
Connaître la Couleur des Objets pour Segmenter les Images et Améliorer la Recherche d'Images sur le Web

Christophe Millet¹²

¹CEA, LIST, Laboratoire d'ingénierie de la connaissance multimédia multilingue
18 Route du Panorama, BP6, FONTENAY AUX ROSES, F-92265 France

²GET-ENST - Dept TSI - CNRS UMR 5141 LTCI
PARIS, France
chr.millet@gmail.com

RÉSUMÉ. Les résultats des moteurs actuels de recherche d'images sur le web sont très bruités, c'est-à-dire que quand nous cherchons un objet précis, un grand nombre d'images retournées par le système ne concerne pas la requête : environ 50% de ces images sont hors-sujet. Dans cet article, nous expliquons comment utiliser la connaissance de la couleur des objets pour trouver l'objet recherché dans les images, et nous proposons également deux méthodes pour connaître automatiquement la couleur d'un objet, afin que le traitement dans son ensemble puisse être complètement automatique. Les résultats obtenus montrent que cette méthode permet de réduire le bruit dans les images retournées d'environ 20% si nous gardons 25% des images, et fournit en même temps une segmentation automatique de l'objet qui pourrait servir par exemple pour le groupement ou pour l'apprentissage d'objets.

ABSTRACT. Results of current web image search engines are very noisy, that is, when we ask for a specific object, we are not ensured that this object is contained in all the images returned by the search engines: about 50% of the images returned are off-topic. In this paper, we explain how knowing the color of an object can help locating the object in images, and we also propose methods to automatically find the color of an object, so that the whole process can be fully automatic. Results reveal that this method allows us to reduce the noise in returned images while providing automatic segmentation that can be used for clustering or object learning.

MOTS-CLÉS : segmentation couleur, sémantique, recherche d'images sur le web.

KEYWORDS: color segmentation, semantics, web image search.

CATEGORIE : Jeune chercheur

1. Introduction

Internet contient beaucoup d'images qu'il pourrait être intéressant d'exploiter, notamment pour l'apprentissage d'objets. Il est possible d'accéder et de rechercher les images sur le web grâce aux moteurs de recherche existants où l'utilisateur tape une requête constituée d'un ou plusieurs mots clés, et reçoit en retour une liste d'images dont le titre, ou le texte des pages web où se trouve l'image, correspondent à ces mots clés. Cependant, la liste brute d'images retournées par ces moteurs est difficilement utilisable telle quelle pour l'apprentissage d'objets, car elle est trop bruitée. Une rapide évaluation des 50 premières images obtenues avec Google Image Search sur 50 requêtes nous a permis de constater qu'en moyenne, environ 50% des images obtenues sont du bruit : elles ne correspondent pas à la requête effectuée. Cela s'explique notamment par le fait que l'indexation et la recherche de ces images utilisent uniquement le texte et ne tient pas compte du contenu des images.

Dans cet article, nous nous concentrons sur l'utilisation des couleurs des objets dans le but d'analyser le contenu des images et d'améliorer les résultats retournés par les recherches sur Internet en termes de précision. Étant donné qu'aucune base de données décrivant les couleurs des objets n'existe actuellement, nous proposons également une méthode pour obtenir cette information automatiquement, même si la construction d'une telle base de données manuellement est également envisageable. Ces méthodes seront appliquées à la segmentation automatique de l'objet principal dans les images, et plus particulièrement des objets constitués de plusieurs couleurs (tel que le zèbre), ainsi qu'à la réduction du bruit et au reclassement des images retournées par un moteur de recherche d'images sur le web.

Segmentation automatique : lors d'une recherche d'images sur Internet, il serait intéressant de pouvoir automatiquement isoler l'objet recherché de son contexte. Une telle segmentation automatique des objets est un problème difficile : la plupart des images contiennent plus d'un objet, et donc, après avoir appliqué une segmentation automatique, il est difficile de savoir quel objet est celui que nous cherchons.

Afin de localiser automatiquement l'objet dans les images provenant d'Internet, (Ben-Haim *et al.*, 2006) segmentent les images en plusieurs régions, puis classent ces régions pour identifier des ensembles de régions similaires présentes dans des images différentes. Le plus grand ensemble est considéré comme celui représentant l'objet. Une approche similaire (Russel *et al.*, 2006) combine plusieurs algorithmes de segmentation pour chaque image afin de découvrir des objets dans une collection d'images et d'améliorer la qualité de la segmentation par rapport aux techniques n'utilisant qu'un seul algorithme de segmentation. Ces approches sont intéressantes, mais nous nous sommes rendu compte que dans le cas par exemple d'animaux d'extérieur, le plus grand ensemble de régions similaires représente souvent le contexte (ciel, herbe, ...) ou contient des régions sombres telles que les ombres ou les fonds noirs.

Nous proposons dans cet article d'utiliser la sémantique des objets : connaissant l'objet supposé être dans l'image, nous utilisons les connaissances sur cet objet, ici la couleur, pour guider la segmentation automatique. Une autre motivation de cette méthode est la possibilité de segmenter les objets tels que le zèbre ou le panda, qui sont constitués de plusieurs couleurs. La segmentation de tels objets n'est pas possible

avec les algorithmes classiques à cause du fort gradient existant aux frontières entre les couleurs. Liapis et al. (Liapis *et al.*, 2004) ont démontré qu'il était possible de segmenter un zèbre en utilisant un histogramme de luminance où chaque pixel est classifié, puis les classes sont propagées, mais cela nécessite de spécifier le nombre de classes avant la segmentation, et ne fonctionnerait pas pour les images complexes pour lesquelles par exemple le fond peut être segmenté en plusieurs régions.

Élimination des intrus et reclassement des images du web : même s'il est bien connu que les résultats des recherches d'images sur le web sont très bruités, peu de travaux se sont intéressés à les améliorer. Un reclassement des résultats à l'aide d'un modèle de pertinence sur le texte des pages web (Lin *et al.*, 2003) permettrait d'obtenir de 30% à 50% d'augmentation de la précision sur les 50 premières images, mais sans utiliser le contenu des images.

Un moyen de nettoyer les images serait de chercher des similarités entre les différentes images. Cette méthode est très difficile à appliquer aux images du web car beaucoup d'images ne contiennent pas l'objet recherché : le bruit est en moyenne de 50% et peut aller jusqu'à 85% pour certaines requêtes. De plus, il y a beaucoup de variations en termes de qualité d'images, si bien qu'il est difficile de trouver un motif se répétant. Dans (Ben-Haim *et al.*, 2006), les images sont reclassées selon la plus petite distance entre une région de l'image et le centre du groupe principal décrit ci-dessus.

Une approche donnant des résultats prometteurs pour nettoyer, réorganiser et faire de l'apprentissage à partir des résultats de Google Image Search (Fergus *et al.*, 2004, Fergus *et al.*, 2005) a été proposée en calculant les descripteurs SIFT sur les régions entourant des points d'intérêts afin d'entraîner un modèle sémantique latent probabiliste invariant par échelle et translation (TSI-pLSA) pour la classification d'objets. Ce modèle est appliqué sur les données brutes pour reclasser les images et enlever les images non pertinentes. Fergus et al. ont annoncé une amélioration d'environ 20% en précision avec un rappel de 15% (i.e. en rejetant 85% des images).

Il est également possible de nettoyer et reclasser les images en utilisant des techniques de groupement (Millet *et al.*, 2006). Les images sont d'abord segmentées automatiquement en 20 régions, puis les régions qui ne sont pas en contact avec le bord de l'image sont fusionnées et considérées comme l'objet à étudier. Cette région est ensuite indexée avec des descripteurs de texture et de couleur, qui servent à grouper les images avec l'algorithme des plus proches voisins partagés (*shared nearest neighbor, SNN*). Les images ne se retrouvant dans aucun groupe sont rejetées, et chaque groupe est trié en comparant les couleurs qu'il contient avec la liste des couleurs probables de l'objet recherché. Les résultats sont prometteurs, mais la segmentation n'est pas toujours capable d'identifier correctement l'objet principal.

Dans le présent article, au lieu d'utiliser la couleur pour trier les groupes, nous l'utilisons pour segmenter l'objet. Par exemple, après avoir récupéré sur Internet des images de zèbres, nous utilisons la connaissance qu'un zèbre a les couleurs "noir" et "blanc" pour chercher et segmenter dans l'image le plus gros objet constitué de ces couleurs. Dans ce cas, il n'est plus possible d'utiliser la couleur pour trier les images, mais en revanche, la taille et la position des régions segmentées peuvent être utilisées. Nous expliquons comment associer les noms des couleurs avec les valeurs TSV des

pixels dans la section 2. Puis, nous discutons en section 3 la possibilité de connaître automatiquement les couleurs des objets, et comment traiter les objets qui n'ont pas de couleur particulière. Dans la section 4, nous détaillons l'algorithme utilisé pour segmenter les images étant donné une ou plusieurs couleurs. Enfin, dans la section 5, nous expliquons et évaluons les applications proposées pour démontrer l'intérêt de la méthode proposée pour la segmentation, avant de conclure sur les travaux futurs.

2. La couleur d'un pixel

Attribuer un nom de couleur aux valeurs d'un pixel n'est pas si facile, d'une part car le nombre de couleurs à considérer est à définir, et d'autre part, la séparation entre les couleurs n'est pas bien définie. Une comparaison de plusieurs systèmes de nommage de couleur selon la facilité de leur utilisation pour un utilisateur (Berk *et al.*, 1982) a permis de conclure que les systèmes existants contenaient trop de noms de couleurs pour pouvoir être facilement mémorisés et utilisables par un annotateur humain. Berk *et al.* ont proposé un système de nommage de couleur (CNS) consistant en 10 couleurs fondamentales : *gris, noir, blanc, rouge, orange, marron, jaune, vert, bleu, violet*. À ces 10 couleurs peuvent se rajouter des adjectifs tels que *foncé, clair, vif, ...*, ou les formes en -âtre, tels que *marron rougeâtre*, formant un total de 627 couleurs.

Cependant, lorsque nous nommons une couleur, nous n'utilisons en général qu'une des 10 couleurs fondamentales énumérées ci-dessus, sans adjoindre d'adjectif. Nous avons donc décidé de considérer seulement 11 couleurs, en rajoutant le rose. Il nous semble en effet que le rose est une couleur très utilisée pour nommer les objets : beaucoup d'habits (principalement féminins) sont de cette couleur, ainsi que certains animaux tels que le cochon domestique ou le flamant rose. Afin d'établir la correspondance entre ces 11 couleurs et les pixels, nous avons recours à l'espace de couleurs TSV (teinte, saturation, valeur) qui a l'avantage d'être plus sémantique que l'espace RVB. En particulier, la teinte est très proche du concept de noms des couleurs. Dans notre espace TSV, chaque composante est dimensionnée entre 0 et 255. Une teinte négative est assignée aux pixels achromatiques ($R=V=B$).

Nous avons d'abord pensé à séparer clairement les différentes couleurs dans l'espace TSV, en assignant une unique couleur à chaque triplet (t,s,v). Cependant, nous avons remarqué que certains pixels pouvaient être nommés différemment selon leur contexte, et avons donc décidé de prendre cette ambiguïté en compte, et d'associer parfois plusieurs noms pour un seul triplet (t,s,v). Ainsi, au lieu d'associer un nom pour chaque valeur (t,s,v), nous associons une plage de valeur (t,s,v) pour chaque nom de couleur. Le tableau 1 détaille cette association.

Les valeurs ont été choisies expérimentalement après observation sur plusieurs centaines d'images, mais nécessiteraient une validation par plusieurs personnes pour assurer une certaine objectivité. D'autres méthodes telles que la logique floue pourraient servir à modéliser les frontières floues entre des couleurs, mais la méthode que nous avons développée suffit pour nos applications. De plus, dans notre méthode, les définitions des couleurs sont indépendantes les unes des autres, ce qui permet d'ajouter des noms de couleurs sans affecter les autres.

Couleur	Teinte	Saturation	Valeur
noir	< 0	tout	0 – 85
	0 – 255	tout	0 – 40
gris	< 0	tout	80 – 180
blanc	< 0	tout	175 – 255
	0 – 30	0 – 90	200 – 255
rose	235 – 245	tout	tout
rouge	0 – 15	tout	tout
	240 – 255	tout	tout
orange	14 – 30	tout	tout
jaune	20 – 50	tout	190 – 255
vert	20 – 50	tout	0 – 200
	50 – 125	tout	tout
bleu	110 – 200	tout	tout
violet	200 – 235	tout	tout
marron	0 – 40	25 – 140	75 – 200
	230 – 255	25 – 135	55 – 190
	< 0	10 – 30	60 – 165

Tableau 1. Correspondance entre les noms des couleurs et l'espace TSV.

3. La couleur d'un objet

Déterminer la couleur d'un objet automatiquement n'est pas facile, et il n'existe actuellement pas de base de données créée automatiquement donnant ce genre d'information. Notamment, tous les objets n'ont pas une couleur unique bien définie : certains objets comme le zèbre ou le panda sont constitués d'au moins deux couleurs, et d'autres objets, tels qu'une maison ou une chaise, existent en différentes couleurs. En général, les objets avec des couleurs bien définies sont les objets naturels, comprenant les minéraux, les animaux et les plantes, tandis que les objets sans couleur spécifique sont ceux créés par l'homme. Dans le cas des animaux, certaines sous espèces ont été nommées en fonction de leurs couleurs, et donc le recours à une ontologie contenant les relations d'hyponymies peut nous aider à déterminer si un animal peut exister de différentes couleurs. Par exemple, l'ontologie WordNet (Fellbaum, 1998) nous indique que "loup" possède les sous-espèces "loup gris", "loup rouge" et "loup blanc"; "ours" peut être "ours brun", "ours noir", "ours blanc". Pour cette raison, il est préférable de ne s'intéresser qu'aux objets qui n'ont pas d'hyponymes dans WordNet, afin que la couleur ne présente pas trop de variations.

Étant donné que les objets fabriqués par l'homme peuvent avoir diverses couleurs, la couleur de l'objet est souvent spécifiée dans la page web contenant l'image, ou dans le nom de l'image. Ainsi, la requête "*red car*" retourne plusieurs milliers d'images. Cela n'est pas valable en revanche pour les objets naturels. Par exemple, les requêtes "*brown lion*" et "*tan lion*" retournent très peu d'images (respectivement 58 et 38 sur ask.com), dont la plupart sont des ours en peluche, qui font donc partie de la catégorie "fabriqué par l'homme". Il n'y a que très peu d'animaux dans les résultats, car les moteurs de recherche d'images utilisent le texte proche de l'image et le nom de l'image pour effectuer la recherche, mais l'information sur la couleur ne sera en général pas

présente étant donné que peu de personnes annotent les informations triviales du type “voici un lion de couleur fauve”. Au contraire, les couleurs rares pour les animaux ont plus de chance d’être annotées : la requête “*white lion*” retourne beaucoup plus d’images (3820 sur ask.com) que “*brown lion*” ou “*tan lion*”.

Par conséquent, lors de la récupération des images sur Internet, nous spécifions la couleur dans la requête pour les objets fabriqués par l’homme, mais pas pour les objets naturels. Les requêtes, en italique dans le texte, sont effectuées en anglais, car les moteurs de recherche retournent beaucoup plus d’images dans cette langue. Nos algorithmes sont testés sur les objets suivants :

- des animaux avec une couleur prédéterminée : *fire ant* (fourmi rouge), *beaver* (castor), *cougar* (puma), *crab* (crabe), *crocodile* (crocodile),
- des animaux avec deux couleurs : *blue-and-yellow macaw* (ara bleu-et-jaune), *ladybug* (coccinelle), *leopard* (léopard), *panda* (panda), *zebra* (zèbre),
- des objets créés par l’homme : *camera* (*white, black*) (appareil photo), *cell phone* (*black, blue, green, red, white*) (téléphone portable), *chair* (*black, blue, green, red, white*) (chaise), *cup* (*black, green, red, white*) (tasse), *Porsche* (*red*) (Porsche).

Nous testons deux méthodes permettant de connaître automatiquement la couleur d’un objet : l’une utilisant un corpus textuel, et l’autre à partir du contenu des images.

À partir du texte : les couleurs des objets peuvent être extraites automatiquement à partir de très grands corpus textuels, et nous proposons de le faire en considérant le web comme corpus. L’idée est d’étudier si le nom d’un objet donné apparaît souvent près du nom de certaines couleurs dans les textes. Nous avons expérimenté deux variantes : pour savoir la couleur d’un castor, la variante 1 compte les occurrences de “*brown beaver*” où *brown* peut être remplacé par n’importe quelle couleur. La variante 2 cherche le texte “*beavers are brown*”. La présence des guillemets indique que les mots doivent être accolés : d’autres noms de couleurs apparaissant dans la même page web peuvent ne pas être liés à l’objet considéré. La catégorie de l’objet peut être utilisée pour réduire le bruit, par exemple avec les requêtes “*brown beaver*” *animal* et “*beavers are brown*” *animal*. La couleur retenue est alors la couleur donnant le plus grand nombre d’occurrences : cette méthode, simple, n’est pas adaptée aux objets composés de deux ou plusieurs couleurs.

Nous utilisons le vocabulaire de 11 noms de couleurs présenté en section 2. Le tableau 2 liste les trois couleurs principales retournées pour *beaver* en utilisant Yahoo ! Search, et le nombre d’occurrences entre parenthèses. *Brown* puis *black* sont les deux couleurs principales attendues, retrouvées par la variante 2.

“C beaver”	“C beaver” an.	“beavers are C”	“beavers are C” an.
brown (43 200)	green (10 800)	brown (98)	brown (26)
black (28 100)	brown (7 550)	red (6)	black (1)
green (20 400)	black (2 800)	black (3)	-

Tableau 2. Recherche de la couleur des objets par méthode textuelle. C est à remplacer par le nom de la couleur, an. est mis pour “animal”

Cet exemple est représentatif de ce que nous observons en général pour les autres objets : la variante 2 (colonnes 3 et 4) donne des résultats plus précis, mais retourne des occurrences plus faibles que la première. L'inconvénient de la variante 1 est qu'elle est sensible aux noms propres et aux locutions. Par exemple, *Green Beaver* est le nom d'une entreprise et le nom d'un cocktail ; *white house* retourne de nombreux résultats, mais nous ne devons pas en déduire que la plupart des maisons sont blanches. Pour les locutions, l'existence de l'animal *blue whale* fait de bleu la couleur prédominante pour les baleines, et *white chocolate* a plus de résultats que *black chocolate* ou *brown chocolate*. La variante 2 n'a pas ce problème, mais elle ne retourne parfois aucun résultat, et dans ce cas, il faut recourir à la variante 1.

À partir de l'image : au lieu d'utiliser le texte pour connaître la couleur d'un objet, une autre méthode consiste à se servir du contenu des images. Nous proposons de faire une moyenne sur toutes les images retournées par un moteur de recherche d'images sur Internet. En partant de l'hypothèse que les images contiennent habituellement l'objet centré dans l'image et entouré par un fond, nous ne prenons en compte que les pixels dans une fenêtre au centre de l'image, dont la largeur et la hauteur sont la moitié de celles de l'image, comme illustré sur la figure 1.



Figure 1. Fenêtre des pixels considérés pour déterminer la couleur d'un objet

Cette fenêtre peut contenir des pixels de l'environnement, mais en prenant suffisamment d'images, la couleur de l'animal devrait prédominer, étant donné qu'elle sera la même pour toutes les images alors que la couleur du fond pourra varier. Le tableau 3 énumère les trois premières couleurs obtenues pour quelques objets. Les tests complets portent sur les 27 objets cités en début de section 3 et 100 images par objet.

red Porsche	beaver	crab	zebra	ladybug
<u>red</u> (44.5)	<u>brown</u> (24.0)	<u>brown</u> (23.0)	<u>black</u> (20.7)	<u>red</u> (23.1)
blue (11.8)	white (16.9)	<u>red</u> (22.8)	<u>white</u> (20.6)	white (20.7)
black (11.0)	red (13.0)	white (14.7)	brown (17.3)	brown (14.7)

Tableau 3. Recherche de la couleur des objets par méthode image : pourcentage des trois couleurs principales pour chaque objet. Les couleurs attendues sont soulignées

Au vu de ces résultats, un seuil à 15% semble raisonnable pour choisir les couleurs à conserver, et cette méthode permet donc, à l'inverse de la méthode textuelle, de tenir compte des objets à plusieurs couleurs. Pour *red Porsche*, et en général pour les objets sans couleur spécifique, pour lesquels la couleur est donc rajoutée à la requête, nous obtenons de bons résultats. La présence du blanc dans *beaver* et *ladybug* est due au fait que beaucoup de ces images sont des dessins en noir sur fond blanc. Les autres

couleurs viennent principalement de l'environnement. Les résultats sont plus intéressants pour les objets dont la couleur n'a pas été spécifiée dans la requête : sur nos 10 animaux de tests, la première couleur estimée est correcte dans 70% des cas pour la méthode textuelle, et 90% pour la méthode image (l'erreur se produit pour *crocodile* où la méthode retourne "marron" comme couleur principale, et la bonne couleur, "vert", en deuxième position). Cette méthode a également l'avantage d'être statistiquement plus significative que la méthode textuelle. Toutefois, il n'est pas évident de savoir combien de couleurs considérer pour chaque objet. Une amélioration serait, pour la méthode textuelle, d'utiliser une source plus fiable que le web, telle qu'un dictionnaire ou une encyclopédie. Pour les expériences qui suivent, nous considérons que nous avons construit manuellement la base de données donnant la couleur des objets, ce qui est tout à fait envisageable en pratique.

4. Segmentation par la couleur

Pour une image et une ou plusieurs couleurs données, la segmentation de l'image se fait en cinq étapes. 1) Classifier chaque pixel comme appartenant ou non aux couleurs données, comme expliqué dans la section 2. Cela produit une image binaire avec les "pixels objets" et les "pixels de fond". 2) Enlever le bruit ou les objets fins avec une ouverture selon un élément structurant de taille 1. 3) Effectuer une fermeture par un élément structurant de taille 5 afin que les régions objets proches se rejoignent. 4) Garder la plus grande région. 5) Enlever les trous, définis comme des "pixels de fonds" entièrement entourés par des "pixels objets". Cette étape est fondée sur l'hypothèse que l'objet n'a pas de trou, ce qui est le cas pour la plupart des objets.

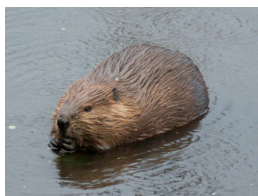


Figure 2. *Résultat de la segmentation d'un castor avec la couleur marron.*

Il est possible d'utiliser plus d'une couleur dans l'étape 1, afin de segmenter des animaux tels que le zèbre en utilisant à la fois les couleurs "noir" et "blanc" comme couleurs d'objets. Les quatre autres étapes servent à ne conserver qu'une région et à adoucir les bords de cette région. Enlever le bruit à l'aide d'une ouverture (étape 2) est utile pour ne pas amplifier le bruit à l'étape suivante lors de la fermeture. La fermeture est utile pour fusionner les régions proches ensemble, et améliorer les contours.

5. Applications

Nous avons appliqué cette méthode de segmentation d'une part à la segmentation automatique, permettant de séparer un objet de son contexte, notamment les objets

composés de plusieurs couleurs ; d'autre part à l'élimination des intrus et au reclassement des images résultats d'une requête sur Internet.

Segmentation automatique : les algorithmes de segmentation automatique utilisant les gradients ne peuvent pas segmenter un zèbre et considèrent les rayures comme des objets séparés, à cause du fort gradient entre elles. Si nous n'ajoutons pas de connaissance à l'algorithme sur ce qu'est un zèbre, il ne peut pas savoir comment nous voyons cet animal. Nous proposons d'utiliser les couleurs des objets pour segmenter une image supposée contenir cet objet, comme expliqué dans la section 4. Des exemples sont présentés figure 3.

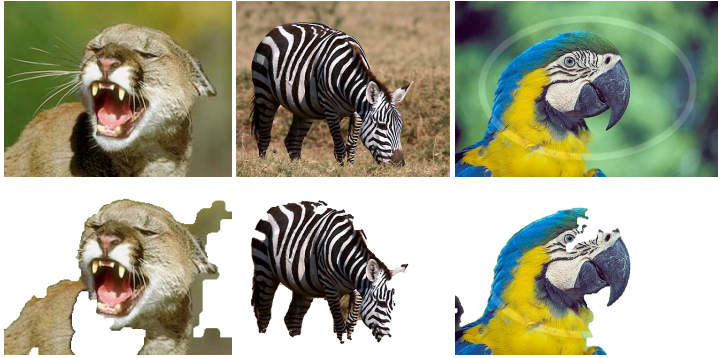


Figure 3. Exemple de segmentations sur un cougar (marron), un zèbre (noir + blanc) et un macaw bleu et jaune (bleu + jaune)

Ces résultats illustrent le rôle de l'étape 5 : elle sert à segmenter les objets qui sont principalement composés de certaines couleurs, mais pas entièrement. Par exemple, le puma est principalement marron, mais contient du rouge, du blanc et du noir. Cependant, cette région est entièrement contenue dans une région marron, ce qui permet de la fusionner avec l'objet. La possibilité de segmenter des objets ayant deux couleurs est démontrée sur le zèbre et le macaw. Néanmoins, segmenter en utilisant seulement les couleurs, comme nous le faisons ici, ne donne pas de bons résultats pour les objets cachés dans leur environnement, et nous pourrions probablement améliorer l'algorithme en le combinant avec un algorithme de segmentation par la texture.

Filtrage des images du web : nous souhaitons réduire le bruit d'environ 50% que nous avons constaté sur les moteurs de recherche d'images sur Internet (voir tableau 4). Ici, nous proposons d'enlever les images non pertinentes en utilisant la couleur des objets. L'idée est d'analyser à quoi la région obtenue par segmentation ressemble. Dans une situation idéale, l'objet est centré dans l'image, et totalement contenu dans celle-ci, sans être trop petit. Nous avons donc décidé de rejeter une image si la région segmentée occupe moins de 20% de la surface de l'image, touche plus de 80% des pixels du bord et est telle que la distance de son barycentre (x_R, y_R) au centre de l'image (x_I, y_I) est plus petite que 40% de la distance du coin de l'image au centre :

$$(x_R - x_I)^2 + (y_R - y_I)^2 < 0.4 * (x_i^2 + y_i^2)$$

Objet	Avant	Après	Objet	Avant	Après
beaver (brown)	40.2%	72.7%	green cell phone	17.9%	46.2%
cougar (brown)	72.0%	93.0%	red cell phone	22.5%	37.5%
crab (red)	55.8%	70.8%	white cell phone	12.9%	30.0%
crocodile (green)	79.5%	85.7%	black chair	66.7%	83.3%
fire ant (red)	61.0%	50.0%	blue chair	50.9%	58.8%
blue-and-yellow macaw	79.4%	82.6%	green chair	22.2%	50.0%
ladybug (red+black)	72.8%	73.3%	red chair	70.6%	88.5%
leopard (brown+yellow)	84.9%	89.7%	white chair	62.0%	83.3%
panda (black+white)	80.0%	81.5%	black cup	35.2%	66.7%
zebra (black+white)	76.7%	89.3%	green cup	34.5%	80.0%
black camera	69.1%	79.1%	red cup	36.5%	50.0%
white camera	18.6%	50.0%	white cup	33.9%	66.7%
black cell phone	43.0%	87.0%	red Porsche	85.0%	85.7%
blue cell phone	28.6%	42.4%			
moy. par classes	52.0%	69.4%	moy. par images	53.5%	72.0%

Tableau 4. Précisions avant et après filtrage pour différents objets en gardant 25 images sur les 100. Seul fire ant a une baisse de précision, car beaucoup d'images de peau irritée par des morsures de fourmis sont gardées avec la couleur rouge.

La précision augmente pour toutes les classes, sauf pour *fire ant* (cf. légende du tableau). Quatre classes ont une précision inférieure à 50% après filtrage (contre 12 avant filtrage) : pour les téléphones portables, beaucoup d'images représentent un autre objet, mais de la couleur demandée. Ces erreurs ne peuvent pas être évitées avec notre méthode mais pourraient l'être en rajoutant un groupement par texture.

Dans le tableau 4, les paramètres ont été choisis pour maximiser la précision tout en gardant au moins 25% des images. Ne garder que 25% des images n'est pas gênant quand on étudie les images du web : étant donné leur quantité, un utilisateur ne regardera pas la totalité des images disponibles, et est donc plus intéressé par la qualité (précision) que par la quantité (rappel). Pour une utilisation en apprentissage d'objets, réduire le bruit est également plus important qu'obtenir beaucoup d'images. Nous avons essayé d'introduire un critère rejetant les régions plus grandes qu'une certaine taille, mais le critère sur les pixels du bord a donné de meilleurs résultats.

La figure 4 détaille l'influence de chaque critère individuellement sur la précision. La plus grande proportion d'objets pertinents se trouve dans les images où la taille de l'objet est entre 10% et 50% de la taille de l'image. Les deux autres graphiques ont été calculés sur cet ensemble, et montrent qu'un objet a plus de chance d'être pertinent s'il ne touche pas le bord de l'image et s'il est centré. Il est possible d'améliorer encore la précision jusqu'à 86% en rendant les paramètres plus contraignants, ce qui réduit le nombre d'images après filtrage jusqu'à seulement 5%, comme montré sur la figure 5. En choisissant de conserver 15% des images comme dans (Fergus *et al.*, 2004), la précision augmente de 24% (de 53% à 77%), ce qui est comparable aux 20% annoncés, même si les deux bases de données sont différentes.

Reclasser les images du web : nous utilisons ici les critères développés ci-dessus pour trier les images retournées par un moteur de recherche classique. Pour une région segmentée donnée dans une image, soit S le rapport de la surface de la région divisé

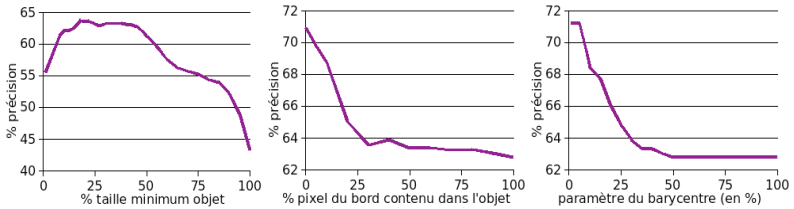


Figure 4. Influence des paramètres sur la précision après filtrage, de gauche à droite : paramètre de la taille, paramètre du bord, et paramètre du barycentre. Les graphiques pour les paramètres du bord et du barycentre sont calculés sur les régions dont la taille est entre 10% et 50% de la taille de l'image, correspondant aux meilleures images pour le critère de la taille (à gauche).

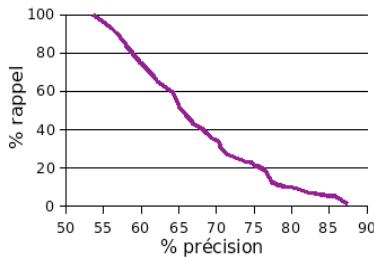


Figure 5. Relation entre la précision (abscisse) et le nombre d'images (ordonnée).

par la surface de l'image, et B le pourcentage de pixels du bord de l'image qui sont contenus dans l'objet. La section 5.2 montre que les objets ont plus de chance d'être pertinents pour B proche de 0 et S entre 0,2 et 0,4. Nous calculons le score Σ suivant :

$$\Sigma = (1 - B) * f(S) \text{ avec } f(S) = \begin{cases} 1 & \text{si } 0,2 \leq S \leq 0,4 \\ \frac{S}{0,2} & \text{si } s < 0,2 \\ \frac{1-S}{0,6} & \text{si } s > 0,4 \end{cases}$$

Les images sont ensuite triées par ordre décroissant de Σ . La figure 6 montre le résultat pour la requête *green cup*. Après reclassement, nous observons principalement des images avec un objet central de la bonne couleur, et dans l'exemple donné, l'amélioration en précision est bien visible. Dans la cas où la recherche serait simplement *cup*, et comme l'objet n'a pas de couleur spécifique, nous pourrions prévoir de lancer les requêtes du type *color cup* pour toutes les couleurs, de reclasser chaque résultat, puis de réunir le tout, ou d'afficher chaque couleur séparément.



Figure 6. À gauche : les 20 premières images retournées par ask.com. À droite : les 20 premières après reclassement des 100 premières images de ask.com.

6. Conclusion

Dans cet article, nous avons décrit comment utiliser la couleur pour traiter les images du web. Nous sommes capables de segmenter les objets dans les images, notamment les objets possédant plusieurs couleurs, et avons également démontré qu'il était possible de réduire le bruit et de reclasser les images. Nous prévoyons dans le futur d'appliquer un traitement similaire pour extraire des informations sur la texture ("rayure", "tache", ...) et sur l'environnement ("désert", "route", ...) des objets et de les traduire en termes de traitement d'images. Nous voudrions également utiliser les bases nettoyées et segmentées obtenues pour faire de l'apprentissage d'objets.

7. Bibliographie

- Ben-Haim N., Babenko B., Belongie S., « Improving Web-based Image Search via Content Based Clustering », *CVPRW*, 2006.
- Berk T., Brownston L., Kaufman A., « A New Color-Naming System for Graphics Languages », *IEEE Computer Graphics Applications*, vol. 2, n° 3, p. 37-44, May, 1982.
- Fellbaum C., *WordNet - An Electronic Lexical Database*, Bradford books, 1998.
- Fergus R., Fei-Fei L., Perona P., Zisserman A., « Learning Object Categories from Google's Image Search. », *Proceedings of the 10th ICCV*, vol. 2, p. 1816-1823, October, 2005.
- Fergus R., Perona P., Zisserman A., « A Visual Category Filter for Google Images », *Lecture notes in computer science*, p. 242-256, 2004.
- Liapis S., Sifakis E., Tziritas G., « Colour and Texture Segmentation using Wavelet Frame Analysis, Deterministic Relaxation, and Fast Marching Algorithms », *IEEE Transactions on Multimedia*, vol. 6, n° 5, p. 676-686, October, 2004.
- Lin W.-H., Jin R., Hauptmann A., « Web Image Retrieval Re-Ranking with Relevance Model », *Proceedings of IEEE/WIC*, p. 242-248, 2003.
- Millet C., Grefenstette G., Bloch I., Moëllic P.-A., Hède P., « Automatically Populating an Image Ontology and Semantic Color Filtering », *International Workshop Ontoimage'2006 Language Resources for Content-Based Image Retrieval*, Genoa, Italy, p. 34-39, 2006.
- Russel B. C., Efros A. A., Sivic J., Freeman W. T., Zisserman A., « Using Multiple Segmentations to Discover Objects and their Extent in Image Collections », *IEEE CVPR*, vol. 2, p. 1605-1614, 2006.