologie sont d'une part liés à des mots lisibles par l'homme (littéraux) et d'autre part, ils sont liés entre eux par des relations sémantiques. Plusieurs ontologies ont vu le jour, on peut citer : Wordnet, LSCOM [1], etc.

## 8. Utilisation du contexte pour la détection de concepts

Afin d'améliorer la performance d'un système initial de détection de concepts dans des images, on pourrait utiliser le contexte. Qu'est ce que le contexte ? Pourquoi l'utiliser ? Comment l'utiliser ? Ce sont des questions auxquelles nous allons répondre dans le chapitre 3.

## 9. Protocole d'évaluation des systèmes de détection de concepts

Pour améliorer la qualité d'un système de détection de concepts une évaluation de ces annotations automatiques résultant est nécessaire. Pour cela, il existe plusieurs mesures d'évaluations que nous allons détailler dans ce qui suit.

### 9.1. Mesures d'évaluation

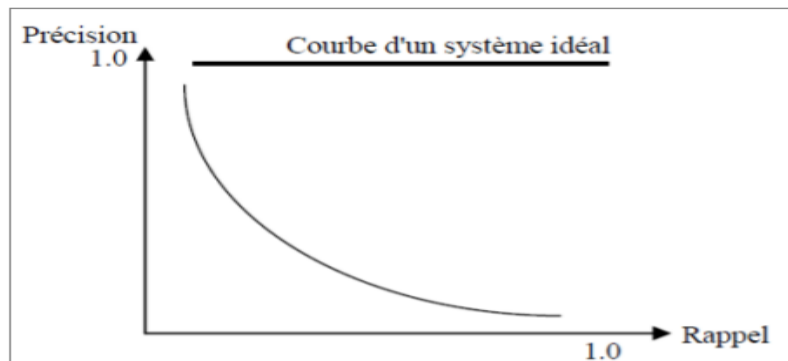
**La précision** est le rapport entre le nombre d'images pertinentes dans l'ensemble des images trouvées et le nombre d'images trouvées.

$$précision = \frac{|documents\ pertinents \cap documents\ retournés|}{documents\ retournés}$$

**Le rappel** est le rapport entre le nombre d'images pertinentes dans l'ensemble des images trouvées et le nombre d'images pertinentes dans la base d'images.

$$rappel = \frac{|documents\ pertinents \cap documents\ retournés|}{documents\ pertinents}$$

**La courbe rappel/précision** permet de suivre la qualité des résultats en fonction du nombre d'images retournées par le système en réponse d'une requête.



**Figure 1. 4 :** Courbe de rappel-Précision.

Il existe plusieurs autres mesures basées sur la précision et le rappel:

- **P(10), P(30), P(N<sub>r</sub>)** : mesure la précision atteinte dans le top de la liste des 10, 30, N<sub>r</sub> document retournés, respectivement.
- **R(10), R(30), R(N<sub>r</sub>)** : mesure le rappel atteint dans le top de la liste des 10, 30, N<sub>r</sub> document retournés, respectivement.
- **Average Precision (AP)** : mesure la précision moyenne non interpolée.

L'AP a l'avantage de résumer la courbe "rappel-précision" en une seule valeur. Elle est largement utilisée comme la mesure officielle dans plusieurs campagnes de recherche d'images et de vidéos, comme TRECVID et Pascal Voc. L'AP est définie par la formule suivante.

$$AP = \frac{1}{R} \sum_{j=1}^S \frac{R_j}{j} \times I_j$$

Où  $R$  est le nombre de documents pertinents dans le un corpus contenant  $S$  documents. A chaque indice  $j$ ,  $R_j$  est la précision après  $j$  documents retournés, et  $I_j$  est égal à 1 si le document  $j$  est pertinent, et 0 sinon.

- **Inferred Average Precision( InfAP)** : mesure la précision moyenne inférée.

## 9.2. Campagnes d'évaluation

Parmi les campagnes d'évaluation reconnues par la communauté qui permettent d'évaluer les systèmes d'indexation et de valider leur évolution on peut citer : ImageClef, Pascal-Voc, Trecvid, etc. C'est campagnes ont comme objectifs le développement de la communication entre l'industrie, l'académie et le gouvernement en mettant en place un forum ouvert pour faciliter les échanges d'idées sur la recherche. Dans notre travail nous allons utiliser le corpus Voc 2012 qui est mis à disposition dans le contexte du défi Pascal-VOC.

## **10. Conclusion**

A travers ce chapitre nous avons pu définir un système de détection de concepts. Nous avons détaillé les différentes étapes le composant en présentant les différentes mesures utilisées pour évaluer et valider un tel système.

Le chapitre qui suit est dédié à la description d'un système de détection de multi-concepts.

## **Chapitre 2**

# **Systeme de detection de multi-concepts**

## 1. Introduction

Avec l'évolution des systèmes de recherche d'informations, les besoins des utilisateurs deviennent plus abstraits et un plus grand nombre de mots composent leurs requêtes. Il devient donc important de penser à l'indexation de documents multimédias avec plus que des concepts individuels, pour aider les systèmes à répondre à des questions aussi complexes. A travers ce chapitre, nous allons présenter la tâche de détection de multi-concepts ainsi que les approches de détection d'un groupe de concepts simultanément.

## 2. La tâche de détection de multi-concepts

Un multi-concept est défini comme la combinaison d'un ensemble fini de concepts singuliers. On peut prendre comme exemple : « voiture+Panneau+personne+vélo ».

Considérer qu'une image est indexée par des multi-concepts est utile : une simple combinaison d'un ensemble de concepts peut présenter d'autres sémantiques pouvant être complexes. Par exemple, la combinaison des concepts « neige », « montagne » et « personne(s) en train de se déplacer » pourrait être liée à une scène de « skieur » ou « une compétition de ski ». Cette problématique est liée à celle de l'utilisation du contexte qui sera détaillé dans le chapitre qui suit

La difficulté de la tâche d'indexation automatique des images par des concepts singuliers n'est plus à prouver, spécialement pour les concepts rares ou difficiles à représenter ou décrire visuellement d'une manière efficace. Cette difficulté s'accroît encore plus quand on veut vérifier l'occurrence conjointe de  $N$  concepts dans un même document. En effet, même dans le cadre d'une paire de concepts (bi-concepts) le défi reste difficile. Cette remarque est confirmée par les résultats médiocres en termes de performance.

En réalité, la requête d'un utilisateur est souvent plus complexe qu'un seul concept peut représenter. Par exemple, la requête : « *une image montrant un cheval à coté d'une voiture* ». Pour répondre à cette requête, un premier type d'approches consiste à utiliser un détecteur de « *voiture* » et un détecteur de « *cheval* » et combiner leurs prédictions. C'est en effet l'approche dominante dans la littérature.

Une autre approche consiste à considérer la combinaison des concepts comme un nouveau concept. Par exemple, pour le cas de deux concepts (bi-concept), nous définissons un bi-concept comme la cooccurrence de deux concepts différents, où sa pleine signification ne peut être déduite d'une de ses concepts (notons que cette définition reste valable pour  $N$  concepts). Selon cette définition, il y a certaines combinaisons de deux concepts qui ne sont pas des bi-concepts, car « cheval + animal » ne donne pas plus d'information que cheval. En outre, les concepts singuliers en plusieurs balises telles que « cheval blanc » et « conducteur de voiture » ne sont pas des bi-concepts puisque les deux font référence au même concept visuel [4].

Nous présentons ces deux types d'approches de détections de multi-concepts dans les images dans ce qui suit.

### 3. Description des approches de détection de multi-concepts

On distingue deux approches possibles pour détecter simultanément un groupe de concepts :

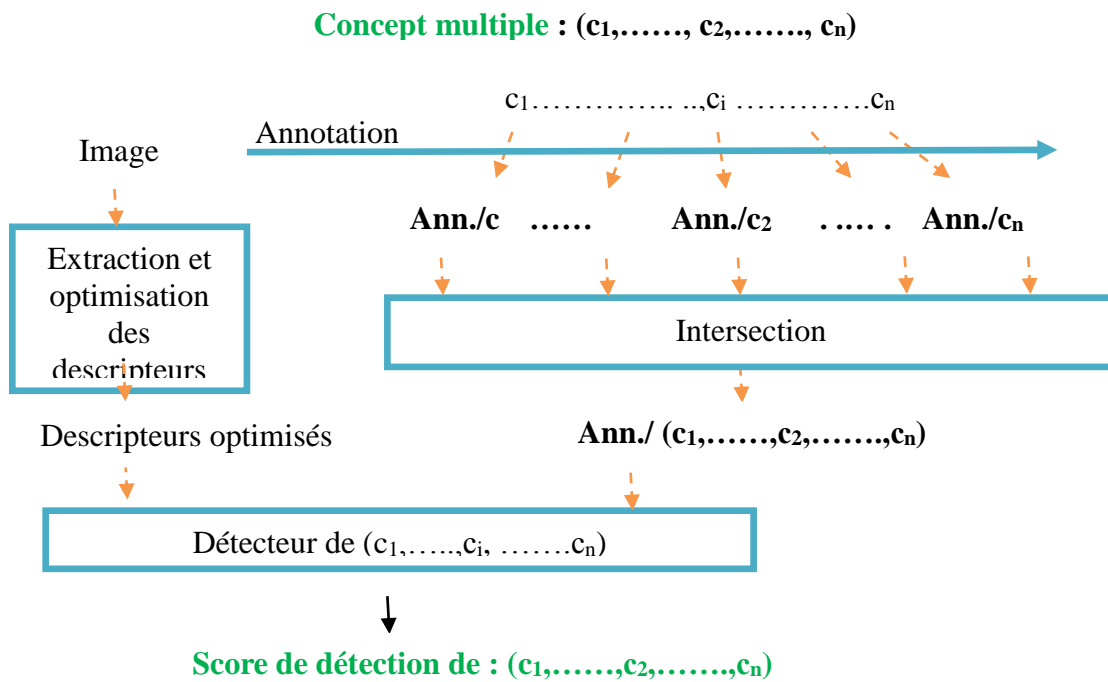
- ✓ Considérer le groupe de concepts comme un nouveau concept et générer un modèle spécifique pour chaque groupe ;
- ✓ Détecter l'ensemble des concepts formant le groupe séparément et combiner ensuite les résultants de leurs détections respectives ;

#### ✓ **Apprentissage direct d'un détecteur de multi-concept**

Cette méthode consiste à générer un modèle spécifique pour chaque multi-concept. Elle se base uniquement sur les données relatives aux concepts singuliers.

Étant donné un ensemble d'échantillons annotés par un ensemble de concepts singuliers, la génération des annotations des mêmes échantillons par un multi-concept est réalisée par une intersection des annotations par les concepts composant ce multi concept. Cette intersection donne généralement peu d'exemples positifs et beaucoup d'exemples non annotés. Les échantillons non annotés ne seront pas utilisés dans l'étape d'apprentissage. Ce phénomène connu sous le nom de « classes déséquilibrées » dans le domaine d'apprentissage et de classification, complique la situation et affecte négativement la performance des classificateurs. Pour pallier cet inconvénient, on peut opter pour une méthode d'ensembles et plus précisément le « Bagging ». Ce genre d'approches est compatible avec le problème de classes déséquilibrées. La méthode « MSVM » qui est décrit dans le chapitre précédent donne de bons résultats dans le contexte de la détection de concepts visuels dans les images et elle est compatible avec le problème de classes déséquilibrées [10].

Généralement, plusieurs types de descripteurs sont utilisés pour décrire le contenu d'une image, et une fusion tardive des résultats obtenus par les différents descripteurs utilisés est réalisée. Une somme pondérée peut être utilisée comme fonction de fusion où les poids sont des scores de confiance [3]. La figure 2.1 illustre les étapes d'apprentissage direct d'un détecteur de multi-concept.



**Figure 2.1:** Apprentissage direct d'un détecteur de multi-concept [3].

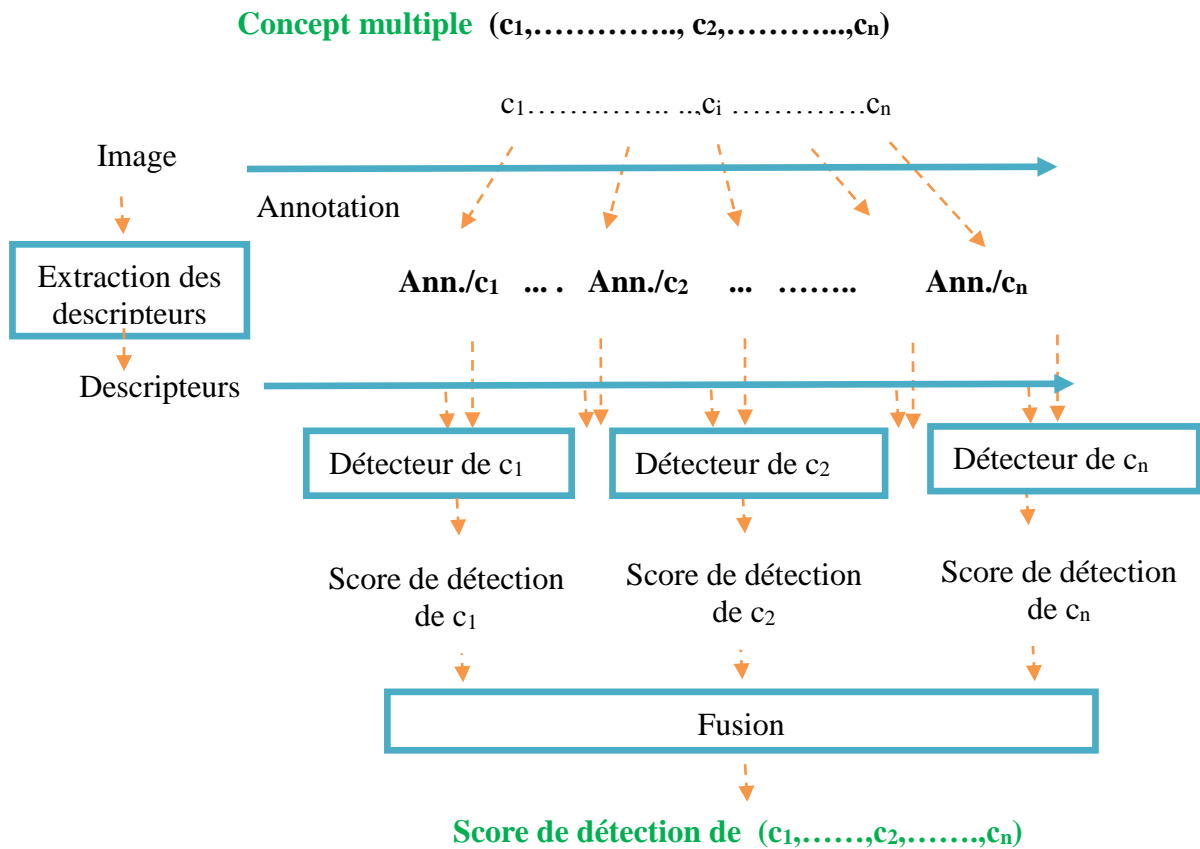
✓ **Fusion de détecteurs de concepts individuels**

Les méthodes qui consistent à fusionner les scores de détection de concepts singuliers formant le multi-concept ont un grand avantage sur la méthode d'apprentissage directe : elles ne nécessitent pas de lourds traitements. Certains exemples de telles approches ont été proposés dans [3]. Ces méthodes n'ont aucun paramètre à apprendre et/ou à optimiser, ce qui permet leur application directement et facilement sur n'importe quel concept multiple [3]. Les détecteurs des concepts singuliers sont générés séparément, comme montré dans le chapitre 1. En ce qui concerne la fonction de décision plusieurs méthodes de fusion de scores peuvent être utilisées telles que :

- ✓ La fusion linéaire : combinaison linéaire des scores, pondérée ou pas
- ✓ La méthode basée sur la notion de probabilité : fonction de produit des scores

La figure 2.2 illustre les étapes de l'approche de fusion des détecteurs de concepts individuels.

Les deux approches décrites dans cette section restent applicables sur n'importe quel type de documents multimédia.

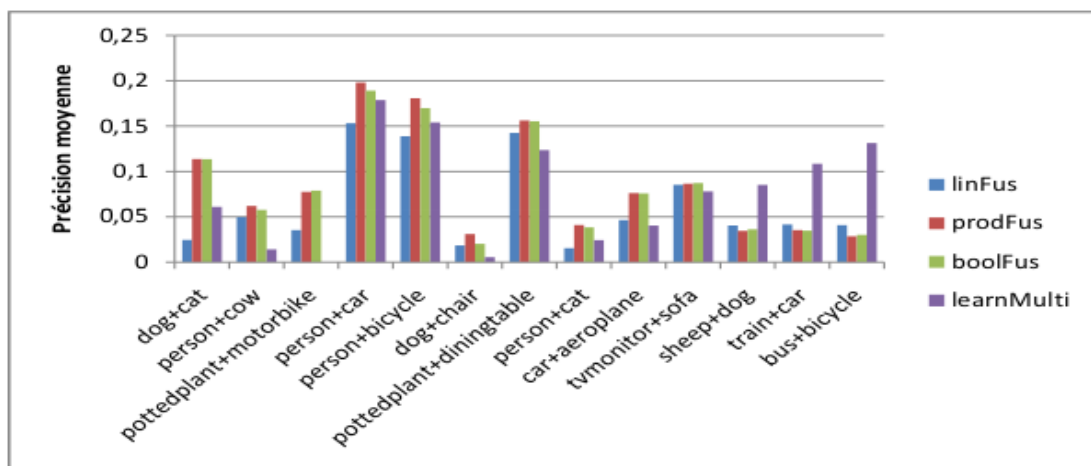


**Figure 2.2** : Fusion de détecteurs de concepts individuels [3].

#### 4. Discussion

Les deux approches de détection de multi-concepts détaillées précédemment ont été testées et évaluées dans [1] et [3] pour la détection de paires et triplets de concept dans les images. L'évaluation a été conduite sur la collection Pascal Voc 2012.

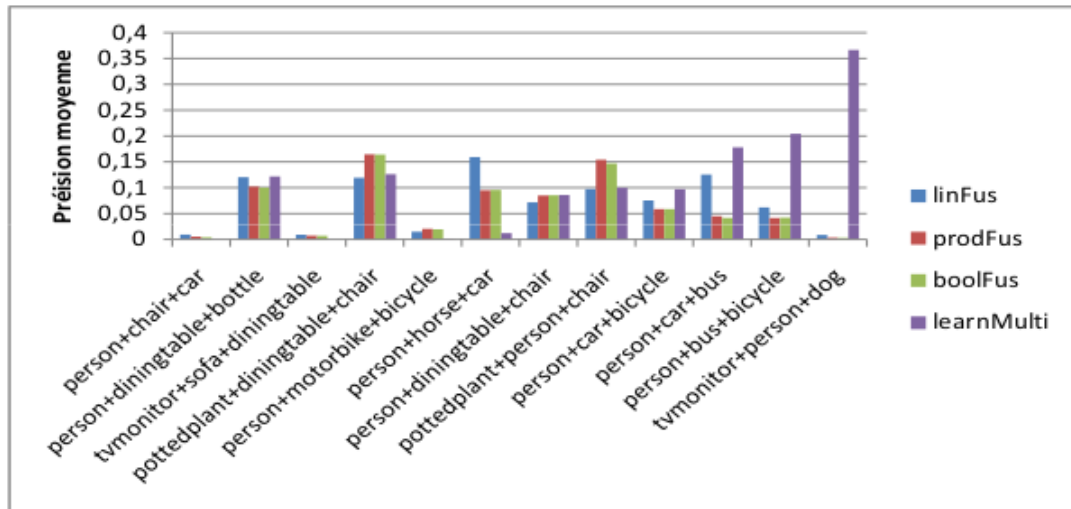
La figure 2.3 décrit les résultats obtenus pour certaines paires de concepts en termes de précision moyenne.



**Figure 2.3** : Résultats (AP) de la détection de bi-concept (AP) par différentes approches sur la collection Pascal VOC 2012 [1].



La figure 2.4 décrit les résultats obtenus pour certains tri-concepts, en termes de précision moyenne.



**Figure 2.4 :** Résultats (AP) de la détection de tri-concept (AP) par différentes approches sur la collection Pascal VOC 2012 [1].

Nous remarquons dans les figures 2.3 et 2.4 que la performance de détection varie d'un multi-concept à un autre. En effet, certains bi-concepts (respectivement tri-concepts) sont mieux détectés que certains autres avec une différence significative en termes de précision moyenne. D'autre part, la meilleure performance ne dépasse pas une AP de 20% pour les bi-concepts et 37% pour les tri-concepts. Pour certains multi-concepts la précision moyenne n'atteint même pas les 1 %.

Ces mauvais résultats reflètent la difficulté de la tâche de détection de multi-concepts. D'autre part, on peut expliquer cela par les problèmes rencontrés suivants :

- Le fossé sémantique qui est plus accru que dans le cas des concepts singuliers
- le problème de disponibilité de données : des exemples d'apprentissage annotés par les multi-concepts surtout les échantillons positifs. D'autre part, la variation des contenus de la classe négative (en termes de contenu visuel et sémantique) complique la tâche à l'apprenant pour trouver une modélisation efficace avec un bon pouvoir de généralisation
- La difficulté de trouver un descripteur visuel efficace pour l'ensemble des multi-concepts

## 5. Conclusion

A travers ce chapitre nous avons décrit les approches de détection des multi-concepts dans les images et nous avons présenté quelques résultats obtenus par l'application de ces deux approches. Le chapitre suivant est dédié à l'utilisation du contexte pour la détection de concepts dans les images.

# **Chapitre 3**

## **Utilisation du contexte pour la détection de concepts dans les images**

## 1. Introduction

Dans la communauté de chercheurs, il existe un large accord au sujet du rôle précieux que joue le contexte dans une tâche de compréhension de l'image et dans l'augmentation de la robustesse d'un système. Afin d'utiliser efficacement le contexte, nous devons comprendre ce qu'est le contexte et comment il peut être utilisé.

## 2. Qu'est ce que le contexte ?

Bien que le terme « *contexte* » est fréquemment utilisé dans la vision par ordinateur, il manque une définition claire. On trouve dans l'état de l'art plusieurs définitions [11], [12], [13] [14]. Dans le cadre de notre travail nous choisissons de retenir la définition utilisée dans [1]: « *Le contexte est toute information additionnelle qu'un système de détection de base peut s'en passer, qui est pertinente et peut aider à améliorer la qualité de la détection* ».

## 3. Pourquoi avoir besoin du contexte ?

En raison de la richesse du contenu d'une image en termes de sémantiques, tenter de détecter un concept visuel seul est une idée très naïve. En effet, les concepts n'existent pas isolément, certains concepts co-occurrents toujours (*Animal et Véhicule*), certains autres très souvent (*Ciel et Avion*). D'autre part, la présence de certains concepts exclut l'occurrence de certains autres (*Féminin et Masculin*). Donc il est important d'utiliser des informations liées à d'autres concepts pour améliorer les performances initiales de détection d'un concept cible.

La figure 3.1 montre l'utilité du contexte pour la détection de concepts visuels.

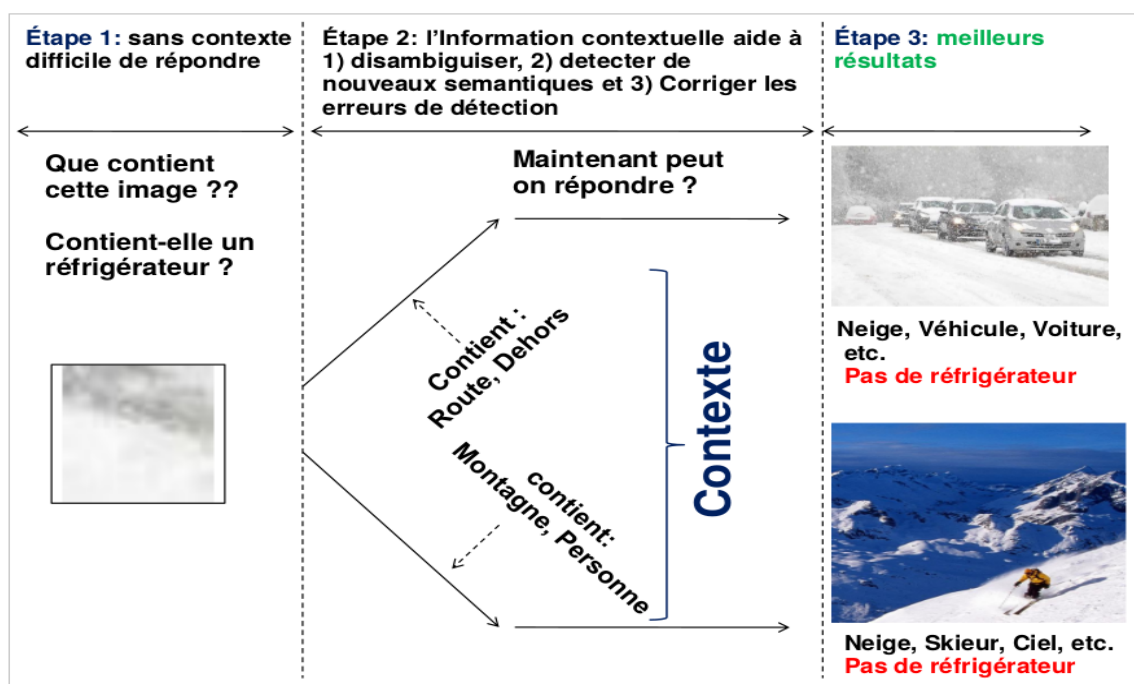


Figure 3.1 : Utilité du contexte pour la détection de concepts visuels [1].

## 4. Quelques types de contexte

Dans un système d'indexation d'images ou de vidéos on peut distinguer en fonction de la source d'information contextuelle différents type de contexte tel que:

### ✓ Le contexte temporel

Le contexte temporel est défini dans le cadre de documents non fixes (ex. Vidéos, audio, etc.). L'aspect temporel de ces documents donne la possibilité de les segmenter en échantillons dans l'axe du temps. On peut prendre l'exemple de la segmentation de vidéos en plans [1]. La source du contexte sémantique

### ✓ Le contexte sémantique

Dans l'état de l'art, le contexte sémantique a été considéré comme étant les relations entre concepts [1]. Ces relations peuvent être issues de corpus de données, ou définies par un expert humain. Dans ce dernier cas, l'expert peut fournir différentes formes de sources de relations sémantiques. Ces ressources peuvent être des ontologies, ou une liste de relation explicites entre concepts, comme par exemple, les relations d'implication (ex. *homme=>personne*), les relations d'exclusion (ex. *Masculin exclut Féminin*).

Dans notre travail nous nous intéressons plus à ce type de contexte, d'une part pour sa complexité raisonnable et d'autre part, parce qu'il permet de modéliser des systèmes génériques pouvant être appliqués pour la détection de n'importe quel multi-concept et dans n'importe quelle configuration expérimentale sans devoir apporter des modifications conséquentes.

### ✓ Le contexte d'échelle :

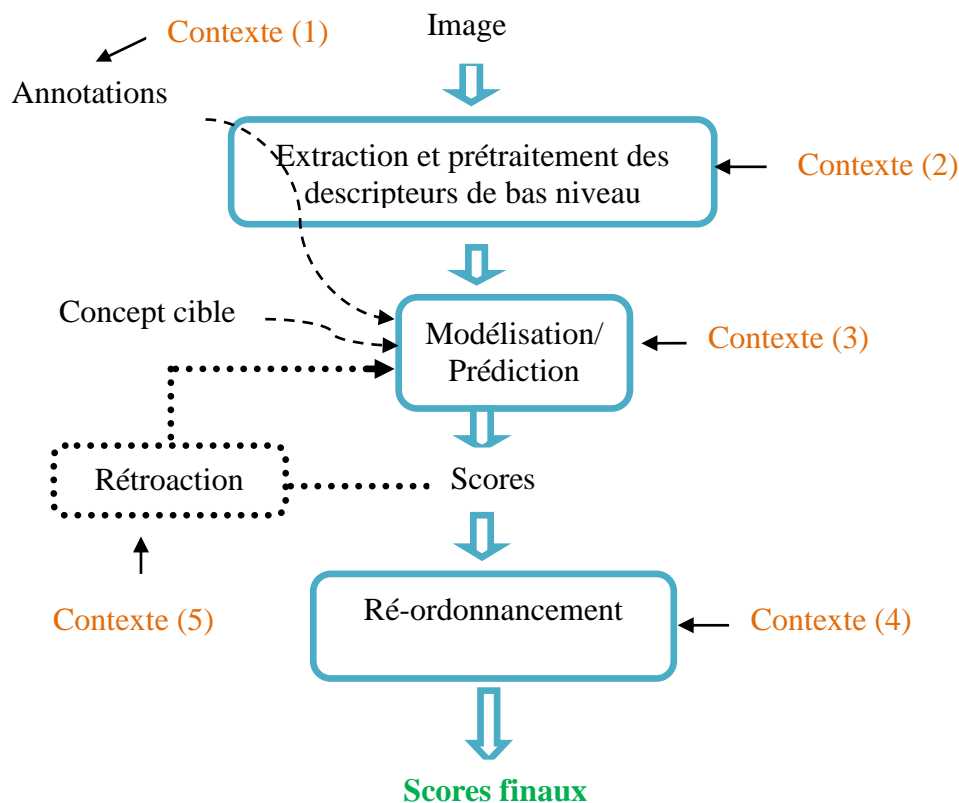
Le contexte d'échelle se base sur les relations d'échelles ou de tailles entre un objet et certains autres. Ceci dit, ce type de contexte exige qu'au moins un autre objet soit détecté et qu'un traitement soit réalisé de façon à déterminer les relations spatiales, de tailles et/ou de profondeurs entre l'objet cible et les autres objets l'entourant qui ont été détectés [1].

### ✓ Le contexte spatial :

Le contexte spatial s'avère crucial pour l'annotation locale des images comme décrit dans [15]. Pour reconnaître les catégories de scènes naturelles, [16] proposent d'extraire « une pyramide spatiale », qui consiste à diviser l'image en sous-régions de plus en plus fines et de calculer des histogrammes de caractéristiques locales extraites de chaque sous-région [1].

## 5. Comment utiliser le contexte dans un système de détection de concepts ?

Le contexte peut être utilisé dans plusieurs étapes d'un système de détection de concepts comme illustre la figure 3.2.



**Figure 3.2 :** Différentes possibilités d'utilisation du contexte dans un système d'indexation de documents multimédia [1] .

Le contexte peut être utilisé dans les étapes suivantes :

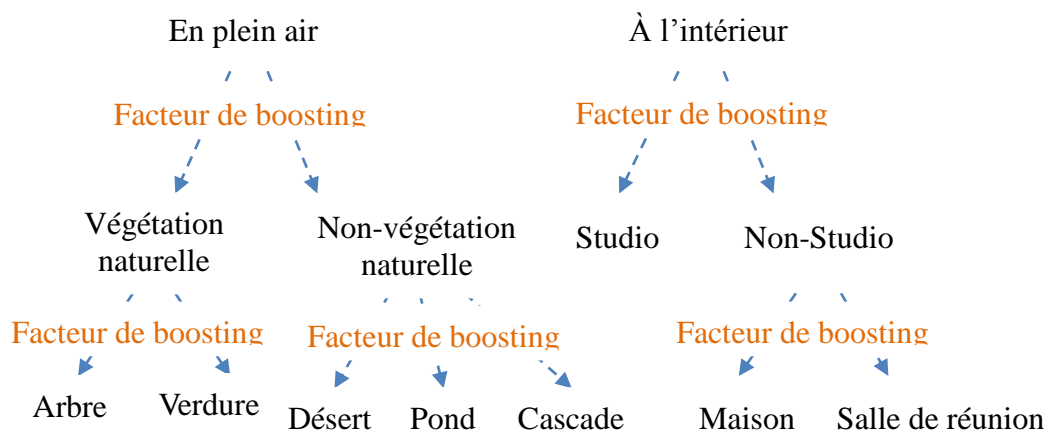
- **Prétraitement de données** (contexte (1) dans la figure 3.2) : le contexte est utilisé dans cette étape pour nettoyer les annotations et donc avoir des données moins vulnérables au bruit qui peut influencer l'étape d'apprentissage.
- **Extraction et optimisation des descripteurs** (contexte (2) dans la figure 3.2) : Le contexte peut être pris en compte dans cette étape en construisant un descripteur qui prend en compte des informations contextuelles.
- **Apprentissage/prédiction** (contexte (3) dans la figure 3.2) : le contexte est utilisé dans cette étape avec diverses manières :
  - En utilisant des noyaux prenant en compte des informations contextuelles ;
  - En développant une méthode d'apprentissage spécifique qui utilise le contexte ;
  - En utilisant des classificateurs discriminatifs pour intégrer l'information contextuelle, comme le boosting, SVM [17], ou des approches génératives comme Naïve Bayes ;

- **Post-classification** (contexte (4) dans la figure 3.2) : le contexte est utilisé dans cette étape de ré-ordonnement en fusionnant les données contextuelles avec les sorties des classificateurs afin de recalculer de nouveaux scores de détection ;
- **Rétroaction** (contexte (5) dans la figure 3.2) : l'information contextuelle dans cette étape est introduite dans chaque itération de l'étape d'apprentissage pour enrichir, corriger, ou appuyer les décisions des différents apprenants.

## 6. Quelques approches retenues utilisant le contexte sémantique

### ✓ Le Facteur de boosting

Cette méthode a été proposée dans [6]. Elle utilise une ontologie et se base sur les relations de type père-fils entre concepts. L'idée principale de la méthode *facteur de boosting* est d'utiliser les concepts « pères » de l'ontologie pour booster la performance des concepts « fils » par la modification de leurs scores et donc un reclassement des échantillons. Cette idée est motivée par le constat que les classificateurs des concepts fils situés dans les niveaux inférieurs de la hiérarchie de l'ontologie sont susceptibles d'être moins performants par rapport aux classificateurs des concepts situés dans les niveaux supérieurs. En effet, ces derniers ont l'avantage d'avoir plus de données positifs. Par exemple, dans la figure 3.3, le classificateur du concept « *En plein air* » va retourner une meilleure précision par rapport à celui de « Pond ». Notons que les concepts sont entraînés indépendamment les uns des autres [6].



*Figure 3.3 : Exemple d'apprentissage basé sur l'ontologie [6].*

Cette méthode considère que si la distribution des scores de détection entre le père et le fils a une corrélation élevée, il y a plus d'influence du père sur l'enfant. La modification du score de confidences en utilisant le facteur de boosting [6] est réalisée suivant l'algorithme ci-après

**Algorithme :**

*Input :*

**S** ; /\* Données

**C<sub>i</sub>** ; /\* Concept singulier

**{C<sub>j</sub>...C<sub>k</sub>}** ; /\*Ensemble des concepts pères de C<sub>i</sub>

**p(s|C<sub>i</sub>)** ; /\*Score de détection de C<sub>i</sub> dans S

*Output :*

**p̄(s|C<sub>i</sub>)** ; /\*Score modifié de détection de C<sub>i</sub> dans S

*Variables :*

**{λ<sub>p</sub><sup>i</sup>, ..., λ<sub>k</sub><sup>i</sup>}** /\* Les facteurs de boosting de {C<sub>j</sub>...C<sub>k</sub>} a C<sub>i</sub>

*Begin :*

(1) **For each** C<sub>l</sub> ∈ {C<sub>j</sub>...C<sub>k</sub>}

(2) λ<sub>l</sub><sup>i</sup> = A/(B + exp(C \* abs(p(s|C<sub>i</sub>)-p(s|C<sub>l</sub>))))

(3) **End for**

(4) **p̄(s|C<sub>i</sub>)**=p(s|C<sub>i</sub>)+λ<sub>j</sub><sup>i</sup> p(s|C<sub>j</sub>)+.....+λ<sub>k</sub><sup>i</sup>p(s|C<sub>k</sub>)

*End*

✓ **Ré-ordonnement sémantique basé sur une ontologie**

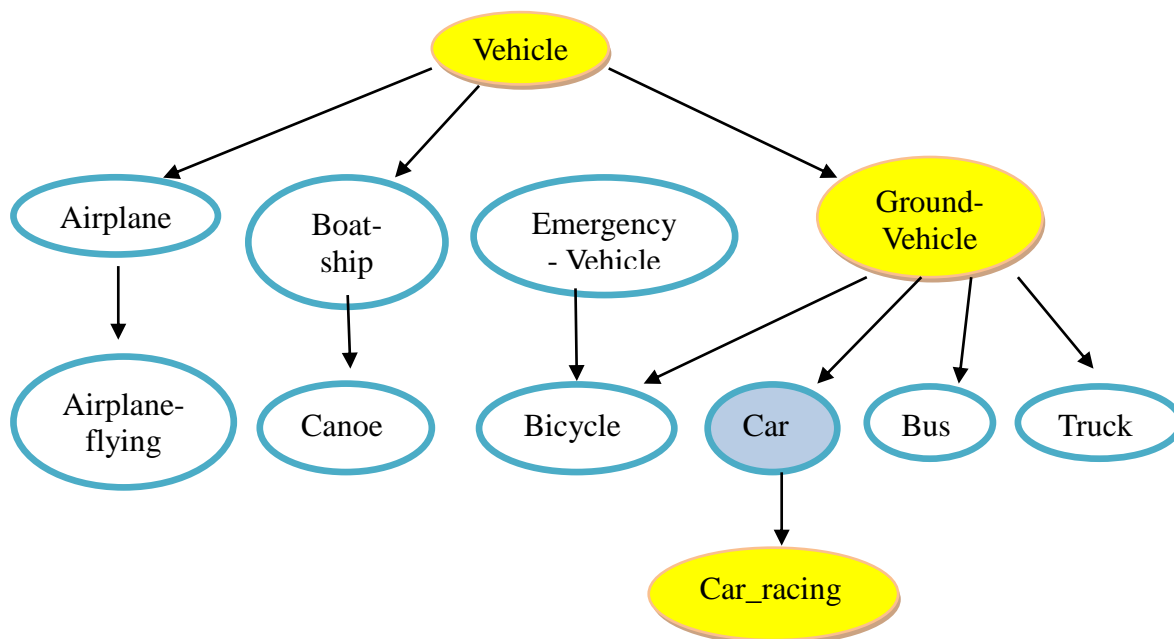
Cette approche utilise la hiérarchie d'une ontologie comme source de contexte sémantique et se sert également des informations extraites d'un corpus de données pour les combiner dans le but de gagner en performance et robustesse.

Le principe de cette approche consiste à modifier le score initial d'une première détection d'un concept cible dans un échantillon en exploitant les scores de détection d'un ensemble d'autres concepts dans ce même échantillon, pour générer un score final de détection basé sur le contexte.

Afin de déterminer les concepts à utiliser pour améliorer la détection d'un concept cible dans un échantillon de test, deux approches ont été proposées dans [5]: « *Ancêtres et/ou descendants* » et « *Famille de concepts* ».

**Ancêtres et/ou descendants**

Pour un concept cible, considérer uniquement les concepts qui sont des ancêtres ou des descendants du concept cible dans la hiérarchie de l'ontologie. La *figure 3.4* montre un exemple d'une telle sélection pour le concept « car » [5].



**Figure 3.4:** Sélection des ancêtres et descendants du concept cible dans la hiérarchie de concepts (ex. Car) [5].

### Famille de concept

Un expert humain sélectionne pour chaque concept cible un ensemble de concepts ayant un lien sémantique avec le concept cible. Cet ensemble est appelé: « *Famille de concept* ». Les relations entre les concepts induisent une structure en treillis dans l'ensemble de concepts. Au sein de cette structure, un expert humain peut identifier et extraire un certain nombre d'arbres ou des fragments, chacun représentant une hiérarchie de concepts compatible avec un sens commun. Les fragments résultants ne sont pas nécessairement déconnectés. En effet, les arbres concernant certains concepts peuvent se chevaucher, comme par exemple pour les deux concepts « *Voiture* » et « *sport* » dont leurs fragments respectifs pourraient contenir des concepts en commun concernant par exemple les sports impliquant l'utilisation d'une voiture (ex. « *Car-Racing* »). L'union des fragments de tous les concepts forme le corps entier de l'ontologie de départ. Ensuite, une famille initiale d'un concept cible est définie comme l'ensemble contenant tous les descendants de chaque ancêtre du concept cible (toute l'arborescence à laquelle le concept cible appartient). Après avoir sélectionné cette famille initiale, en se basant sur un corpus de développement (par validation croisée), on élimine les concepts qui n'aident pas à améliorer la détection du concept cible. Par exemple, comme montré dans la figure 3.4, la famille du concept « *Car* » contiendra : *Vehicle*, *Emergency-Vehicle*, *Ground-Vehicle*, *Airplane*, *Airplane-Flying*, *Bicycle*, *Bus*, *Truck*, *Car* et *Car-Racing*. De même, « *AirPlane-Flying* » aura dans sa famille : *Airplane*, *Airplane-Flying* et *Vehicle*. Même si la méthode de ré-ordonnancement a amélioré un système ayant une bonne performance, le gain n'était pas spectaculaire [5].

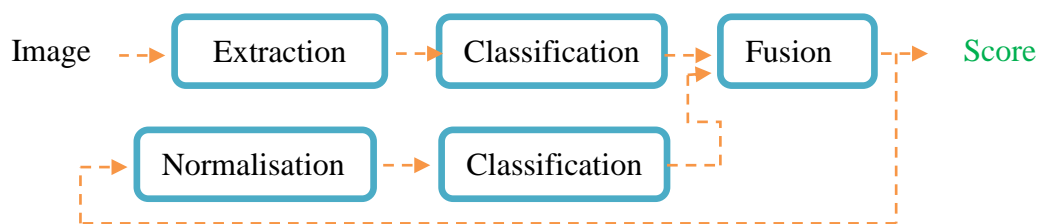


✓ **Ré-ordonnancement sémantique par regroupement**

Cette approche modélise le contexte sémantique en regroupant les exemples selon leurs contenus sémantiques issus d'une étape préalable d'apprentissage. Elle se base sur l'hypothèse que si on décrit les individus par leurs contenus sémantiques, les échantillons ayant un lien sémantique se verront regroupés dans l'espace. La mise à jour des scores se fait via une combinaison de valeurs calculées après une étape de regroupement (*clustering*) basé sur une description des données par les scores de détection d'un ensemble de concepts [1].

✓ **Rétroaction conceptuelle**

Cette approche consiste à construire un nouveau descripteur conceptuel de haut niveau à partir des scores obtenus via une première étape de classification et à l'utiliser de la même façon qu'un descripteur de bas niveau dans un système de détection de concepts en considérant une nouvelle étape d'apprentissage qui utilise le descripteur conceptuel généré. Pour la construction de ce descripteur, des sorties de détecteurs de concepts sont utilisés [1], comme montré dans la *figure 3.5*.



**Figure 3.5:** Système d'indexation sémantique avec rétroaction conceptuelle [1].

**7. Conclusion**

A travers ce chapitre nous avons décrit la notion du contexte ainsi que quelques approches d'utilisation du contexte sémantique pour la détection de concepts dans des images fixes.

# **Chapitre 4**

## **Contributions et expérimentations**

## 1. Introduction

Dans ce chapitre nous allons présenter nos approches d'utilisation du contexte sémantique pour la détection de multi-concepts dans les images fixes. Nous y détaillerons également les expérimentations réalisées et nous discuterons les résultats d'évaluation obtenus.

## 2. Description des approches proposées

### 2.1 Approche 1 : Facteur de boosting

Dans cette approche, nous nous sommes inspirés du principe de la méthode proposée par [6] décrite dans le chapitre 3, et nous l'avons adapté pour le cas des multi-concepts (bi-concepts et tri-concepts). Les multi-concepts ont été détectés par l'application de deux approches différentes : l'approche d'apprentissage direct d'un détecteur de multi-concept et l'approche de fusion de détecteurs de concepts individuels (voir chapitre 2).

Notre approche se base sur le fait que les classificateurs des multi-concepts pères sont plus performants que les classificateurs des multi-concepts fils. Pour cela, nous avons exploité les relations pères-fils entre les multi-concepts, en utilisant les multi-concepts pères pour booster la performance des multi-concepts fils, par la modification de leurs scores. Nous proposons de calculer un nouveau score de détection d'un multi-concept fils par la formule suivante:

$$\text{Nouveau\_score\_du\_fils} = \text{ancien\_score\_du\_fils} + \text{lambda} * \text{score\_du\_père}$$

$$\text{Où : } \text{Lambda} = A / (A + (B + \exp(C * \text{abs}(\text{ancien score du fils} - \text{score du père}))))$$

Où :  $A$ ,  $B$  et  $C$  sont des constantes.

La figure 4.1 illustre plus clairement notre approche.

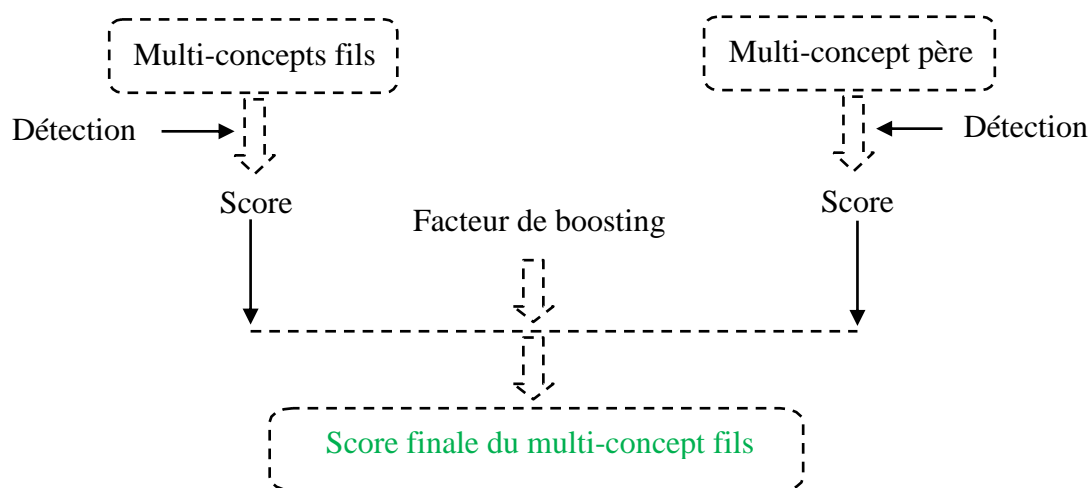
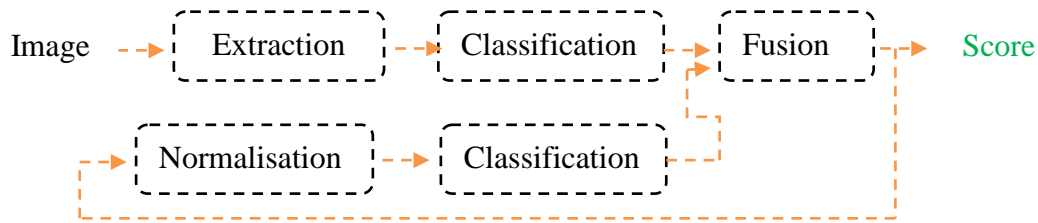


Figure 4. 1 : l'approche facteur de boosting.

### 2.2. Approche 2 : Rétroaction conceptuelle

Inspiré par les bons résultats de l'approche de rétroaction conceptuelle proposée par [1] qui consiste à construire un nouveau descripteur sémantique de haut niveau, à partir des scores obtenus par la détection initiale des concepts singuliers, puis à mettre à jour les scores de détection du concept singulier cible par une nouvelle étape d'apprentissage, qui utilise le

descripteur conceptuel généré. On a décidé dans notre approche d’exploiter ce principe dans le contexte de la détection d’un groupe de concepts simultanément, en considérant que les bi-concepts et les tri-concepts comme concepts multiple. La figure 4.2 résumé le principe de notre approche.



**Figure 4. 2** : L’approche rétroaction conceptuelle.

### 3. Expérimentation

#### 3.1. Données utilisées

✓ Corpus de données

L’évaluation de nos approches proposées pour l’utilisation du contexte pour la détection de paires et de triplets de concepts, a été réalisée sur le corpus international standard pascal voc 2012. Ce dernier est divisé en deux sous-corpus : un pour l'apprentissage et l'autre pour le test. Le tableau 4.1 présente quelques détails concernant le corpus de données.

	Apprentissage	Test
Nombre d’images	5717	5823

**Tableau 4. 1** : La collection voc 2012.

✓ Concepts

On a formé les paires et les triples de concepts à partir de la liste principale des 20 concepts fournis par la communauté pascalle voc, en supprimant ceux pour lesquels on a trop peu d’exemples positifs, on a gardé à la fin 60 paires et 45 triples. Le tableau 4.2 résume le nombre de concepts utilisé.

Concepts		
Singuliers	Doubles	Triples
20	60	45

**Tableau 4. 2** : le nombre de chaque type de concept.

✓ Catégories

On a catégorisé les concepts ci-dessus en 3 catégories où chacune d'entre elles correspond à un type de concept comme montré dans le tableau 4.3.

Catégories		
Singuliers	Doubles	Triples
4	9	8

**Tableau 4.3** : le nombre de chaque type de catégorie.

✓ Annotations

- Les annotations des images par les concepts singuliers ont été fournies par la collection voc 2012.
- Les annotations des images par les multi-concepts ont été réalisées par une intersection des annotations des concepts singuliers composant ce multi-concept. (voir chapitre 2)
- Les annotations des images par des catégories ont été générées à partir des concepts de la même catégorie.

✓ Descripteurs

Comme entrée à nos détecteurs, on a utilisé trois descripteurs :

Descripteur	Caractéristique
Histogramme de couleurs	Descripteur de bas niveau, globale.
SIFT	Descripteur de bas niveau, locale.
GoogleNet	Descripteur de haut niveau.

**Tableau 4.4** : Descripteurs utilisés.

### 3.2. Méthode d'apprentissage utilisée

On a choisi d'utiliser MSVM [10] comme classificateur supervisé, en raison de ses bons résultats dans le domaine des documents multimédia (voir chapitre 1).

### 3.3. Détecteurs de concepts

On a utilisé trois types de détecteurs de concepts, qui diffèrent selon le descripteur utilisé en entrée :

- ✓ En utilisant l'histogramme de couleurs.
- ✓ En utilisant Le SIFT.
- ✓ En utilisant GoogleNet.

### 3.4. Fusion

Une fusion tardive des scores obtenus dans l'étape de détection initiale des concepts est appliquée dans le but d'améliorer les performances. Cela est réalisé en fusionnant pour chaque exemple, les scores obtenus par les différents détecteurs.

### 3.5. Détecteurs de multi-concepts

On a utilisé deux détecteurs de multi-concept, en appliquant les deux approches de détection d'un groupe de concepts simultanément citées dans la section 2.1 :

- ✓ En utilisant l'approche de détection directe d'un multi-concept.
- ✓ En utilisant l'approche de détection par fusion des détecteurs des concepts individuels.

## 4. Processus d'expérimentation

Avant de tester et évaluer l'expérimentation sur nos approches, on a lancé une optimisation des paramètres SVM et des coefficients A, B, C (utilisée dans l'approche 1) sur le corpus d'apprentissage, afin d'aboutir à de meilleurs résultats.

### 4.1. Approche 1 : Facteur de boosting

Les expérimentations sur notre approche ont été déroulées comme suit :

✓ Expérimentation 1 :

- 1) On a lancé MSVM pour détecter les concepts singuliers fils en utilisant l'un des descripteurs cités dans la section 3.1 séparément.
- 2) On a appliqué l'étape 1 sur les concepts singuliers pères.
- 3) Pour chaque descripteur et après avoir eu les scores de détection des concepts singuliers pères et fils, une modification de score pour chaque concept singulier fils est effectuée.
- 4) On a appliqué une fusion tardive des scores de détection obtenus par les différents descripteurs de bas niveaux.
- 5) On a évalué nos résultats en utilisant la précision moyenne comme mesure de performance.

✓ Expérimentation 2 :

- 1) Pour chaque approche de détection de multi-concept (voir section 2.1) on a lancé MSVM pour détecter les concepts multiple fils paires et triple en utilisant les descripteurs cités dans la section 3.1 séparément.
- 2) On a appliqué l'étape 1 pour détecter les concepts multiples pères.
- 3) Pour chaque descripteur et après avoir eu les scores de détection des concepts singuliers pères et fils, une modification de score pour chaque concept multiple fils est effectuée.
- 4) On a appliqué une fusion tardive des scores de détection des différents descripteurs de bas niveau.
- 5) On a évalué nos en utilisant la précision moyenne comme mesure de performance.

## 4.2 Approche 2 : rétroaction conceptuelle

Les expérimentations sur notre approche ont été réalisées en appliquant les étapes suivantes :

- 1) Détection initiale classique des multi-concepts en utilisant l'un des différents descripteurs de bas niveau séparément.
- 2) Détection classique des concepts singuliers pour chaque multi-concept.
- 3) génération d'un descripteur sémantique à partir des scores de détection des concepts singuliers
- 4) Fusion du descripteur sémantique avec le descripteur initial de bas niveau.
- 5) Re-détection des multi-concepts en considérant le descripteur résultant de l'étape 4
- 6) Fusion des scores de détection des descripteurs de bas niveaux.
- 7) Evaluation des résultats.

Remarque : Les expérimentations des deux approches ont été réalisées séparément pour :

- ✓ Chaque type de concept (singulier, paire, triple)
- ✓ Chaque descripteur.
- ✓ Chaque méthode de détection de multi-concept.

## 5. Résultats et discussions :

L'approche « facteur de boosting » :

- ✓ Détection des concepts singuliers

	Système de base	Facteur de boosting
<b>Histogramme</b>	0.1250	0.1301 (+4.08%)
<b>SIFT</b>	0.1836	<b>0.1952 (+6.32%)</b>
<b>Fusion</b>	0.2283	0.2402 (+5.21%)
<b>GoogleNet</b>	0.8081	0.7989 (-1.2%)

**Tableau 4.5** : Résultats (MAP (gain %)) de notre approche et une comparaison avec les résultats du système de base pour les concepts singuliers.

### Discussion :

Le tableau 4.5 présente une comparaison entre les résultats obtenus par notre approche : Facteur de boosting et notre système de base. Nous remarquons que l'approche : Facteur de boosting améliore mieux les résultats pour le descripteur « SIFT » atteignant un gain relatif  $((0.1952-0.1836)/0.1836)$ : +6.32%). Par opposition, ce gain est moins bon pour le descripteur « Histogramme » et pour « la fusion ». Nous remarquons aussi que l'approche n'a pas amélioré le résultat du descripteur « GoogleNet ». Cette détérioration s'explique par la performance extraordinaire atteinte par ce descripteur. Nous savons qu'il est plus difficile d'améliorer la performance d'un système qui est assez bon. Nous supposons que

l'apprentissage profond à conduit à une saturation d'exploitation des informations sémantiques. En effet, dans cette situation, notre approche apporte un bruit qui mène à une détérioration de performances.

✓ Détection de concepts multiples

- Détection de bi-concepts: approche par modèle

	Système de base	Facteur de boosting
<b>Histogramme</b>	0.0145	0.0164 (+13.10%)
<b>SIFT</b>	0.0197	0.0232 (+17.67%)
<b>Fusion</b>	0.0276	<b>0.0337</b> (+22.10%)
<b>GoogleNet</b>	0.1802	0.1885 (+4.60%)

**Tableau 4. 6** : Résultats (MAP (gain %) de notre approche et une comparaison avec les résultats du système de base pour les concepts doubles.s

Discussion

Le tableau 4.6 montre les résultats de notre approche appliquée sur un détecteur de bi-concepts (approche de modèle). Nous remarquons que l'approche améliore tous les résultats, atteignant un gain allant de 4% à 22%. Le meilleur résultat a été obtenu en appliquant la fusion tardive sur les résultats de détection des deux descripteurs « Histogramme et SIFT ». Dans ce cas, la précision moyenne (MAP) est égale à 0.0337.

- Détection de bi-concepts: approche fusion des concepts singuliers

	Système de base	Facteur de boosting
<b>Histogramme</b>	0.0203	<b>0.0221</b> (+8.90%)
<b>SIFT</b>	0.0273	0.0273 (0%)
<b>Fusion</b>	0.0371	0.0396 (+6.74%)
<b>GoogleNet</b>	0.2707	0.2650 (-2.11%)

**Tableau 4. 7** : Résultats (MAP (gain %) de notre approche et une comparaison avec les résultats du système de base pour les concepts doubles.

Discussion



Le tableau 4.7 illustre les résultats de notre approche, ainsi qu'une comparaison avec le système initial. On remarque que l'approche améliore les résultats pour le descripteur « Histogramme » et ainsi pour la fusion, mais le gain n'est pas significatif. En effet, la MAP du système initial est basse. De plus, cette amélioration n'est pas stable. Ceci est confirmé pour la dégradation des résultats pour le descripteur GoogleNet et la non amélioration des résultats pour SIFT.

- Détection de tri-concepts : approche par modèle

	Système de base	Facteur de boosting
<b>Histogramme</b>	0.0033	0.0050 (+51.52%)
<b>SIFT</b>	0.0056	0.0085 (+51.78%)
<b>Fusion</b>	0.0061	<b>0.0109</b> (+78.68%)
<b>GoogleNet</b>	0.0567	0.0607 (+7.05%)

**Tableau 4. 8 :** Résultats (MAP (gain %)) de notre approche et une comparaison avec les résultats du système de base pour les concepts triples.

#### Discussion

Le tableau 4.8 présente les résultats en MAP de notre approche et celle du système de base. On remarque avec notre approche on a obtenu des résultats intéressants atteignant un gain relatif allant de +7% jusqu'à +78%. De plus même pour le concept, GoogleNet qui a une bonne performance comparé aux trois autres résultats, on a eu une amélioration intéressante.

- Détection de tri-concepts : approche par fusion de concepts singuliers

	Système de base	Facteur de boosting
<b>Histogramme</b>	0.0088	0.0106 (+20.45%)
<b>SIFT</b>	0.0103	0.0117 (+13.60%)
<b>Fusion</b>	0.0152	0.0176 (+15.79%)
<b>GoogleNet</b>	0.0797	<b>0.1186</b> (+48.80%)

**Tableau 4. 9 :** Résultats (MAP (gain %)) de notre approche et une comparaison avec les résultats du système de base pour les concepts triples.

Discussion

On remarque à travers le tableau 4.9 que notre approche améliore les résultats atteignant un gain relatif en MAP allant de 13% à 48%. Notons le résultat atteint pour GoogleNet, qui peut réaliser avec les résultats de l'état de l'art

**L'approche rétroaction conceptuelle**✓ Détection des concepts singuliers

	Système de base	Rétroaction conceptuelle
<b>Histogramme</b>	0.1250	0.1237 (-0.01%)
<b>SIFT</b>	0.1836	<b>0.2058 (+12.10%)</b>
<b>Fusion</b>	0.2283	0.2437 (+6.75%)
<b>GoogleNet</b>	0.8081	0.8004 (-9.50%)

**Tableau 4. 10** : Résultats (MAP (gain %)) de notre approche et une comparaison avec les résultats du système de base pour les concepts singuliers.

Discussion :

Le tableau 4.10 présente les résultats de notre approche et du système initial. On remarque que notre approche améliore mieux pour le descripteur SIFT atteignant un gain relatif de +12.10% en termes de MAP. Nous remarquons une dégradation de performances pour le descripteur GoogleNet. Ceci peut être expliqué comme cité précédemment, par la très bonne performance initiale du descripteur.

✓ Détection de concepts multiples

- Détection de bi-concepts :approches par modèles

	Système de base	Rétroaction conceptuelle
<b>Histogramme</b>	0.0145	0.0179 (+23.45%)
<b>SIFT</b>	0.0197	<b>0.0294 (+49.24%)</b>
<b>Fusion</b>	0.0276	0.0385 (+39.50%)
<b>GoogleNet</b>	0.1802	0.1898 (+5.33%)

**Tableau 4. 11** : Résultats (MAP (gain %)) de notre approche et une comparaison avec les résultats du système de base pour les concepts paires.

Discussion

A travers le tableau 4.11 on remarque que notre approche améliore les résultats pour tous les descripteurs et pour la fusion tardive, atteignant un gain allant de +5% à +49%.

- Détection de bi-concepts : approche par fusion des détecteurs singuliers

	Système de base	Rétroaction conceptuelle
<b>Histogramme</b>	0.0203	<b>0.0210 (+3.45%)</b>
<b>SIFT</b>	0.0273	0.0345 (+26.37%)
<b>Fusion</b>	0.0371	0.0424 (+14.30%)
<b>GoogleNet</b>	0.2707	0.2375 (-12.26%)

**Tableau 4. 12** : Résultats (MAP (gain %) de notre approche et une comparaison avec les résultats du système de base pour les concepts paires.

Discussion

Le tableau 4.12 présente les résultats de notre approche et la comparaison avec le système initiale. Nous remarquons une amélioration pour tous les descripteurs à part GoogleNet. Pour ce dernier, la MAP est déjà très bonne pour le système de base.

- Détection de tri-concepts: approche par modèle

	Système de base	Rétroaction conceptuelle
<b>Histogramme</b>	0.0033	0.0053 (+60.60%)
<b>SIFT</b>	0.0056	0.0092 (+64.28%)
<b>Fusion</b>	0.0061	<b>0.0103 (+68.85%)</b>
<b>GoogleNet</b>	0.0567	0.0583 (+2.82%)

**Tableau 4. 13** : Résultats (MAP (gain %) de notre approche et une comparaison avec les résultats du système de base pour les concepts triples.

Discussion

On remarque que notre approche améliore mieux pour la fusion tardive, L’histogramme et le « SIFT ». Le gain est moins important pour « GoogleNet ».

- Détection de tri-concepts: approche fusion de détecteurs de concepts singuliers

	Systeme de base	Retroaction conceptuelle
<b>Histogramme</b>	0.0088	0.0080 (-9.09%)
<b>SIFT</b>	0.0103	<b>0.0123 (+19.42%)</b>
<b>Fusion</b>	0.0152	0.0144 (-5.26%)
<b>GoogleNet</b>	0.0797	0.0652 (-18.19%)

**Tableau 4. 14** : Résultats (MAP (gain %)) de notre approche et une comparaison avec les résultats du système de base pour les concepts triples.

#### Discussion

A travers le tableau 4.14 on remarque que notre approche a mieux amélioré la détection pour le descripteur «SIFT ». Par contre elle n'a pas obtenu des bons résultats pour les autres descripteurs.

## 6. Synthèse

D'après les résultats présentés ci-dessus nous concluons que :

- ✓ l'approche de facteur de boosting est mieux efficace lorsqu'il s'agit de la détection des concepts triples utilisant l'approche par modèle.
- ✓ l'approche de rétroaction conceptuelle est performante dans le cas de détection des concepts double par fusion en utilisant l'histogramme comme descripteur.

## 7. Conclusion

A travers ce chapitre, nous avons présenté deux approches d'utilisation du contexte pour la détection de multi-concepts dans les images fixes. La première consiste à exploiter les concepts père pour booster la détection d'un multi-concept file. La deuxième quant à elle, extrait un descripteur sémantique pour faire un nouveau apprentissage afin de modifier les résultats d'un apprentissage initial. Nous avons détaillé les différentes expérimentations des deux approches ainsi que les résultats obtenus. Les résultats montrent que les deux approches peuvent aider à améliorer la détection de bi-concept et tri-concept.

# **Chapitre 5**

## **Conception et développement du système**

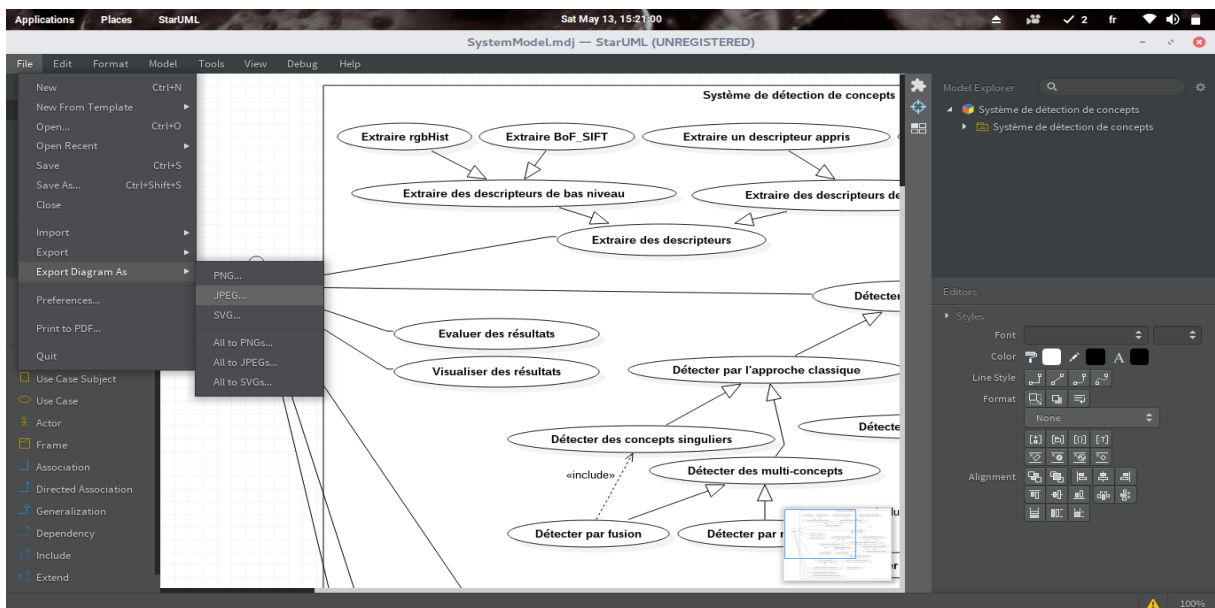
## 1. Introduction

Dans ce chapitre nous abordons les aspects de conception de notre système, en montrant les différents diagrammes de conception, qui décrit l'architecture et le fonctionnement de notre système de détection de concepts. Nous entamons ensuite la présentation de l'environnement de travail, en détaillant les différents outils utilisés. Nous allons présenter aussi, les différentes interfaces principales du système.

## 2. Conception

Nous avons modélisé les besoins de notre système de détection de concepts, en se basant sur le langage de modélisation UML « Unified Modeling Language ». Nous avons choisi pour cette modélisation deux diagrammes principaux, le diagramme de cas d'utilisation et le diagramme de classes, qui représentent les besoins fonctionnels ainsi que l'architecture du système.

Nous avons utilisé comme outils de modélisation UML, l'outil StarUML qui est un logiciel de modélisation, qui aide à tracer les différents diagrammes UML. Cet outil donne la possibilité d'exporter les diagrammes réalisés en images de format « png » ou « jpg » afin de les insérer au sein du document. La figure 5.1 montre l'interface principale de l'outil de modélisation StarUML version 2.8.0.



**Figure 5. 1** Interface principale de StarUML version 2.8.0

## 2.1. Diagramme de cas d'utilisation

Nous avons commencé la modélisation par le diagramme de cas d'utilisation, qui permet de recueillir, d'analyser et d'organiser les besoins, et de recenser les grandes fonctionnalités du système. La figure 5.2 montre le diagramme de cas d'utilisation de notre système de détection de concepts.

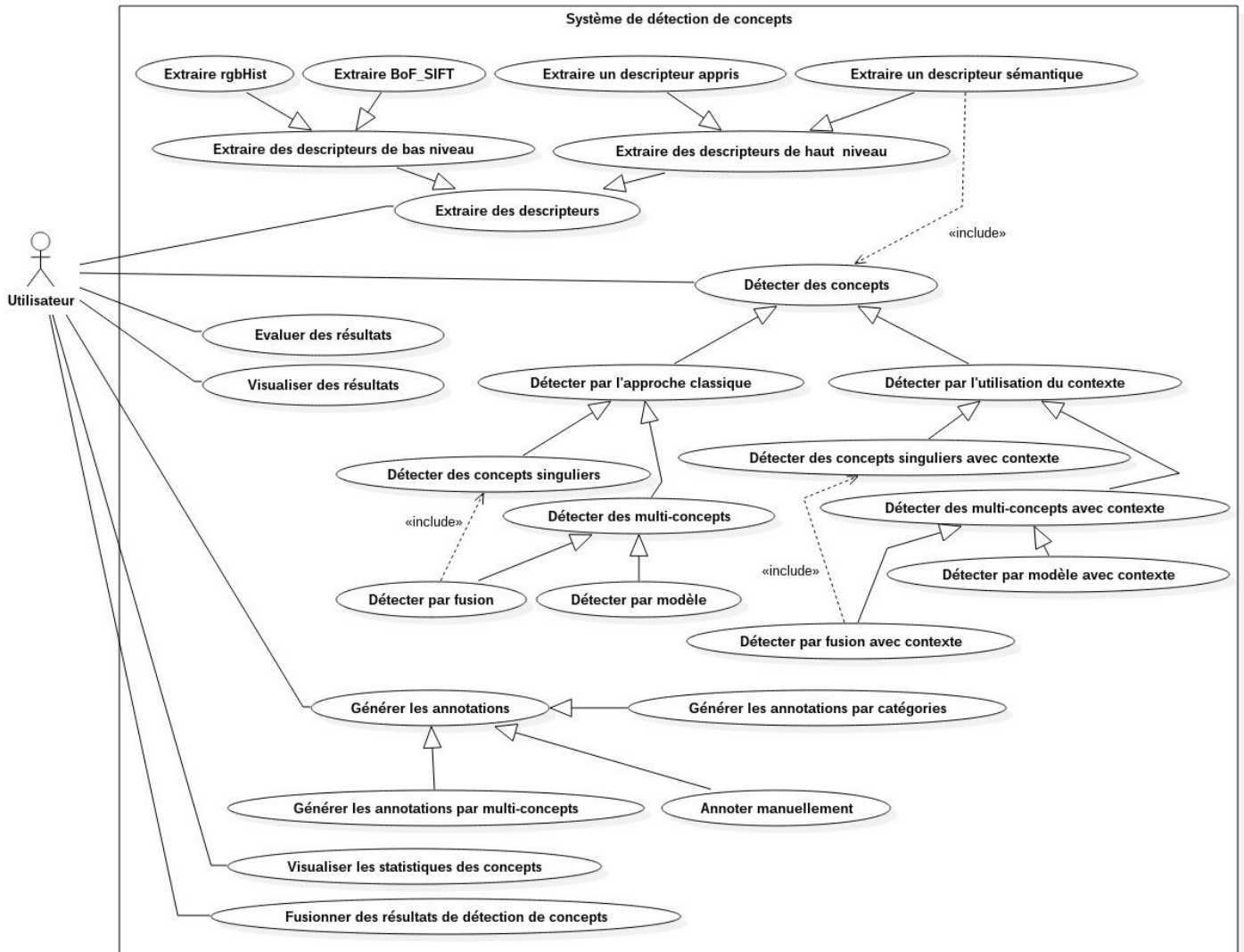


Figure 5.2 Diagramme de cas d'utilisation

## 2.2. Diagramme de classes

Nous avons utilisé le diagramme de classes pour son importance dans la modélisation orientée objet. Alors que le diagramme de cas d'utilisation montre un système du point de vue de l'interaction entre les utilisateurs et le système, le diagramme de classes en montre la structure interne du système. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation. Il s'agit d'une vue statique, car on ne tient pas compte du facteur de temps dans le comportement du système. La figure 5.3 montre le diagramme de classes de notre système de détection de concepts.

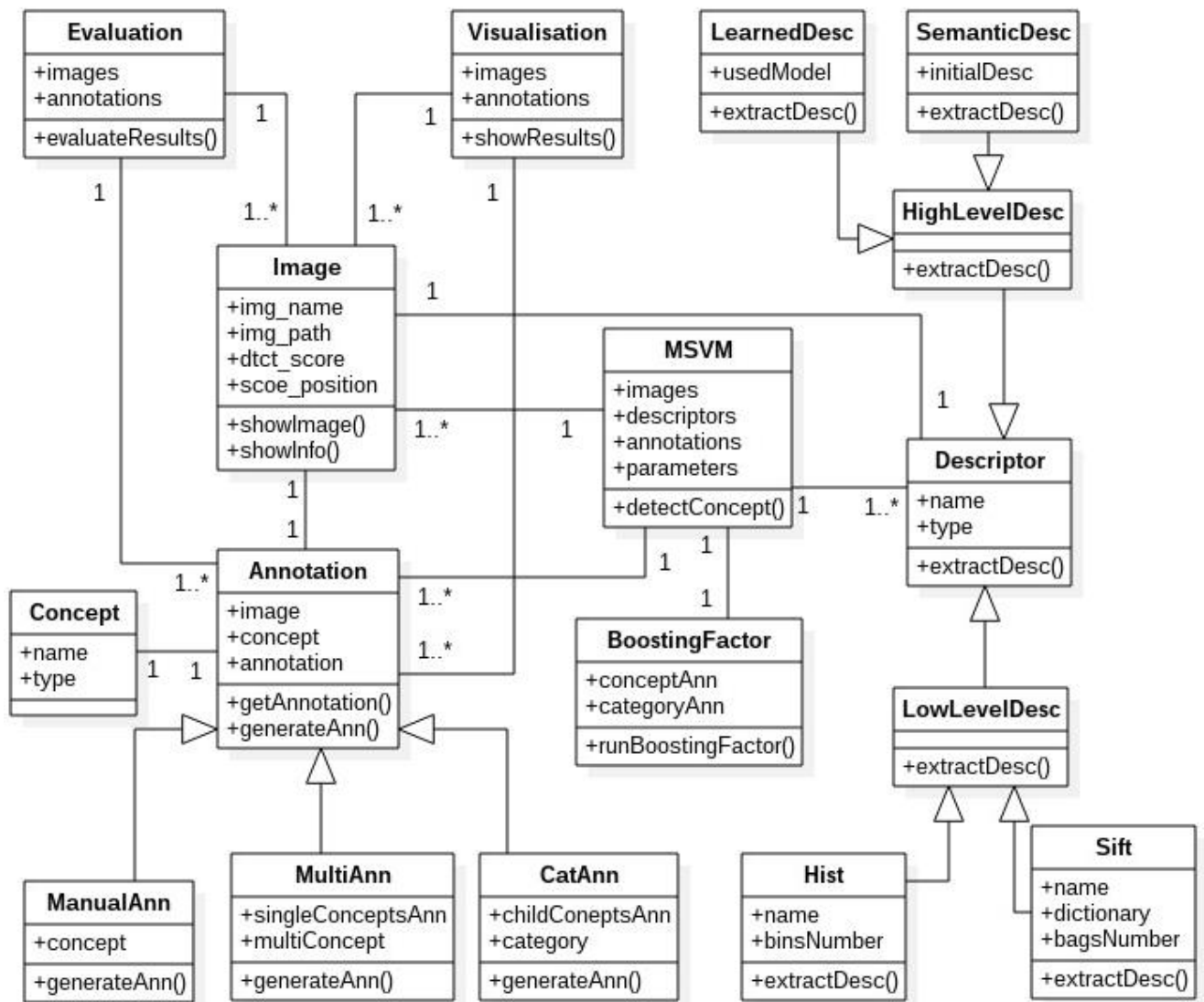


Figure 5.3 Diagramme de classes

### 3. Implémentation

#### 3.1. Environnement matériel et logiciel

La réalisation de notre système et les différentes expérimentations et évaluations de nos approches ont été faites dans l'environnement matériel et logiciel suivant :

- ✓ **CPU** : Intel ®Core™i5-4200M CPU @2.50GHz x 4.
- ✓ **RAM** : 4 GO.
- ✓ **Système d'exploitation** : Linux, distribution Ubuntu 16.04.2 LTS 64-bit.
- ✓ **Disque** : 500 Go.
- ✓ **Outil de modélisation** : StarUML version 2.8.0.
- ✓ **Outils de développement** :

NetBeans IDE 8.2 et java : on a utilisé ces deux langages pour le développement l'interface graphique.

Sublime Text : on utilisé cet éditeur de texte pour écrire nos code.



C, C++, , opencv : on a utilities ces langages pour le traitement d'image.

Python et coffe: on a utilize cues language pour la generation des descripteur appris.

### 3.2. Les interfaces graphiques principales

#### 3.2.1. Ecran de démarrage

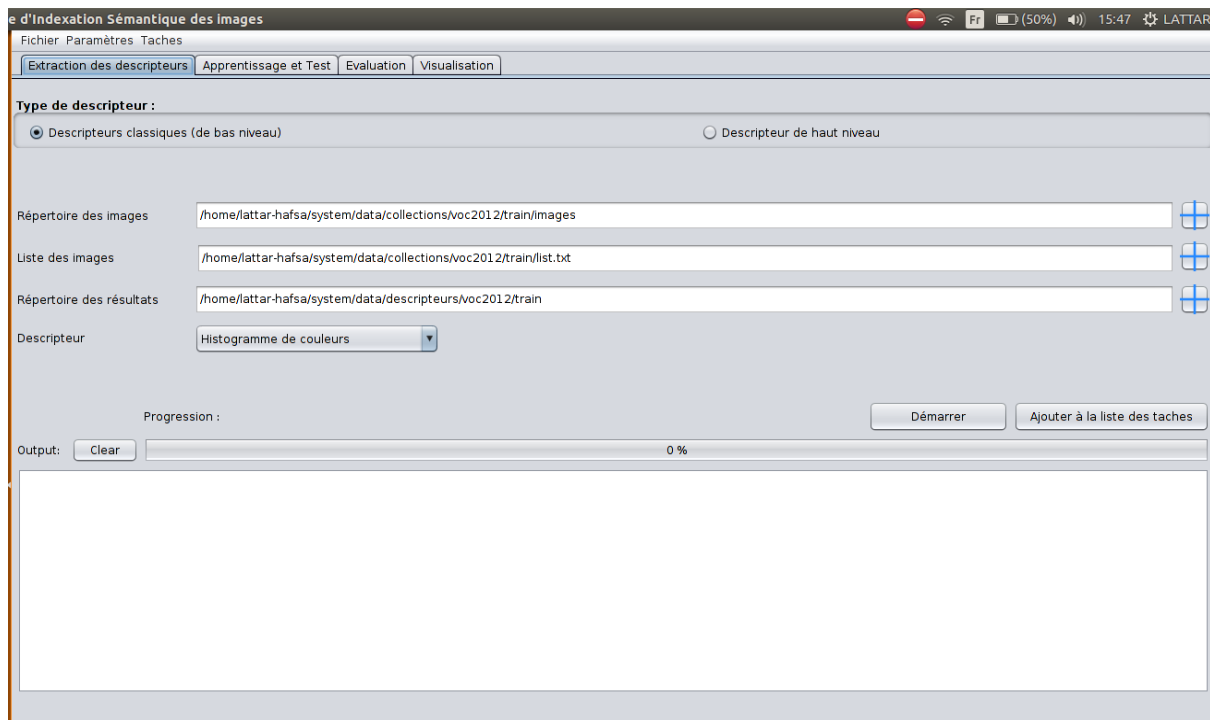
Au démarrage de notre système, une fenêtre (Figure 5.4) s'affiche qui représente l'interface d'accueil du système.



**Figure 5. 4:** écran de démarrage.

#### 3.2.2. Interface principale

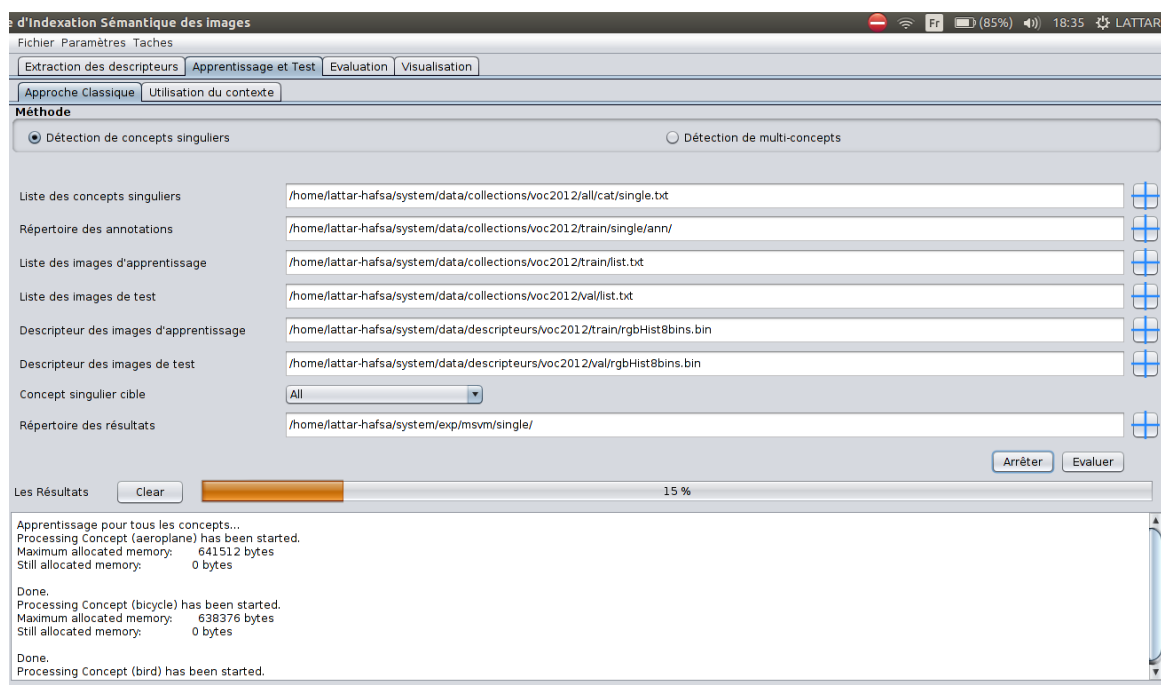
La figure ci-dessous montre l'interface principale de notre système, son avantage se trouve dans la simplicité de son fonctionnement.



**Figure 5. 5:** Interface principale.

### 3.2.3. Interface de détection des concepts singuliers et multiple par l'approche classique

L'interface présentée dans la figure 5.6 permet de détecter les concepts singuliers, doubles et triples.



**Figure 5. 6:** interface de détection de concepts “ approche classique”.

### 3.2.4. Interface de détection des concepts singuliers et multiple en utilisant le contexte

L'interface présentée dans la figure 5.7 permet de détecter les concepts en utilisant le contexte.

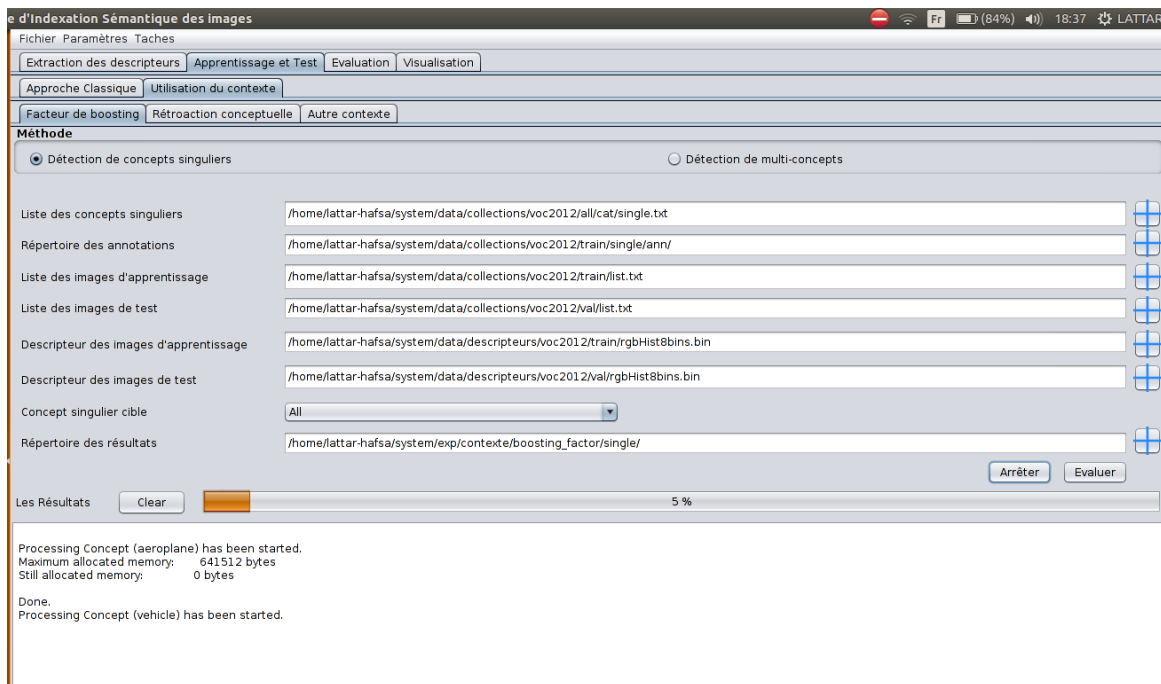


Figure 5. 7: interface de détection de concepts “ approche du contexte”.

### 3.2.5. Evaluation des résultats

A travers l'interface présentée dans la figure 5.8 on peut évaluer nos résultats de détection de concepts en terme de AP lorsque il s'agit d'un seul concept et en MAP dans le cas de plusieurs concepts.

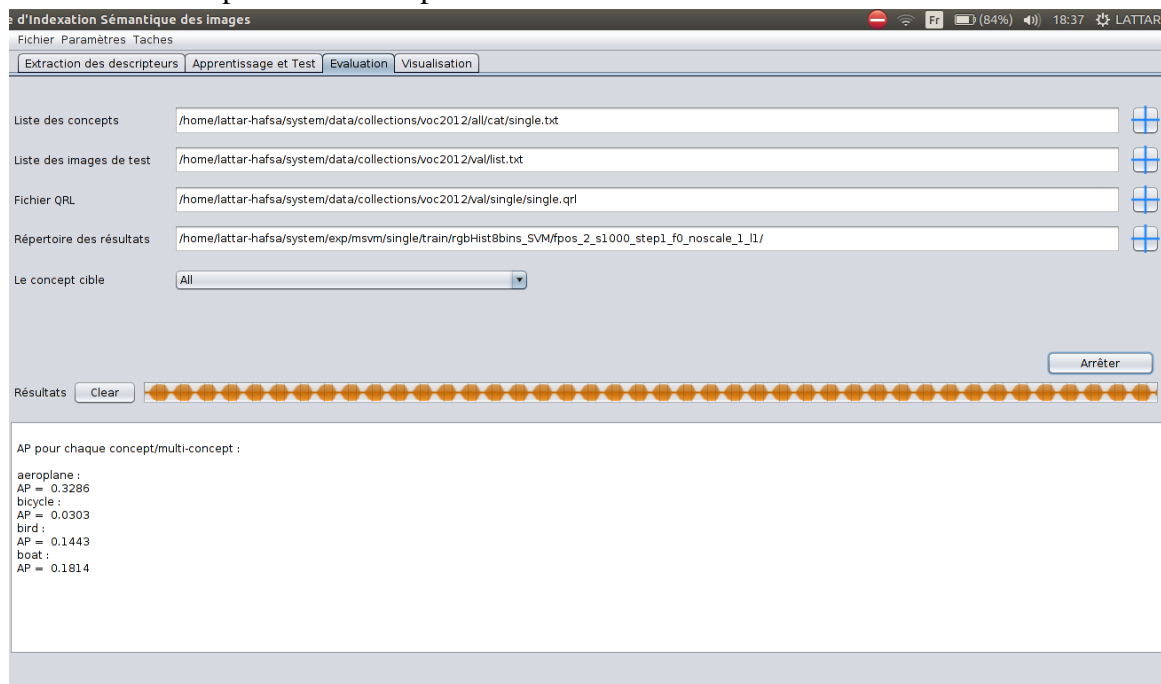


Figure 5. 8: Interface d'évaluation des résultats.

### 3.2.6. Interface de visualisation des résultats

Cette interface (Figure 5.9) permet de visualiser de près les résultats d'indexation des images.

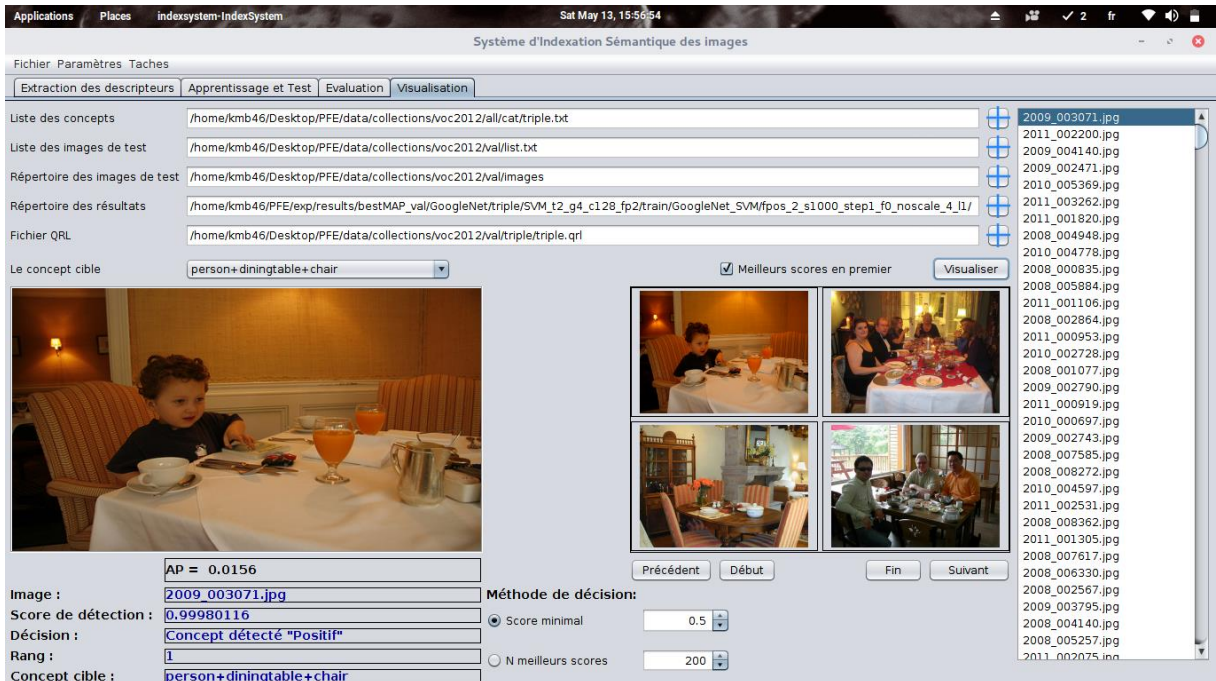


Figure 5. 9: Interface de visualisation des résultants.

### 3.2.7. Interface des paramètres d'apprentissage

Cette fenêtre de dialogue (Figure 5.10) permet de paramétrer les paramètres d'apprentissage.

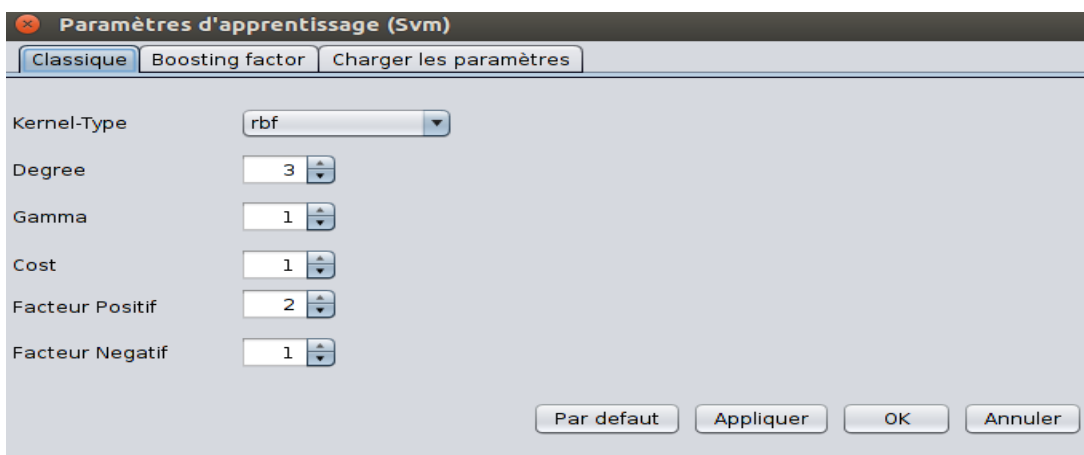
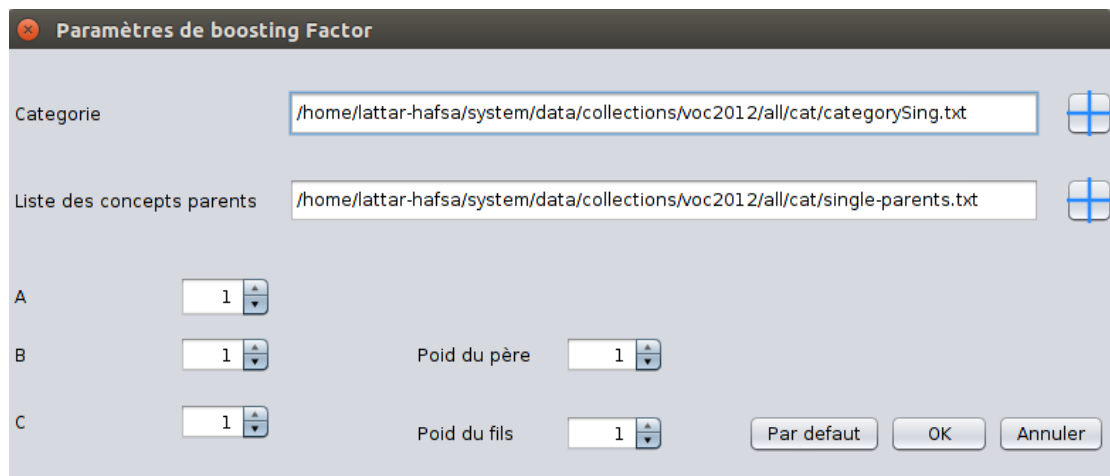


Figure 5. 10 : interface des paramétrages d'apprentissage.

### 3.2.8. Interface des paramètres de la méthode Facteur de Boosting

Cette fenêtre de dialogue présenter dans la figure 5.11 permet de paramétrer la méthode facteur de boosting.



**Figure 5. 11:** interface des paramètres de Facteur de boosting.

## 4. Conclusion

Dans ce chapitre nous avons présenté la conception de notre système à travers les diagrammes. Nous avons illustré également les différentes interfaces de notre système et l'environnement matériels et logiciel utilisé.

## Conclusion générale

Nous avons présenté dans ce rapport la description d'un système de détection de concepts dans les images fixes en détaillant ses différentes étapes ainsi que le processus d'évaluation. De plus, nous avons abordé la problématique de détection simultanée d'un ensemble de concepts (multi-concept).

Certains travaux de l'état de l'art ont montré l'utilité du contexte pour la détection des concepts singuliers. Nous avons présenté certains d'entre eux qui utilisent le contexte sémantique. Nous nous sommes focalisé sur ce contexte qui nous a paru moins complexe à mettre en œuvre, plus intéressant en termes du rapport efficacité/complexité et le mieux qualifié pour générer des systèmes génériques pouvant être appliqués dans diverses circonstances expérimentales pour la détection de n'importe quel multi-concept sans devoir apporter des modifications conséquentes.

Inspirés par les bons résultats de l'utilisation du contexte sémantique pour détecter des concepts singuliers, nous nous intéressons dans ce travail à l'utilisation du contexte pour la détection de multi-concepts dans les images fixes.

Nous avons proposé deux approches. La première exploite la relation père-fils entre les concepts multiples pour booster la détection des concepts multiples fils. La deuxième extrait un descripteur sémantique pour réaliser un nouveau apprentissage dans le but de booster la performance d'un apprentissage initial réalisé avec un autre concept, généralement de bas niveau.

Nous avons testé et évalué nos approches d'utilisation du contexte pour la détection de multi-concepts dans les images sur le corpus pascal voc 2012.

Les résultats d'évaluation montrent que nos approches peuvent améliorer les résultats de détection de concepts multiples atteignant des performances comparables, ou même meilleures que ceux de l'état de l'art.

## Bibliographie

- [1] Abdelkader Hamadi, *Utilisation du contexte pour l'indexation sémantique des images et vidéos*. Grenoble, France, 2014.
- [2] Stéphane Ayache, *Indexation de documents vidéos par concepts et par fusion de caractéristiques audio, image et texte*. Grenoble, 2007.
- [3] Abdelkader Hamadi, Philippe Mulhem, and Georges Quénot, *A comparative study for multiple visual concepts detection in images and videos*. New York , 2015.
- [4] Xirong Li, C. G M Snoek, M. Worring, and A.W.M and. Smeulders, *Harvesting social images for bi-concept search.:* Multimedia, IEEE Transactions on, 14(4) :1091–1104, 2012.
- [5] A. Hamadi, G. Quenot, and and P. Mulhem, *Two-layers re-ranking approach based on contextual information for visual concepts detection in videos.:* In Content-Based Multimedia Indexing (CBMI), 10th International Workshop, 2012.
- [6] Y. Wu, B. L. Tseng, and and J. R. Smith, *Ontology-based multi-classification learning for video concept detection, volume 2.,* 2004.
- [7] Bahjat Safadi et al., *Semantic Indexing and Multimedii Event Detection*. In TREC Video Retrieval Evaluation workshop, Gaithersburg, MD USA, 2011.
- [8] S.Bedouhene, *Recherche d'images par le contenu*. Université Mouloud Mammeri , TIZI-OUZOU.
- [9] D. Lowe, *Distinctive image features from scale-invariant keypoints.:* International Journal of Computer Vision, 60(2) :91–110, 2004.
- [10] Bahjat Safadi and and Georges Quénot, *Evaluations of multi-learners approaches for concepts indexing in video documents*. In RIAO,Paris, France, 2010.
- [11] DerekHoiem,JamesHays,AlexeiA.Efros, and Martial Hebert SantoshKumarDivvala, "An empirical study of context in object detection. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)," June 2009.
- [12] Anind K. Dey, "Understanding and using context. Personal Ubiquitous Comput.," January 2001..
- [13] Terry Winograd, "Architectures for context," December 2001.
- [14] Carolina Galleguillos and Serge Belongie, "Context based object categorization : A critical survey. Computer Vision and Image Understanding (CVIU)," 2010.
- [15] Jianmin Li, and Bo Zhang Jinhui Yuan, "Exploiting spatial context constraintsforautomaticimageregionannotation," 2007.
- [16] C. Schmid, and J. Ponce S. Lazebnik, "Beyond bags of features : Spatial pyramid matching for recognizing natural scene categories. InComputer Vision and Pattern Recognition," 2006.
- [17] Xian-ShengHua,YongRui,JinhuiTang,TaoMei,and Hong-Jiang Zhang Guo-JunQi, "Correlative multi-label video annotation," 2007.