Filtrage de l'indexation textuelle d'une image au moyen du contenu visuel pour un moteur de recherche d'images sur le web

Sabrina Tollari

Laboratoire LSIS - Equipe INCOD
Université du Sud Toulon-Var, Bât. R, BP 20132
F-83957 La Garde cedex
tollari@univ-tln.fr

Jeune chercheur

RÉSUMÉ. *Dans cet article, nous décrivons une méthode de filtrage de l'indexation textuelle d'images qui traitent de sujets généralistes. Pour cela, nous construisons d'abord des classes visuelles pour chaque mot-clé du lexique au moyen de classifications ascendantes hiérarchiques des vecteurs visuels dans l'espace visuel. Puis, nous testons la validité de l'association mot-clé/classes visuelles à l'aide d'une base de test et nous mesurons la performance de la classification obtenue à l'aide du score « normalized score ». Enfin, nous utilisons les classes visuelles pour filtrer l'indexation textuelle et obtenir une amélioration de l'indexation allant jusqu'à 40% suivant la consistance visuelle du mot.*

ABSTRACT. *In this article, we describe a method for filtering the textual indexing of images which cover general subjects. To this purpose, we build first visual clusters for each keyword of the lexicon by means of an ascending hierarchical clustering of visual vector. Then, we test the validity of the association keywords/visual clusters using a test base and we measure the performance of the classification with the "normalized score". Lastly, we use visual clusters to filter the textual indexing and to obtain an improvement of the indexing going up to 40% depending of the visual consistency of the word.*

MOTS-CLÉS : *Filtrage visuel, indexation textuelle, recherche d'images, classification*

KEYWORDS: *Visual filter, textual indexation, images retrieval, clustering, classification*

1. Introduction

La recherche d'images sur de grandes masses de données généralistes comme celles que l'on peut trouver sur le web (Google permet la recherche sur 880 millions d'images, Lycos Image Gallery sur 18 millions) nécessite des outils adaptés pour, d'une part, extraire efficacement des descripteurs significatifs, et d'autre part, retrouver les images pertinentes. Les systèmes actuels permettent des recherches par mots-clés sur les textes associés aux images comme pour les systèmes de recherche d'informations textuelles, ou bien par une image requête ressemblant aux images que l'on recherche. Certains travaux proposent de rechercher des images par construction d'une image requête à partir d'une composition de régions d'images ou d'imagettes [FAU 03] ce qui permet à l'utilisateur, dans une certaine mesure, de retrouver des images visuellement similaires à ce qu'il désire, mais ne lui permet toujours pas de rechercher des images contenant les symboles qu'il souhaite. De plus, dans la pratique les utilisateurs préfèrent poser des requêtes avec des mots-clés qui correspondent à des objets, des concepts, des caractéristiques visuelles... qui se trouvent dans le contenu de l'image, et non pas seulement dans le texte associé à l'image. Actuellement, peu de systèmes permettent d'indexer les images à partir de leur contenu et du texte qui les accompagne.

L'indexation textuelle des images sur le web peut s'effectuer à partir des mots présents dans le titre de la page ou des mots les plus fréquents ou pertinents de cette page. Cependant, il faut bien reconnaître que toutes les images présentes sur une même page web ne devraient pas être indexées avec les mêmes mots. Beaucoup de moteurs de recherche utilisent aussi l'URL et le nom de l'image pour l'indexation textuelle, mais la plupart des images ne sont pas nommées efficacement, et bien souvent par des noms génériques comme img001.jpg qui ne contient pas de sens. D'autres techniques considèrent les mots associés à l'attribut ALT de la balise IMG d'une image ou bien le texte proche de l'image, ou bien une fusion de toutes ces informations [Alm 04]. Mais, dans la pratique, peu d'images sur le web sont indexées de cette façon car cela nécessite une indexation manuelle d'images que l'on sait être très coûteuse en temps. De plus, le texte proche de l'image n'est pas forcement celui que l'on associerait à l'image. C'est pourquoi il serait intéressant de pouvoir utiliser le contenu de l'image pour améliorer son indexation textuelle, et donc de permettre des résultats de recherche plus cohérents avec le contenu de l'image.

Pour combiner les informations textuelles et visuelles, différents modèles ont déjà été proposés : une extension multi-modal du modèle LDA (Latent Dirichlet Allocation) [BAR 03], les modèles LSA (Latent Semantic Analysis) et PLSA (Probabilistic LSA) [MON 03], des modèles de champs de Markov cachés comme les 2D MHMMs (two-dimensionnal multiresolution hidden Markov models) [LI 03] ou bien encore des modèles de traduction de langues [JEO 03]. Cependant, associer des mots à des images uniquement à partir du contenu visuel de l'image (auto-annotation par le contenu) reste toujours un problème très difficile. Dans le cas des bases d'images qui sont entourées par du texte, nous disposons déjà d'un ensemble de mots qui sont souvent pertinents pour l'image, mais il faudrait vérifier la pertinence de chacun de ces mots

par rapport au contenu de l'image. L'originalité de ce travail est donc d'étudier la possibilité d'affiner l'indexation textuelle d'une image en exploitant son contenu visuel.

Dans la seconde partie, nous expliquons notre problématique. Dans la troisième partie, nous donnons une méthode pour construire des associations entre des mots et des traits visuels. Dans la quatrième partie, nous montrons comment évaluer la qualité de l'association mots/traits visuels. Dans la cinquième partie, nous donnons les résultats expérimentaux. Enfin, dans la dernière partie, nous en faisons une application.

2. Positionnement du problème

Décrire une image est un problème subjectif. Cependant, une image peut être décrite par un petit nombre de concepts. Au niveau textuel, chacun de ces concepts peut-être exprimé par un ou plusieurs mot-clés. Une image peut donc être indexée textuellement par un ensemble de mot-clés. Au niveau visuel, on peut utiliser des méthodes de segmentation d'images pour séparer une image en différentes parties, chacune correspondant à un concept. Dans la pratique, il est difficile de trouver automatiquement une correspondance entre une image et des concepts, ou entre une région de l'image et un concept.

Nous partons du principe que pour chaque image d, nous disposons d'un ensemble de mots de référence $W_{ref}(d)$ qui annotent cette image, ainsi que d'un ensemble de segments visuels. Chaque segment (appelé aussi région ou « blob ») b_j d'une image d est représenté par un vecteur $\vec{V}(b_j)$ de l'espace visuel multidimensionnel. Il est actuellement difficile d'associer des mots directement à des blobs. Donc nous supposerons que chaque blob peut-être associé avec l'ensemble des mots de l'image auquel il appartient : si $b_j \in d$ alors $W_{Ref}(b_j) = W_{Ref}(d)$.

Comme nous souhaitons appliquer ces méthodes aux images du web, il faut que les traits visuels soient rapidement calculables, adaptés à tout style d'images (paysages, personnes, objets, textures, intérieur, extérieur...). La figure 1 montre un exemple de segmentation en région.

Pour vérifier la pertinence visuelle d'un mot par rapport à une image, nous allons tout d'abord construire des classes visuelles qui nous serviront de classes de référence visuelle sur une base d'apprentissage. Puis nous vérifierons la qualité des classes visuelles sur une base de test.

3. Construction d'une classification de référence

Dans cette première étape, nous allons associer à chaque mot w_i du lexique un ensemble de classes visuelles $\mathcal{C}(w_i) = \{C_1(w_i), C_2(w_i), \cdots, C_K(w_i)\}$. Nous définissons une classe visuelle $C_k(w_i)$ comme étant un hyperrectangle dans l'espace multidimensionnel visuel.

Figure 1. *Exemple de segmentation d'une image. Chaque région de l'image est appelée un blob. Chaque image de la base est annotée manuellement par des mots appelés mots de référence. Par exemple, les mots de référence de cette image extraite de [BAR 03] sont : « sun », « sky » et « water ».*

Pour construire les classes visuelles d'un mot, nous allons chercher des regroupements d'individus (dans notre cas des blobs) de la base d'apprentissage dans l'espace multidimensionnel visuel. Nous avons choisi d'utiliser une Classification Ascendante Hiérarchique (CAH) [LAN 67] (construction non-supervisée de clusters) avec comme critère d'agrégation le plus proche voisin. Le principe de cette CAH est de regrouper les blobs ayant une distance faible dans l'espace multidimensionnel. Cette méthode donne en général en fonction du critère d'arrêt une ou plusieurs classes contenant un très grand nombre d'individus et beaucoup de classes contenant très peu d'individus. Nous ne garderons que les classes qui ont un nombre significatif de blobs.

Pour chaque mot w_i, nous construisons un sous-ensemble $\mathcal{A}(w_i)$ d'apprentissage composé des images d de l'ensemble d'apprentissage $\mathcal{A}$ possédant le mot w_i :

$$\mathcal{A}(w_i) = \{d | w_i \in W_{Ref}(d) \; et \; d \in \mathcal{A}\}. \qquad [1]$$

La figure 2 montre un exemple d'un sous-ensemble d'images de la base d'apprentissage indéxées par le même mot. Sur cet ensemble $\mathcal{A}(w_i)$, nous réalisons une CAH. Nous déterminons alors la valeur d'arrêt de la CAH en choisissant celle qui donne le meilleur score (le calcul du score est expliqué section 4.2). Nous gardons alors seulement les classes qui contiennent un nombre significatif de blobs. Nous associons ainsi des classes visuelles au mot w_i. Chaque classe $C_k(w_i)$ est représentée par un couple de vecteurs de même dimension :

$$(\bar{C_k}(w_i), \; \vec{\sigma}(C_k(w_i))) \qquad [2]$$

où $\bar{C_k}(w_i)$ est le vecteur centroïde de la classe visuelle dans l'espace multidimensionnel et $\vec{\sigma}(C_k(w_i))$ est le vecteur des écarts-types de la classe pour chaque dimension de l'espace.

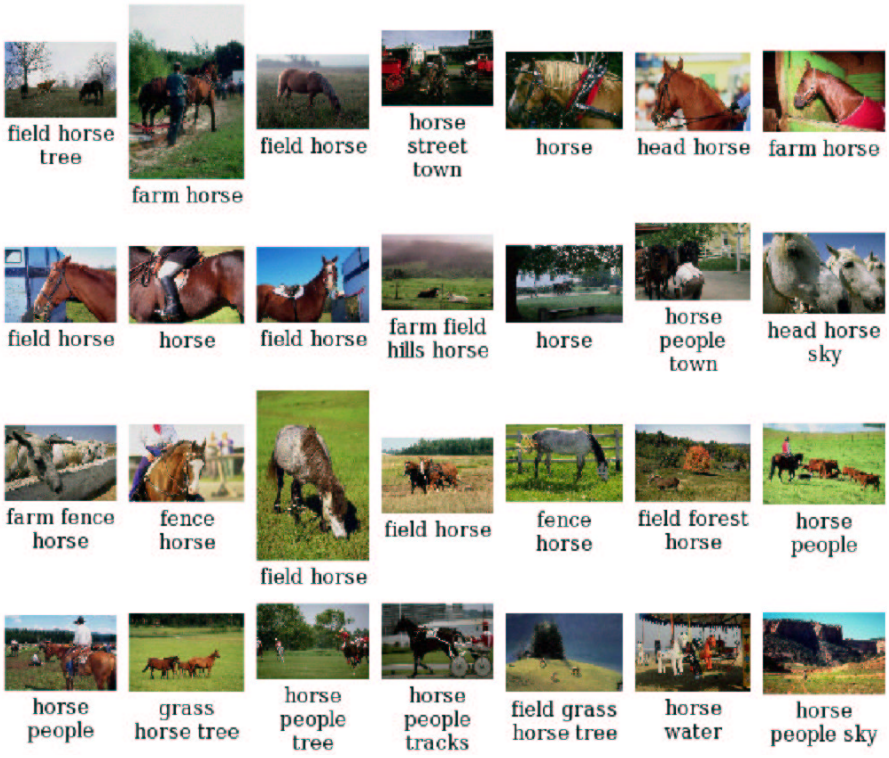

Figure 2. *Quelques unes des 173 images de la base d'apprentissage étiquetées par le mot-clé « horse ». Les blobs de ces images sont utilisées pour construire les classes visuelles associées à ce mot et présentées figure 3.*

Remarquons que les classes visuelles d'un mot sont disjointes, car aucun blob ne peut appartenir à deux classes à la fois :

$$\forall k \neq k' \; C_k(w_i) \cap C_{k'}(w_i) = \emptyset \qquad [3]$$

ce que l'on peut interpréter comme le fait qu'un mot peut avoir plusieurs sens visuels. Par exemple, le soleil peut être jaune, mais il est aussi souvent rouge lors des couchers de soleil. Un mot peut On remarque aussi que deux mots différents peuvent avoir des classes visuelles non disjointes :

$$\exists \; k, k' \; et \; w_i, w_i' \; C_k(w_i) \cap C_{k'}(w_i') \neq \emptyset. \qquad [4]$$

En effet, deux mots peuvent avoir des visuels très proches. Par exemple, les mots : humain, homme, femme, enfant ont, sur les bases d'images généralistes, des sens visuels très proches parce qu'on retrouve dans les images le même type de texture représentant la peau.

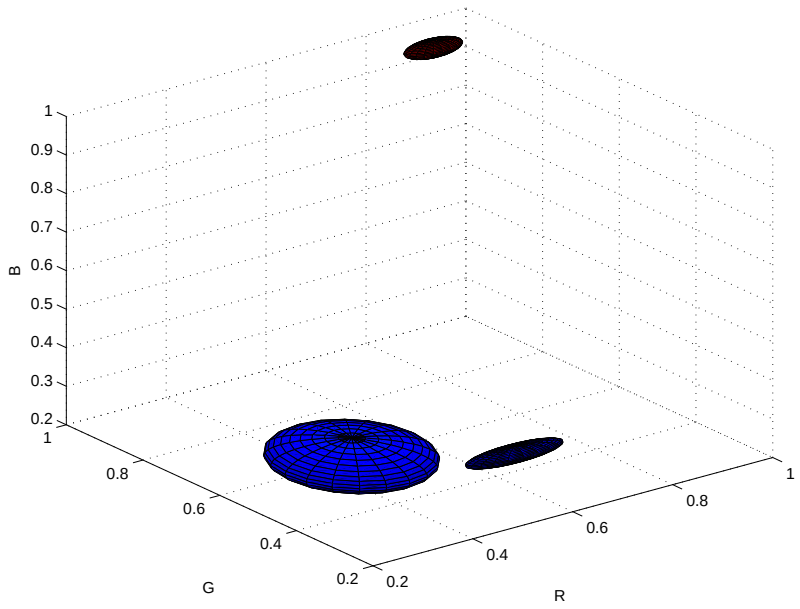

Figure 3. *Représentation des trois classes visuelles du mot « horse » dans trois (RGB) des 40 dimensions de l'espace visuel. Les classes visuelles sont représentées ici par des ellipsoïdes, mais ce sont en fait des hyperrectangles. La plus grande classe correspond à la couleur vert-marron, la seconde à la couleur marron, la plus petite en haut à du gris bleuté très clair. Ces couleurs sont bien celles que l'on s'attend à retrouver associées au mot « horse » (voir figure 2). On remarque que comme nos classes sont construites par classification sur des mots associés à des images, et non pas à des blobs, le contexte est aussi pris en compte.*

4. Evaluation des classes visuelles des mots

Nous venons de décrire une méthode permettant d'associer des classes visuelles à des mots, nous allons maintenant évaluer la qualité de cette association afin de déterminer pour chaque mot la meilleure valeur d'arrêt de la CAH. Pour cela, nous utilisons des images d'un ensemble de test $\mathcal{T}$. Pour chaque mot, nous classons d'abord les blobs des images de test, puis les images de test, dans l'espace visuel. Enfin, nous calculons le score.

4.1. *Association de mots à une image*

Un blob b_j d'une image de test appartient par définition à une classe visuelle $C_k(w_i)$ si le vecteur visuel de b_j est dans l'hyperrectangle de la classe $C_k(w_i)$, autrement dit si pour chaque dimension visuelle p, la valeur du vecteur visuel du blob b_j

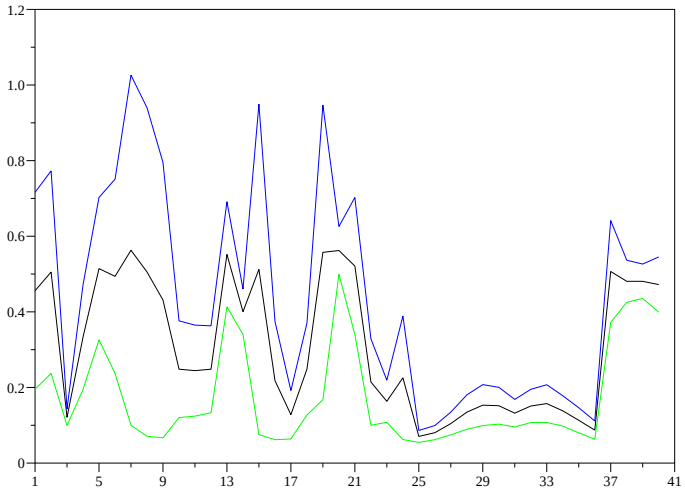

Figure 4. *Exemple de classe visuelle pour le mot « woman » représentée par le vecteur centroïde (courbe du milieu). Les deux autres courbes sont respectivement la somme du vecteur centroïde plus le vecteur d'écarts-types mutiplié par le paramètre D=3.5 et le vecteur centroïde moins le vecteur d'écarts-types mutiplié par le paramètre D=3.5. En abscisse, les 40 dimensions visuelles (6 de formes, 18 de couleurs, 16 de textures). En ordonnée, la valeur de chaque trait comprise entre 0 et 1.*

pour la dimension p est à une distance de la valeur du centroïde de la classe $C_k(w_i)$ pour la dimension p inférieure à la valeur de l'écart-type pour cette dimension multipliée par une constante :

$$b_j \in C_k(w_i) \; ssi \; \forall p \; |\bar{C_{k,p}}(w_i) - \vec{V_p}(b_j)| \leq D \times \vec{\sigma_p}(C_k(w_i)) \qquad [5]$$

où D est une constante déterminée de manière empirique. Si l'on suppose que la distribution des blobs des classes visuelles dans la phase d'apprentissage suit une loi normale alors pour $D = 2$ nous savons que 95% des individus sont dans la classe ([DUD 00] page 33). Dans notre cas, nous n'avons pas forcement des lois normales, nous supposerons cependant que $0 < D \leq 4$.

Pour chaque blob b_j de test, nous associons le mot w_i à b_j si le blob b_j appartient à l'une des classes visuelles de ce mot :

$$w_i \in W_{Sys}(b_j) \; ssi \; \exists k \; tq \; b_j \in C_k(w_i) \qquad [6]$$

où $W_{Sys}(b_j)$ est l'ensemble des mots associés par le système au blob b_j. Remarquons que comme les classes visuelles d'un mot sont disjointes, un blob appartient au plus a

une classe. Donc s'il existe k tel que $b_j \in C_k(w_i)$ alors il est unique. Donc $|\{k|b_j \in C_k(w_i)\}| = 1$ si le mot w_i est associé à b_j, 0 sinon.

Nous supposons qu'un mot w_i est associé à une image d de test si ce mot est associé avec au moins B blobs de cette image :

$$w_i \in W_{Sys}(d) \ ssi \ \sum_{b_j \in d} |\{k|b_j \in C_k(w_i)\}| \geq B \qquad [7]$$

où $W_{Sys}(d)$ est l'ensemble des mots associés par le système à l'image d et B une constante inférieure ou égale au nombre minimal de blobs d'une image. Cette constante est dépendante du nombre de blobs dans une image. En effet, plus il y a de blobs dans une image et plus la constante B sera grande, car un mot correspondra alors à plusieurs blobs de l'image.

4.2. *Evaluation du score*

Chaque image de la base de test possède initialement un ensemble de mots de référence $W_{Ref}(d)$. Nous pouvons donc calculer les taux de sensibilité et de spécificité. Nous utilisons également le score « Normalized Score » (noté NS par la suite) employé dans [BAR 03, MON 03].

La sensibilité (aussi appelée rappel) est le nombre de documents pertinents retrouvés parmi le nombre de documents pertinents. Nous l'utilisons pour mesurer le nombre d'images de test auxquelles on a associé w_i qui étaient indexées initialement par w_i.

La spécificité est le nombre de documents non-pertinents non-retrouvés parmi le nombre de documents non-pertinents. Dans la pratique, nous utilisons le complément à 1 de la spécificité, car nous voulons connaître le nombre de documents non-pertinents retrouvés parmi le nombre de documents non-pertinents. Nous l'utilisons pour mesurer le nombre d'images auxquelles on a associé w_i qui n'étaient pas indexées initialement par w_i.

Le score NS s'écrit finalement comme étant la somme de la sensibilité et de la spécificité moins 1 :

$$NS = sensibilité - (1 - spécificité) \qquad [8]$$

ou bien encore

$$NS = \frac{right}{n} - \frac{wrong}{N - n} \qquad [9]$$

où $right$ est le nombre d'images qui avaient w_i pour mot de référence et qui vérifiaient $w_i \in d$, $wrong$ est le nombre d'images qui n'étaient pas indexées par w_i mais que le système a accepté, N est le nombre total d'images dans la base de test et n est le nombre d'images ayant le mot w_i. On remarque que $-1 \leq NS \leq 1$. Le score vaut 1 quand on trouve les n mots de références, et aucun des autre mots, -1 quand on ne trouve que les mots qui ne sont pas de référence, 0 quand on trouve tous les mots.

5. Expérimentations

5.1. *Description du corpus*

La base d'images utilisée est un sous-ensemble de COREL [WAN 04, MUL 02, WAN 01]. Elle est composée de 10000 images. Chaque image possède de 1 à 5 mots-clés choisis manuellement parmi un ensemble de 250 mots environ. En moyenne, il y a 3.6 mots-clés par image.

Les images ont été prétraitées par des chercheurs du Computer Vision Group de l'université de California (Berkeley) et du Computing Science Department de l'université d'Arizona [BAR 03]. Chaque image a été segmentée en utilisant la méthode « normalized cuts » [SHI 00] et les 10 plus grands blobs ainsi créés ont été conservés. En moyenne sur notre corpus, il y a 9.5 blobs par image. La figure 1 donne un exemple de segmentation par « normalized cuts ».

Les auteurs ont choisi d'extraire des caractéristiques visuelles générales calculables sur tout type de segments. Chaque blob est donc décrit par un vecteur de 40 dimensions, composées de :

– 6 dimensions de formes (aire, périmètre sur aire, convexité, moment d'inertie, position en x et y du barycentre du segment),

– 18 dimensions de couleurs (RGB, RGS, LAB et leurs std),

– 16 dimensions de texture.

Nous avons ensuite normalisé les vecteurs visuels par estimation MLE de distributions Gamma. Finalement, chaque blob est représenté par un vecteur de 40 dimensions dont chaque composante est dans $[0, 1]$. Nous réduisons notre lexique aux mots ayant plus de 20 occurrences dans la base d'apprentissage : il est donc finalement composé d'un ensemble de 161 mots-clés.

5.2. *Réalisation*

Nous séparons aléatoirement le corpus en une base de test de 2922 images et une base d'apprentissage de 6998 images. Pour chacun des mots qui apparaissent au moins dans 20 images de base d'apprentissage, nous réalisons une CAH. Nous calculons les scores au fur et à mesure du déroulement de la CAH à pas d'arrêt régulier et à chaque fois pour différentes valeurs de D ($0 < D \leq 4$). Nous stockons à chaque fois les valeurs des scores obtenus ainsi que diverses informations : les vecteurs centroïdes des classes visuelles, les vecteurs d'écarts-types, les valeurs de D correspondant aux scores, la valeur δ de la distance pour la dernière fusion de deux classes visuelles lors de la CAH. Les figures 4 et 3 montrent des exemples de classes visuelles représentée par le vecteur centroïde et les vecteurs d'écarts-types. On remarque que les valeurs d'écarts-types sont différentes pour chaque dimension visuelle. Plus l'écart-type est grand, et moins le trait visuel correspondant est discriminant pour ce mot.

5.3. *Résultats pour chaque mot*

Le tableau 1 résume les meilleurs scores NS obtenus pour chacun des mots les plus fréquents. On remarque que certains mots que l'on aurait pensés visuellement similaires n'obtiennent pas le même score. Par exemple, les mots « people » (NS=0.05) et « woman » (NS=0.22). On peut supposer qu'une image est indexée manuellement par le mot « people » quand une personne est peu visible sur l'image, tandis qu'elle sera indexée par « woman » quand la personne est bien visible. Pour les mots « pattern » et « texture », les meilleurs scores NS sont obtenus avec une sensibilité faible ce qui indique que le mot est reconnu sur peu d'images, alors que ce sont des mots pour lesquels la texture devrait être discriminante. On peut proposer deux hypothèses pour expliquer ces mauvais scores soit les indices visuels de texture sont descriptifs mais les autres traits visuels comme la couleur et la forme ne sont pas adaptés ce qui provoque du bruit lors de la construction des classes visuelles ou soit les descripteurs de texture ne sont pas adaptés pour réaliser des CAH. Il peut paraître surprenant que le système donne un score NS acceptable pour certains mots comme « flower » (NS=0.16) et « fish » (NS=0.19) alors que ce sont des mots qui peuvent se présenter sous des traits visuels différents. Par exemple, il y a des fleurs de couleurs très différentes. Mais il se peut que la classe visuelle associée à « flower » correspondent aux feuilles vertes associées généralement à toutes fleurs. De même, pour « fish », il se peut que la classe associée correspondent à l'eau associée généralement avec un poisson. On voit sur ces exemples que le système peu prendre aussi en compte le contexte d'un mot si ce contexte est très récurrent.

Finalement, nous remarquons qu'en fonction du score NS obtenu, nous pouvons conclure si un mot peut être facilement détecté par le système dans une image (on dira qu'il a une forte consistance visuelle), mais si le score est faible cela ne veut pas forcement dire que le mot ne peut pas être détecté dans une image, car d'autres traits visuels ou une autre façon de construire les classes visuelles de référence permettraient peut-être d'obtenir un meilleur score.

6. Application au filtrage d'une indexation

Sur le web, les images sont associées avec un ensemble de mots-clés parmi lesquels certains sont pertinents et d'autres non. Nous souhaitons filtrer les mots en fonction de la pertinence visuelle des mots par rapport à une image. Malheureusement, nous ne disposons pas de bases d'images permettant de valider ce filtrage. De plus, il faudrait une validation de l'efficacité du filtrage par des utilisateurs. Nous proposons donc une autre méthode. Nous supposons que les mots de références associés à une image de la base de test sont les mots pertinents pour l'image, et que tous les autres mots du lexique représentent les mots non-pertinents de la page web. Nous utilisons alors les classes visuelles obtenues précédemment (dont les résultats sont décrits dans le tableau 1) et la base de test (normalement, nous devrions utiliser un autre sous-ensemble de la base d'images pour réaliser cette application) afin de filtrer l'ensemble des mots du lexique pour chacune des images. Cette méthode pourrait être de l'an-

Tableau 1. *Meilleur score NS pour chacun des 31 mots ayant au moins 150 images dans la base d'apprentissage. Les mots sont ordonnés du plus fort score NS au plus faible. NbI est le nombre d'images possédant le mot w_i parmi les 2922 images de test. La valeur δ est la distance maximale qui a permis à deux blobs de fusionner dans une même classe. D est la constante multiplicative de l'écart-type. K est le nombre de classes visuelles pour le mot w_i. Le score NS donne une échelle de consistance visuelle du mot.*

Données		Paramètres		Résultats			
Mot w_i	NbI	δ	D	K	Sensibilité	Spécificité	Score NS
cat	138	0.78	2.5	1	0.77	0.63	0.4
snow	174	0.68	3	1	0.69	0.69	0.38
grass	233	0.7	2.5	1	0.65	0.72	0.37
ground	60	0.71	2.5	2	0.57	0.79	0.36
field	122	0.83	2.5	1	0.73	0.62	0.35
cloud	128	0.64	3	1	0.55	0.75	0.31
mountain	191	0.6	3	1	0.61	0.69	0.3
ruins	82	0.89	3	1	0.9	0.37	0.27
horse	96	0.67	3.5	3	0.57	0.69	0.26
rock	181	0.75	2.5	1	0.65	0.6	0.25
sand	84	0.86	2.5	1	0.68	0.57	0.25
leaf	119	0.79	2.5	2	0.56	0.67	0.24
forest	77	0.82	2.5	1	0.74	0.49	0.23
woman	66	0.75	3.5	1	0.82	0.4	0.22
stone	112	0.69	3.5	1	0.79	0.42	0.21
plants	129	0.94	2	3	0.71	0.48	0.19
fish	100	0.83	3	1	0.85	0.34	0.19
boat	114	0.85	3	3	0.85	0.33	0.18
water	583	0.69	3	1	0.73	0.43	0.17
flower	222	0.89	2	1	0.66	0.5	0.16
bird	164	0.62	3.5	4	0.53	0.62	0.16
pattern	86	0.6	3.5	1	0.17	0.97	0.14
sky	532	0.55	3.5	1	0.45	0.68	0.13
street	88	0.93	2.5	1	0.8	0.34	0.13
building	242	0.81	3	4	0.86	0.27	0.13
tree	534	0.81	3	3	0.89	0.23	0.11
garden	80	0.95	2.5	1	0.86	0.25	0.11
texture	72	0.99	2	3	0.35	0.76	0.1
house	59	0.65	3.5	3	0.36	0.72	0.08
closeup	202	0.95	2.5	4	0.77	0.31	0.08
people	491	0.95	2.5	1	0.91	0.14	0.05

Tableau 2. *Moyenne des scores NS du filtrage de l'ensemble des mots du lexique pour chaque image de la base de test : pour la méthode BESTALLNS qui consiste à maximiser le score NS, et pour la méthode BESTSENSINS qui consiste à fixer une sensibilité supèrieure à 0.80 (afin de garder lors du filtrage le maximum de mots de référence), puis à maximiser le score NS.*

Méthode	Sensibilité	Spécificité	NS
BESTALLNS	0.70	0.59	0.29
BESTSENSINS	0.77	0.50	0.27

notation d'images à partir du contenu visuel, mais ce n'est pas notre objectif, nous cherchons juste un filtrage grossier qui permette d'éliminer les mots qu'il est visuellement impossible d'associer à une image donnée de la base de test.

Pour chaque image de test, nous calculons le score NS (décrit partie 4.2) en prenant $right$ comme étant le nombre de mots de référence de l'image qui ont été associés à l'image par le système, et $wrong$ le nombre de mots qui ne sont pas de référence que le système a quand même associé à l'image. Nous faisons ensuite la moyenne des scores NS obtenus pour toutes les images de test. Le tableau 2 donne les résultats obtenus pour la méthode BESTALLNS qui consiste à prendre comme précédemment les classes visuelles pour lesquelles on a obtenu le meilleur score NS. Nous réalisons une deuxième expérience de filtrage appelée BESTSENSINS où nous gardons les classes visuelles des mots pour lesquelles la sensibilité à un taux supérieur à 0.80, et qui maximise le score NS. Nous fixons la sensibilité supérieure à 0.80 afin de garder lors du filtrage le plus de mots de référence possible et nous maximisons quand même le score NS pour éviter que tous les mots du lexique soient acceptée. Nous obtenons ainsi des score NS légèrement moins bons pour chacun des mots qu'avec la méthode BESTSENSINS. Cependant, en moyenne, la perte de score NS total (tableau 2) entre les deux méthodes n'est que de 2 points alors que la sensibilité a augmenté de 7 points, ce qui signifie que l'on gardera 77% des mots de référence au lieu de 70%. Dans notre application au filtrage de l'indexation des mots d'une image, il est donc plus intéressant de prendre une sensibilité forte, puis de maximiser le score NS. La figure 5 montre des exemples de filtrages d'indexation avec la méthode BESTALLNS.

Nous pouvons de plus critiquer ces résultats en remarquant que dans les mots de référence d'une image plusieurs mots du lexique peuvent décrire des concepts proches et qu'il donc normal que le système visuel associe un très grand nombre de mots. Prenons par exemple, les mots « woman » et « people ». En général, quand une image est indexée par le mot « woman » alors elle pourrait l'être aussi par « people », l'utilisation d'un thésaurus hiérarchique adapté aux images permettraient de résoudre ce type de problème. De plus, souvent même les mots de références sont mal choisis. Par exemple, le mot « people » est ajouté même quand les personnages sont très petits sur l'image est donc non-détectables par le système. Mais le système a l'avantage d'agir comme un filtre pour éliminer les mots qui ne peuvent être sur l'image. Une autre

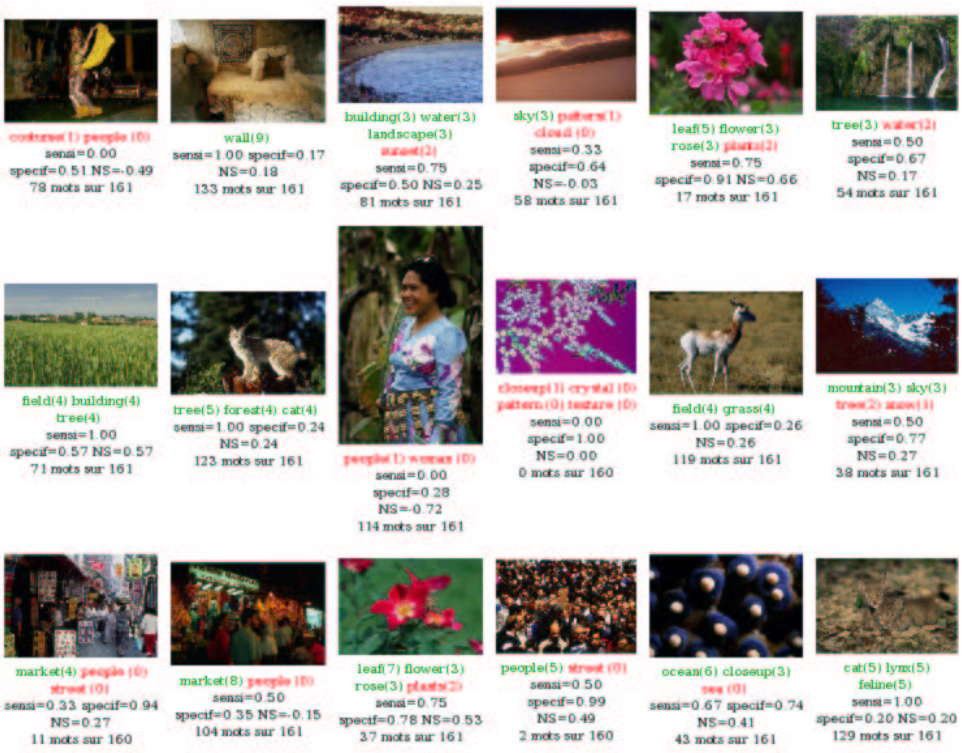

Figure 5. *Exemples de filtrage d'indexation sur les images de la base de test. Les 161 mots du lexique sont associés à chacune des images. Puis, le système filtre ces mots en regardant si les blobs de ces images sont dans les classes visuelles des mots. Ensuite, le système compte le nombre de blobs de l'image ainsi associés à chaque mot (valeur entre entre parenthèses après le mot). Tous les mots qui ont au moins B blobs sont associés à l'image. (pour cet exemple $B = 3$). Enfin, nous calculons les valeurs de sensibilité et de spécificité, le score NS et le nombre de mots associées à l'image parmi les 161 mots. Les mots qui ont une forte consistance visuelle comme « field » permettent un filtrage d'indexation pertinent. Les mots qui ont une faible consistance visuelle comme « people » ne permettent pas de filtrer efficacement les indexations.*

discussion sur la méthode est le choix du nombre de mots que l'on filtre. Nous avons choisi de filtrer pour tous les mots du lexique afin d'obtenir une bonne estimation du pourcentage de mots qui seront supprimés de l'indexation par le système. Dans la pratique, tous les mots du langage ne sont pas sur la même page web. Remarquons enfin que pour les mots qui ont un faible score NS, il est normal que le filtrage visuel ne permette pas d'améliorer l'indexation textuelle, puisque ces mots n'ont pas de consistance visuelle. Pour ces mots, nous ne pouvons pas filtrer l'indexation à partir du contenu visuel.

7. Conclusion

Dans cet article, nous avons présenté une méthode permettant d'associer des mots et des traits visuels à partir d'une base d'images indexées textuellement et visuellement au moyen de classes visuelles construites par une méthode de classification non-supervisée. La qualité de la méthode a été évaluée par un calcul de score. Une application de ce travail est le filtrage visuel de l'indexation textuelle d'une image. En effet, la recherche d'images sur le web n'est actuellement pas performante car les moteurs de recherche d'images ne prennent pas en compte le contenu visuel. Or si nous indexons mieux les images avec moins de mots, mais en gardant les plus pertinents, alors lors de la phase « recherche » les résultats devraient donner une meilleure précision. Nous avons montré que certains mots ont une consistance visuelle plus forte que d'autres. Le filtrage de l'indexation textuelle pour ces mots est donc plus efficace que pour d'autres.

Un des avantages de cette méthode est que l'extraction des caractéristiques visuelles de l'image n'est nécessaire que dans la phase « indexation », mais pas dans la phase « recherche » qui doit être très rapide. De plus, elle utilise les techniques classiquement utilisées en recherche d'informations pour le texte, techniques qui ont l'avantage d'être rapides, et pratiques pour l'utilisateur qui préfère chercher à l'aide de mots-clés. Un autre avantage de cette méthode est qu'elle est utilisable avec n'importe quel trait visuel du moment que le regroupement d'individus dans l'espace visuel a un sens. Elle permet de comparer l'efficacité des traits visuels en fonction du mot-clé étudié. Enfin, cette méthode est facile à mettre en oeuvre, car elle ne necessite que le stockage des vecteurs centroîdes et d'écarts-types des classes visuelles de chaque mot.

Nos perspectives sont de proposer un filtrage plus fin et non booléen afin d'élaborer des coefficients de confiance pour les mots associés à une image en fonction de leur pertinence visuelle pour chaque image. Ceci afin de ranger les résultats d'une recherche, des images les plus pertinentes visuellement au moins pertinentes. Une autre perspective est de construire des classes visuelles non pas sur l'ensemble de l'espace, mais sur plusieurs sous-espaces et de combiner ensuite efficacement les résultats obtenus afin de choisir les meilleurs traits pour chaque mot en fonction des scores obtenus pour chaque sous-espaces comme dans [TOL 04]. Enfin, nous envisageons d'appliquer nos résultats sur les images du web.

8. Bibliographie

[Alm 04] ALMEIDA SOUZA COELHO T., PEIREIRA CALADO P., VIEIRA SOUZA L., RIBEIRO-NETO B., MUNTZ R., « Image Retrieval Using Multiple Evidence Ranking », *IEEE Transactions on Knowledge and Data Engineering*, 2004, p. 408-417.

[BAR 03] BARNARD K., DUYGULU P., DE FREITAS N., FORSYTH D., BLEI D., JORDAN M. I., « Matching Words and Pictures », *Journal of Machine Learning Research*, vol. 3, 2003, p. 1107-1135.

[DUD 00] DUDA R. O., HART P. E., STORK D. G., *Pattern Classification*, John Wiley and Sons, Inc., 2000.

[FAU 03] FAUQUEUR J., BOUJEMAA N., « Recherche d'images par Régions d'intérêt : Segmentation Grossière Rapide et Description Couleur Fine », *Techniques et Sciences Informatiques (TSI)*, vol. 22, n° 9, 2003.

[JEO 03] JEON J., LAVRENKO V., MANMATHA R., « Automatic image annotation and retrieval using cross-media relevance models », *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 2003, p. 119–126.

[LAN 67] LANCE G., WILLIAMS W., « A general theory of classificatory sorting strategies : I. Hierarchical systems », *Computer Journal*, vol. 9, 1967, p. 373-380.

[LI 03] LI J., WANG J. Z., « Automatic Linguistic Indexing of Pictures by a Statistical Modeling Approach », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, n° 9, 2003, p. 1075–1088, IEEE Computer Society.

[MON 03] MONAY F., GATICA-PEREZ D., « On image auto-annotation with latent space models », *Proc. ACM Int. Conf. on Multimedia (ACM MM)*, 2003, p. 275-278.

[MUL 02] MULLER H., MARCHAND-MAILLET S., PUN T., « The Truth about Corel – Evaluation in Image Retrieval. », *The Challenge of Image and Video Retrieval (CIVR2002)*, 2002.

[SHI 00] SHI J., MALIK J., « Normalized Cuts and Image Segmentation », *IEEE Transactions on Pattern Analysis and machine Intelligence*, vol. 22, n° 8, 2000, p. 888-905.

[TOL 04] TOLLARI S., GLOTIN H., LE MAITRE J., « Rehaussement de la classification textuelle d'images par leur contenu visuel », *Actes du 14ème Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle*, janvier 2004, p. 1383-1392.

[WAN 01] WANG J. Z., LI J., WIEDERHOLD G., « SIMPLIcity : Semantics-Sensitive Integrated Matching for Picture Libraries », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, n° 9, 2001, p. 947-963.

[WAN 04] WANG J., « http ://wang.ist.psu.edu/docs/home.shtml », 2004.