Institut de la Francophonie pour l'Informatique

RAPPORT FINAL

TRAVAIL PERSONNEL ENCADRÉ

Sujet : « Recherche d'image par le contenu sur les requêtes textuelles »

Encadrement : Prof. Alain BOUCHER (IFI)

Étudiant : DAO Van Sang

Promotion : 16

Hanoï, juillet 2012

Remerciements

Je tiens tout d'abord à remercier notre encadrement Alain Boucher, professeur à l'Institut de la Francophonie pour l'Informatique. Il m'a aidé beaucoup dans la réalisation de notre sujet. Il m'a aussi donné des connaissances de base et des documents nécessaires pour réaliser notre sujet.

Je tiens aussi remercier mes amis qui m'ont aidé beaucoup à résoudre les problèmes difficiles que j'ai rencontré sur la réalisation du sujet. Particulièrement, merci bien à M. Tran Thi Cam Giang, qui a le même sujet du TPE avec moi.

Finalement, j'adresse un grand merci à toute ma famille et plus particulièrement à ma femme pour leur soutien et leur énorme encouragement au long de la réalisation de mon TPE.

Table de matières

Chapitre I – Introduction

1.1 Contexte du travail

Notre travail à rendre dans ce Travail Personnel Encadré (TPE) se situe dans le domaine de la recherche d'image.

De nos jours, la branche informatique se développe de plus en plus fort. Avec l'explosion d'Internet et aussi le développement à grande échelle de l'équipement numérique. Il n'est pas difficile d'avoir des bases d'images numériques contenant plusieurs milliers, voire plusieurs dizaines de milliers d'images, que ce soit des bases ciblées pour un domaine d'activité professionnelle (journalisme, tourisme, éducation, musées…) ou tout simplement pour les particuliers qui accumulent d'immenses bases de photographies numériques (souvenirs, voyages, famille, événements…). Il continue de plus en plus chaque jour. C'est-à-dire, les besoins sont énormes avec la quantité de plus en plus grande d'informations stockées sous forme multimédia.

Pour gérer et utiliser efficacement ces bases d'images, un système de recherche d'image est vraiment nécessaire. Il est essentiel d'étudier et de construire les outils qui nous permettent de trouver rapidement et exactement les images dont nous avons besoin. C'est pourquoi le sujet de la recherche d'images devient un sujet très actif dans la communauté internationale depuis plus d'une vingtaine d'années. Il est ce qui a attiré l'attention et la force de beaucoup de chercheurs dans le notrede. Je suis parmi eux. En plus, ce domaine est recherché depuis longtemps, mais il n'existe pas de système (en ligne et hors ligne aussi) qui nous permet de trouver exactement. Plusieurs concepts proposés se situent encore en théorie. Je m'intéresse donc beaucoup à choisir ce domaine pour étudier.

1.2 Objectifs

1.2.1 **Point de vue général du sujet de TPE**

Comme ce que nous avons dit ci-dessus, les moteurs de recherche d'images sont considérés comme des outils très utiles qui permettent aux gens d'accéder facilement et de trouver efficacement les images que l'on recherche parmi la grande masse d'images disponible. Actuellement, on peut distinguer deux principales techniques de recherche d'images. Ce sont la technique de *la recherche d'image sur les requêtes textuelles* et la technique de *la recherche d'image par le contenu.*

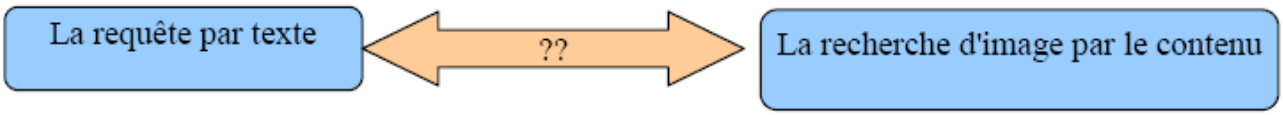
Sur l'internet, il existe déjà les moteurs de recherche d'image. Nous pouvons citer par exemple Google Image, Bing, Yahoo, etc. La plupart de ces systèmes sont les systèmes en utilisant le nom des images, en utilisant des annotations manuelles. Qu'est-ce qu'il se passera si on n'a pas de texte (noms, annotations, descriptions, etc.) accompagnant les images? C'est impossible ou trop difficile à trouver les images dont on a besoin. La recherche est encore plus problématique.

L'idée donnée est de faire les liens entre la recherche d'image par les requêtes textuelles et le contenu de l'image que l'on recherche. Plusieurs différents moyens sont proposés pour cela comme l'apprentissage, l'ontologie, etc. Dans le cadre de ce TPE, nous allons donner une approche différente et travaillerons avec une base d'images hors ligne sans annotation. Pour résoudre le problème abordé ci-dessus, nous suivrons des étapes suivantes :

- D'abord, l'utilisateur fera une requête texte par mots-clés comme soleil, ciel... ou n'importe quel mot. Cette requête textuelle sera envoyée à un moteur de recherche d'images sur Internet par exemple Google Images.
- Puis, avec les images résultats, on choisira quelques bonnes images qui seront utilisées pour construire un ensemble de caractéristiques (histogrammes, texture, etc.) représentant le mot-clé.
- Enfin, les caractéristiques images ainsi définies seront utilisées pour recherche dans la base d'images qui nous intéresse comme les bases d'images: Wang, Caltech 256....

1.2.2 **Objectif du TPE**

Dans le cadre de ce TPE, je vais chercher les liens entre une requête par texte (mots-clés) et le contenu de l'image que l'on recherche. L'objectif de notre TPE est de proposer et d'étudier les méthodes qui permettent d'apprendre les associations texte-image et de construire un ensemble de caractéristiques d'image représentant le mot-clé.



1.3 Travaux à rendre

1.3.1 **Travail théorique**

- Etudier le domaine de l'indexation, de la segmentation et la recherche d'images par le contenu, CBIR : les concepts et les méthodes à résoudre CBIR.
- Découvrir l'outil Google Image pour comprendre comment fonctionne-t-il avec la requête textuelle.
- Découvrir le site http://Image-Net.org à comprendre la façon d'organiser et fonctionner. Nous pouvons profiter de ce site dans cadre du TPE. Il fournit des outils et des API.
- Faire un état de l'art sur les méthodes existantes permettant de faire une requête texte et une recherche d'images par le contenu.
- Étudier les méthodes permet d'apprendre les associations texte-images (par exemple annotation automatique ou semi-automatique).
- Sélectionner parmi les images retournées par un moteur de recherche, quelles sont les images pertinentes pour un mot-clé. Tester l'homogénéité des images retournées (variance intra-classe après clustering), multiples requêtes au moteur de recherche avec des mots proches ou des traductions en plusieurs langues.

1.3.2 Travail pratique

- Développer un prototype de requête texte pour une recherche d'images par le contenu en utilisant Internet pour l'association entre le texte et le contenu de l'image.
- Tester ce prototype en utilisant différents moteurs de recherche d'images ou bases d'images disponibles sur Internet.
- Mesurer la performance de cette application.

1.4 *Termes techniques principales*

La requête textuelle : C'est-à-dire, l'utilisateur envoie un mot-clé ou une phrase qui se compose des mots-clés à un moteur de recherche. Par exemple, on tape quelque chose (nombres, caractères, etc.) dans le champ de recherche du moteur de recherche Google. Puis, on appuie le bouton « Recherche » ci-contre. On a réalisé donc une requête textuelle.

Le contenu de l'image : cela peut être décrit à deux niveaux : au niveau «numérique», une image contient des pixels colorés desquels on peut extraire des descripteurs de couleurs, de textures et de formes..., et au niveau «sémantique», une image peut être interprétée, elle a au moins une signification. Malheureusement, dans les systèmes d'image actuels, les images sont décrites au niveau numérique alors que les utilisateurs sont intéressés par leur contenu sémantique, et il est actuellement difficile de trouver des correspondances entre le niveau numérique et le niveau sémantique.

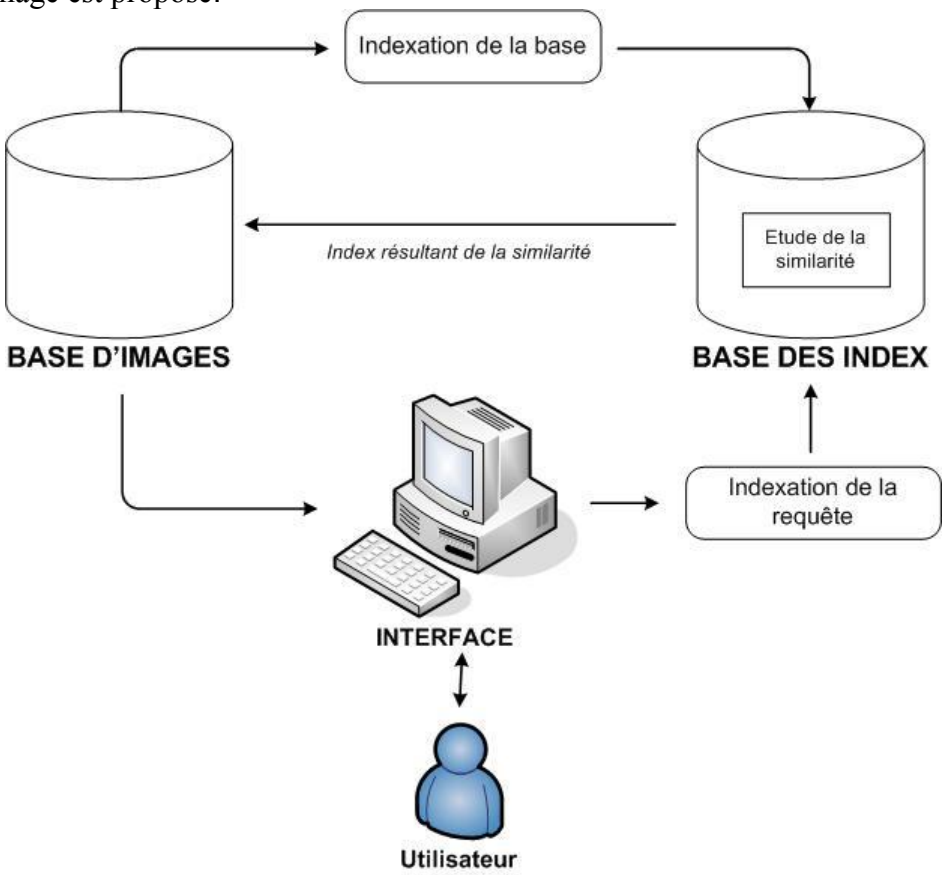
La recherche d'images basée sur les requêtes textuelles : (connue sous le nom en l'anglais de « Text-based Image Retrieval » ou TBIR). Dans cette approche, on utilise des techniques de bases de données traditionnelles pour gérer les images. Par exemple, les images sont annotées par le texte. Grâce à des descriptions textuelles, les images peuvent être organisées suivant le thème ou la

sémantique pour faciliter la recherche d'image. Au début, on a étudié et construit les systèmes sous cette forme. Plusieurs systèmes commerciaux ont adopté cette technique. Mais, il existe quelques inconvénients :

- *Premièrement*, les images ne sont pas toujours annotées, et leur annotation manuelle peut s'avérer très coûteuse en temps.

– *Deuxièmement*, l'annotation de l'homme est subjective: la même image peut être annotée différemment par différents observateurs. En outre, si on se fond exclusivement le texte, cela peut s'avérer insuffisant, par exemple, lorsque l'utilisateur s'intéresse aux éléments visuels de l'image. Ce sont des choses qui peuvent être difficilement décrites par des mots.

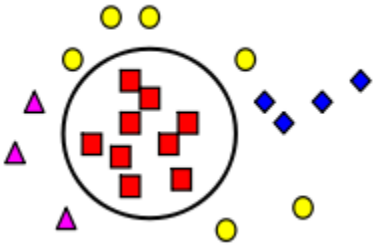
La recherche d'image par le contenu : (connue sous le nom en l'anglais de « Content-based Image Retrieval » ou CBIR). Dans années 90, pour surnotreter les défaites des systèmes de recherche l'image basée sur l'annotation textuelle manuelle d'images, le système de recherche d'image basée sur le contenu d'image est proposé.



Un système CBIR .

Dans le système de la recherche d'images par le contenu, quand l'utilisateur veut chercher des images, initialement, il donne une requête sous forme une image exemple. Le moteur de recherche calcule les similarités entre les images exemplaires et les images dans la base d'images en utilisant la fonction de similarité. Puis, Il ordonne les résultats en basant sur les similarités. Finalement, le moteur de recherche donne les résultats à l'utilisateur par une liste d'images ordonnées.

L'apprentissage d'image : un ensemble d'images passe un algorithme d'apprentissage. Après cela, on peut récupérer des images souhaites : des images pertinentes ou l'estimation de distribution des images dans cet ensemble. On utilise souvent des caractéristiques d'images pour faire l'apprentissage, par exemple : histogramme, texture, moment de couleur, etc. Dans le cadre de ce TPE, on va utiliser la méthode SVM one-class pour faire l'estimation de distribution des images récupérées è partir de Google Image.



Méthode SVM one-class.

1.5 Contenu du rapport

Ce rapport est organisé en deux parties principales. Le chapitre 1 contient l'introduction du sujet, le contexte du travail. Le chapitre 2 se consacre à l'état de l'art. Je vais présenter et analyser certaines solutions existantes pour résoudre la recherche d'image par le contenu sur les requêtes textuelles. Ensuite, je propose une solution à réaliser dans notre TPE.

Dans la partie suivante, je vais présenter notre système de recherche d'image par le contenu sur les requêtes textuelles basé sur des caractéristiques des images et un algorithme d'apprentissage, les résultats obtenus et l'analyse de ces résultats. En fin, je vais donner quelques conclusions et les perspectives.

Chapitre II – Etat de l'art

Le but principal de ce projet est de faire le lien entre une requête textuelle et le contenu de l'image que l'on recherche en utilisant Internet pour l'association entre le texte et le contenu d'image. Dans le chapitre précédent, nous avons présenté une analyse sommaire du sujet, y compris des exigences et des problèmes à résoudre dans le cadre du TPE. Pour analyser de façon profonde des systèmes de recherche d'image par le contenu sur une requête textuelle, dans ce chapitre, nous allons citer en détail quelques systèmes de recherche d'image en utilisant Internet pour l'association entre une requête texte et le contenu d'image existants. On donne aussi quelques points de vue sur ces systèmes. Enfin, on va proposer un système pour faire la recherche d'image par le contenu sur des requêtes textuelles.

2.1 Système proposé par Gang Wang et David Forsyth

2.1.1 Introduction

Le premier système est proposé par Gang Wang et David Forsyth « **Récupération des images en exploitant les ressources de connaissances en ligne** » (Le titre en anglais est *Object image retrieval by exploiting online knowledge resources*, CVPR. 2008.). [1]

Dans le domaine de recherche d'image, il existe deux méthodes principales. Ce sont la méthode de recherche par le contenu (CBIR) et par le texte (TBIR). Dans cet article, les auteurs décrivent une méthode pour récupérer des images souhaitées sur le web pages avec des étiquettes de classe d'objet spécifiée, en utilisant une analyse du texte autour de l'image et l'apparence de l'image (*Cette dernière est considéré le contenu visuel d'image, par exemple : le contraste de l'image, le contraste de l'image, la couleur, l'histogramme, etc*.). Autrement dit, c'est une association entre le contenu et le texte d'une image. Cette méthode permet de déterminer si un objet est à la fois décrit dans le texte et apparaît dans une image, en utilisant une image discriminante et un modèle de textes génératif. Cette modèle est étudiée et développé grâce à l'exploitation en ligne mis en place les ressources de connaissances (pages Wikipédia de textes; Flickr et Caltech ensembles de données d'image). Ces ressources fournissent des informations de texte et d'apparences d'objets. Pour concevoir le modèle de textes, les auteurs comptent sur le site Wikipédia. Ils pensent qu'avec un mot-clé, ce site leur donnera les résultats plus exacts que les autres. Cependant, les résultats expérimentaux notretrent aussi l'efficacité de cette approche sur ce nouvel ensemble de données.

2.1.2 La structure du système

Comme ce qui est présenté ci-dessus, le système se compose de deux modèles : *le modèle de texte* et *le modèle d'image*.

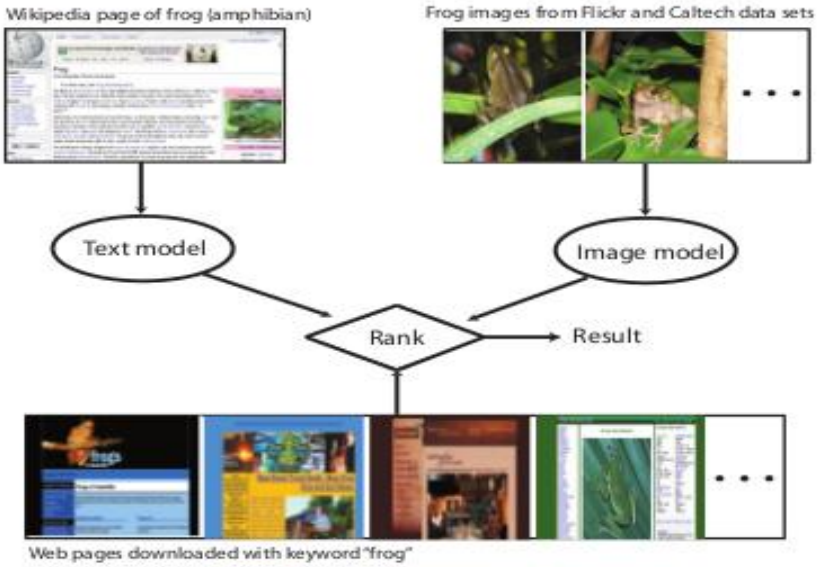


Figure 1 : Dans l'approche de Wang Dang et David Forsthy. Quand on fait une requête textuelle avec un mot « frog », nous recueillons beaucoup de pages Web en saisissant bruyants «frog» au moteur de recherche Google. La page Wikipédia de grenouille (Il appartient à l'amphibien) est extraite et un modèle de textes est construit avec sa description textuelle. De même temps, le modèle d'image est formé avec Caltech et Flickr «grenouille» des images. En combinant le texte et les indices d'image, des images de pages Web sont classés.

Notre objectif est de récupérer des images d'objets de la page Web bruyant avec des indices d'image et le texte autour de l'image. Hypothèse que l'on fait une requête textuelle avec un mot *q*, par exemple : « *frog* ». On va obtenir un ensemble des web sites du la moteur de recherche Google, par exemple. Le site *i* est représenté comme un paquet $\{W_i, I_i\}$, i=1,..N, où I_i représentant les images du site i et W_i représentant des textes proche I_i du site i. On définit c_i=1 si I_i est l'image pertinente pour la requête textuelle avec le mot *q*, au contraire : c_i=0. On définit aussi θ_t qui est un paramètre de la modèle de texte set θ_v qui est un paramètre de la modèle d'image quand c_i=1 ; θ_b est un paramètre du modèle de textes quand c_i=0. On classe des images en fonction de :

$p(c_i=1| W_i, I_i, q ; \theta_t, \theta_v, \theta_b)$ (1)

Nous adoptons un modèle de texteset un modèle génératif d'image discriminante. Equation (1) est réécrit suivante :

$$\frac{p(W_i \mid c_i = 1, q; \theta_t)p(c_i = 1 \mid I_i, q; \theta_v)}{p(W_i \mid I_i, q)}$$

P (W_i, | I_i, q) est :

$$p(W_i \mid c_i = 1, q; \theta_t)p(c_i = 1 \mid I_i, q; \theta_v) + p(W_i \mid c_i = 0, q; \theta_b)p(c_i = 0 \mid I_i, q) \quad (3)$$

Quand p (c_i, =0 | I_i, q) est égal 1- p (c_i =1 | I_i, q).

Avec θ_t et θ_v s'est entrainé en se basant le texte et l'image dans les ressources de connaissances (Dans ce modèle : Wikipédia pour le texte, Caltech ou Flickr pour l'image). Figure 1 fait un exemple avec une requête textuelle le mot « *frog* » pour illustrer cette approche. Maintenant, on va notretrer comment étudier p (W_i | c_i =1, p ;θ_t) et p (W_i | c_i =0, q ; θ_b) dans *Sec 1.2.1*. p (c_i=1 | I_i, q ; θ_v) est étudié dans *Sec 1.2.2*.

Modèle de texte.

On utilise un modèle de textes général pour le but de trouver la bade de texte d'apprentissage des images. W_i est une suite de mots $\{w_i^j, j= 1,\ldots,L\}$. θ_t est un polynôme de paramètre sur les mots et il est

estimé des ressources de connaissances. Hypothèse que chaque mot dans W_i est indépendant avec les autres dans W_i.

$$p(W_i \mid c_i = 1, q; \theta_t) = \prod_{j=1}^{L} p(w_i^j \mid c_i = 1, q; \theta_t) \qquad (4)$$

Mais équation (4) a tendance de diminuer la contribution de long texte. Autrement dit, avec un paragraphe, l'équation (4) ne satisfait pas. Nous utilisions donc la formule suivante :

$$p(W_i \mid c_i = 1, q; \theta_t) = (\prod_{j=1}^{L} p(w_i^j \mid c_i = 1, q; \theta_t))^{\frac{1}{L}} \qquad (5)$$

Dans cet article, θ_t est évalué suivant :

$$\theta_t^j = \frac{N_K^j + \lambda \eta^j}{N_K + \lambda} \qquad (6)$$

Où K est les ressources de connaissances du texte, qui est considéré la combinaison toutes les pages Wikipédia (le corps du texte uniquement) d'une requête d'une classe d'objets et leurs classes descendantes dans Wikipédia taxonomie, N_K^j est le nombre de *j*ième mot dans K, N_K est le nombre de tous les mots dans K. Similairement, $\eta^j = \frac{N_A^j}{N_A}$. λ est un paramètre de contrôler. Les mots sont mis d'être indépendants et des uniformes probabilités, lorsque c = 0. Alors, la manière de calculer p (W_i $|c_i=0, q ;\theta_b$) comme l'équation (5).

Modèle d'image

Dans ce modèle, les auteurs utilisent l'algorithme SVM (Machines à Support de Vecteurs ou *Support Vecteur Machines* en anglais) pour apprendre directement
p ($c_i=1 \mid I_i, q ;\theta_v$). On utilise les images de chaque classe d'objets demandée dans la base d'images Caltech ou Flickr comme les positifs exemples d'apprentissage. La catégorie « *clutter* » dans Caltech 256 est utilisée comme un négatif exemple. Chaque image est représentée comme un histogramme normalisé avec la dimensionnelle *l*.

Les théories et les algorithmes appliqués dans ce modèle sont très compliqués. L'algorithme et quelques transformations sont réalisés dans cette partie pour apprendre l'image.

2. 1. 3. Points de vue

L'idée que les auteurs ont proposée très originale et très intéressante pour exploiter des ressources de connaissances en ligne pour la recherche d'objet image souhaitées. De nos jours, ces ressources en ligne fournissent des données compilées à l'homme de construire des modèles d'objets. On a effectué des expérimentations sur deux ensembles de données. Les résultats notretrent l'efficacité de cette approche. Bien que les expérimentaux résultats soient bons et clairs, mais les algorithmes et les théories installés dans le système sont complexités.

Cette idée peut être élargit avec n'importe quelle ressource de connaissances et n'importe quelle base de donnée d'images (sans seulement le site Wikipédia et les bases CalTech, Flickr)

2.2 Le deuxième système proposé par Keiji Yanai

2.2.1 Introduction

C'est le système proposé par Keiji Yanai « ***Système de classification des images génériques en utilisant des connaissances visuelles sur le Web*** » [2]. (En anglais, c'est *Generic Image Classification Using Visual Knowledge on the Web*, Proc. of ACM Multimedia 2003, Berkeley USA, pp. 67-76 (2003/11)).
Dans cet article, l'auteur a décrit un système permettant à l'utilisateur de récupérer automatiquement un grand nombre image à partir du Web. Il contient 3 modules : le module de récupération d'images,

le module d'apprentissage d'images et le module de classification d'images. On peut illustrer ce système par la photo suivante :

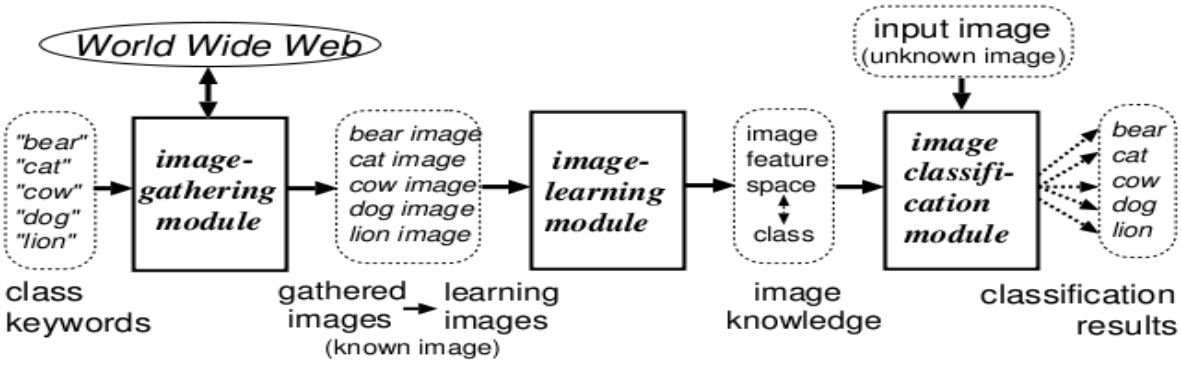


Figure 2 : Structure du système proposé par Keiji Yanai, il est conçu comme un système interagit par 3 modules : un module de récupération d'image, un module d'apprentissage d'image et un module de classification d'image.

Le processus de ce système se compose 3 étapes. Tout d'abord, l'utilisateur fait une requête textuelle avec un mot-clé. Dans le module de récupération d'image, le système va récupérer automatiquement des images en relation avec le mot-clé à partir du Web. Dans la couche prochaine – le module d'apprentissage d'image, les images récupérées ci-dessus sont extraites pour obtenir les caractéristiques et ces derniers sont distribués dans chaque classe. (*Les caractéristiques d'une image sont considérées la contenu visuel d'une image. Ce sont la couleur, la forme, l'histogramme, etc.*) . Enfin, dans le module de classification, le système va classer une image inconnue dans l'une des catégories correspondantes aux classes représentant un mot-clé, en comparant les caractéristiques de cette image inconnue avec ceux de chaque class. Autrement dit, en utilisant l'association entre les caractéristiques d'image et les classes.

2.2.2 La structure du système.

Module de récupération d'images.

Ce module récupère des images à partir du Web représentant un mot-clé. A partir d'une image sur le Web, embarquée dedans un fichier HTML, ce module exploite quelques moteurs de recherche Text-based existants pour récupérer des URL (Universal Ressource Locator) des fichiers HTML en relation avec le mot-clé. L'étape prochaine, en utilisant ces URL, le module va ramener des fichiers HTML à partir du Web, les analyse et évalue l'intensité de la relation entre le mot-clé et les images embarquées dans des fichiers HTML. Si ces images ont la relation du mot-clé, elles sont ramenées à partir du Web. Il dépend de l'intensité de la relation au mot-clé, on divise les images ramenées en deux ensembles : les images dans le groupe A ont la relation forte au mot-clé et les autres dans le groupe B. Pour toutes les images récupérées, ses caractéristiques sont calculées.

Dans les systèmes CBIR, on devra fournir une image ou un croquis au moteur de recherche pour faire une requête parce que ces systèmes se basent la similarité des caractéristiques d'images entre l'image de requête et les images dans la base d'image. Dans ce module de récupération d'image, au lieu de donner une image ou un croquis pour faire une requête, l'utilisateur donne seulement un mot-clé au module de récupération d'image. Ensuite, le système va sélectionner les images ayant la relation forte au mot-clé appelées le groupe A. On affinera ce groupe et les images appartenant au groupe sont considérées les images de sortie (des images de la requête). Avec les images du groupe B, on fait des requêtes comme des requêtes de CBIR, et ajouter des images récupérées aux les images de sortie. Ce processus est décrit par cette figure suivante :

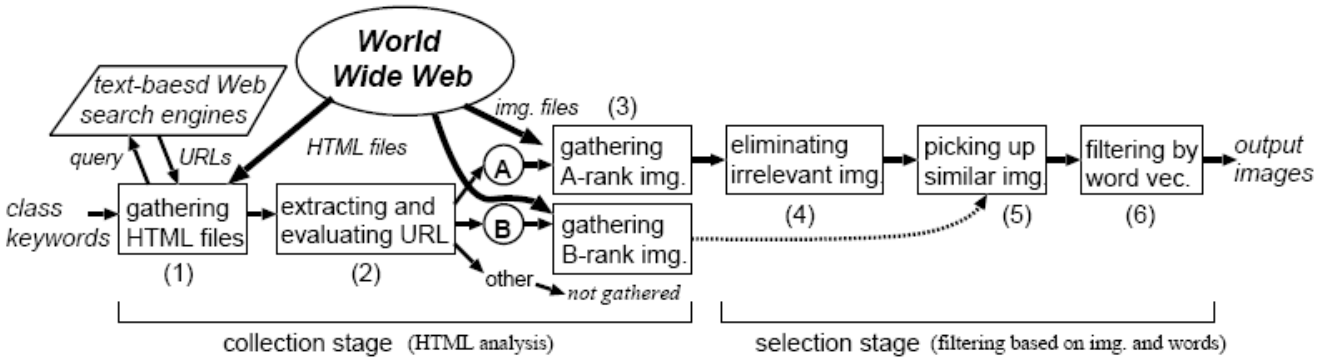


Figure 3 : Processus du module de récupération d'image à partir du Web.
Il se compose de deux étapes : l'étape de collection d'image
Et l'étape de sélection d'image.

A/ Couche de collection

Le détail de l'algorithme de collection d'image suivante :

1. L'utilisateur fournit au système avec deux mots-clés. L'un est considéré de représenter mieux une image souhaitée, l'autre est un mot-clé subordonné. Par exemple : quand on veut récupérer les images « *lion* », on utilise le mot « *lion* » comme le mot-clé principal, et « *animal* » comme un mot-clé subordonné.

2. Le système envoie le mot-clé et le mot-clé subordonné au moteur de recherche d'image pour obtenir des URL des fichiers HTML en relation avec les mots-clés. (*Figure 2 (1)*).

3. Le système ramène des fichiers HTML indiquant par des URL.

4. Le système va analyser des fichiers HTML, et extrait des URL des images embarquées dans fichiers HTML avec les tags embarqués d'image (comme « *IMG SRC* » et « *A HREF* ») (*Figure 2 (2)*). Pour chaque image, le système va calculer une note représentant l'intensité de la relation entre l'image et le mot-clé. Une méthode d'évaluation utilisée est réalisée sur les tags HTML simples. La note est calculée par la manière suivante :

Condition 1 :

Chaque fois de l'une des conditions suivantes satisfaites, 3 points sont ajoutés à la note :

- Si l'image est embarquée par « IMG SRC » tag, le champ « ALT » de ce tag contenant le mot-clé.
- Si l'image est liée directement par le tag « A HREF ». La suite qu'elle se situe entre « A HREF » et « /A » contenant le mot-clé.
- Le nom de l'image contient le mot-clé.

Condition 2 :

Chaque fois de l'une des conditions suivantes satisfaites, 1 point est ajouté à la note :

- Le tag « *TITLE* » du fichier HTML contient le mot-clé.
- Les tags « *H1, …, H6* » du fichier HTML contient le mot-clé.
- Le tag « *TD* » (pour créer les tableaux dans le fichier HTML) contient le tag image-embarquée qui contient le mot-clé.
- Dix mots se situe avant le tag image-embarquée ou dix mots après, ils contiennent le mot-clé.

Enfin, si la note d'une image est supérieur 3, cette image est classée dans le groupe A. Si la note d'une image est supérieur 1 et inférieur 3, elle est classée dans le groupe B. D'autres images sont ignorées. Autrement dit, elles ne sont pas considérées. Le système ramène seulement les images dans le groupe A et le groupe B (*Figure (3)*).

5. Dans ce cas, le fichier HTML ne contient pas tout à fait de tag image-embarquée, le système va ramener et analyser d'autres fichiers HTML liés à ce fichier HTML et les étapes ci-dessus sont implémentées sur ces HTML. La profondeur est 1.

B/ Couche de sélection

Dans cette couche, le système va choisir les images plus appropriées. La sélection se base sur les caractéristiques d'images, et est décrite suivante :

0. Tout d'abord, le système va créer les vecteurs de caractéristiques d'images pour toutes les images dans l'ensemble d'images récupérées dans la dernière couche. Actuellement, le système utilise un histogramme 6 × 6 × 6 couleurs dans la Lu^*v^* espace de couleur.
1. Pour chaque image dans le groupe A, la distance entre deux images est calculée en se basant sur la forme quadratique.
2. La distance entre deux images quelconques des images dans le groupe A, est regroupée par la méthode d'analyse de classification hiérarchique. Ici, le système utilise la méthode de voisin la plus loin *FN* (*the farthest neighbor method*). Au début, chaque groupe a seulement une image, et le système répète la fusion jusqu'à toutes les distances parmi d'eux sont plus que un certain seuil.
3. Il jette de petits groupes qui ont moins d'images d'un certain seuil, les autres sont considérées comme étant sans pertinence. Il stocke toutes les images dans les groupes restantes comme des images de sortie (par la requête textuelle avec le mot-clé) (*Figure 2(4)*).
4. Il sélectionne les images du groupe B dont les distances aux images dans les groupes restantes du groupe A sont petites et les ajoute aux images de sortie (*Figure 2 (5)*).

Dans cette couche, les auteurs présentent aussi une méthode de filtre par le vecteur de mot (*figure 2 (6)).*C'est la méthode « *Latent Semantic Indexing » (LSI)* qui peut extraire le contenu sémantique des documents HTML à partir des vecteurs de mots pour sélectionner des images plus appropriées basé sur des caractéristiques textuelles et visuelles.

Module d'apprentissage et classification d'image.

Premier, le module d'apprentissage va extraire les caractéristiques d'images des images récupérées par le module de récupération d'image et associer (*classer*) ces caractéristiques avec les classes représentant les mots-clés. Ensuite, dans le module de classification d'images, les auteurs classent une image inconnue dans l'une des classes correspondantes pour les mots-clés en comparant les caractéristiques d'images.

Dans ce système, les auteurs utilisent deux genres des caractéristiques d'images pour l'apprentissage et la classification : *la signature de couleur pour le bloc segmentation* et *la signature de région pour la région segmentation* (en anglais *: color signature for block segments and region signature for region segments*). Pour calculer la dissemblance entre deux signatures, Earth Mover's Distance (*EMD*) est proposé. Les auteurs ont décrits deux genres méthodes d'extraire des caractéristiques et de classification en utilisant *EMD* dans deux sections suivantes.

La signature de couleur :

Pour obtenir *la signature de couleur*, d'abord, on normalise la taille des images de formation en 240 × 180, et les diviser en 16 et 9 régions bloc comme le notretre dans *Figure 4*. Nous faisons une signature de couleur pour chacun de ces 25 régions bloc. Ensuite, nous sélectionnons quelques couleurs dominantes par le regroupement des vecteurs de couleur de chaque pixel en grappes de couleur par la méthode *k-means*. Nous faisons une signature de couleur pour chaque bloc avec des éléments comprenant un vecteur de couleur moyenne de chaque cluster et son taux de pixels appartenant à cette grappe. Un vecteur de couleur moyenne est représentée par l'espace de Lu*v* de couleur qui est conçu afin que la distance euclidienne entre deux points dans cet espace.

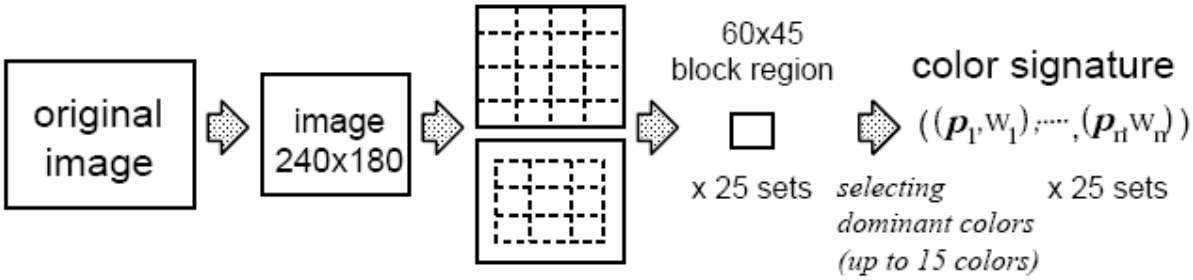


Figure 4 : La signature de couleur pour le bloc de segmentation

Dans le module de classification d'image, d'abord, on une image inconnue, on extrait *la signature de couleur* pour chaque bloc dans cette image en même manière dans la couche d'apprentissage ci-dessus après avoir normalisé la taille de cette image. On obtient 25 ensembles de signature pour l'image inconnue. Ensuite, on cherche tous les blocs des images d'apprentissage de chaque classe pour ramener un bloc ayant la minimum distance à chaque bloc de l'image inconnue. Ici, la distance est calculée par *EMD*. Pour continuer, on groupe les minimums distances entre l'image inconnue et les images d'apprentissage de chaque classe sur 25 blocs. Cette recherche et ce calcul sont effectués pour tous les classes. Nous comparons le total de distances entre tous les classes, et nous classons l'image inconnue dans la classe dont le total de distance est la plus petite.

La signature de région :
Pour obtenir la signature de région, on segmente les images et divise en segmentations blocs après avoir normalisé leur taille la notretre dans *Figure 5*. Dans cet article, les auteurs emploient une méthode simple de segmentation en se basant *k-means clustering* et une autre, c'est une méthode de segmentation de couleur, appelée *JSEG*. Ensuite, en utilisant *k-means* ou *JSEG*, on va récupérer un vecteur de caractéristiques de neuf-dimensionnelle. Finalement, on obtient une signature de région sur image.

Dans la couche de classification, on emploie la méthode *k-ANN (the k-nearest neighbor)* pour classer une image inconnue entrée à une certaine classe.

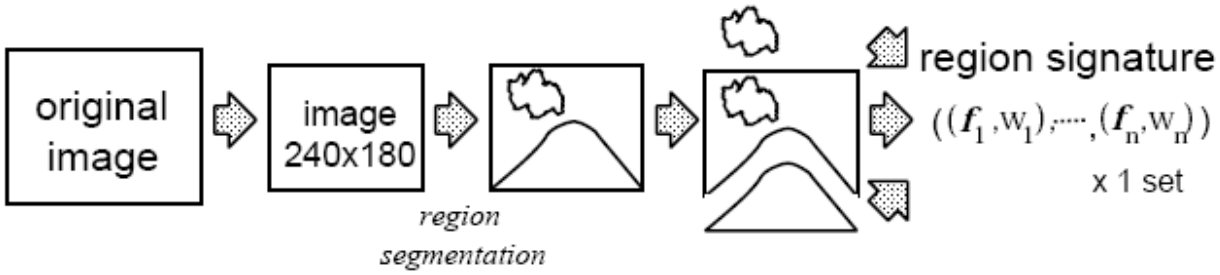


Figure 5 : La signature de région pour la région segmentation.

2.2.3. Points de vue

D'abord, le processus de collectionner et de sélectionner proposé par l'auteur est vraiment original bien qu'il a l'air complexité un peur et il peut avoir des limites.

Comme ce qui est présenté ci – dessus, le système utilise deux caractéristiques d'image pour apprendre et classer les images retournées dans les classes. Ce sont la signature de couleur et la signature de région. D'après moi, ce sont de bons choix parce que la couleur et la région sont deux factures très importantes de caractéristiques visuelles d'images. De plus, le traitement se basant sur ces deux genres demande peu de temps et de calculs. C'est pourquoi pour lesquelles ce système fonctionne vraiment efficace sur des ensembles sélectionnés du Web.

Dans les expérimentaux, les résultats de classer ne sont pas élevés (dans les tables 2,3 et 5 sont inférieurs 30%). Je pense que le problème émerge quand le système utilisait l'algorithme EMD pour calculer la similarité entre les caractéristiques d'images.

D'autre part, ce système nous fournissait une méthode pour extraire des contenus textuels d'images comme (i) le nom de fichier d'image, (ii) le titre de page qui contient l'image que nous recherchons, (iii) ALT-tag d'image. Ces caractéristiques sont facilement extraites et elles ont tendance à donner une description la plus exacte d'image intéressée. Cependant, on trouve que, ces caractéristiques ne donnent pas souvent les informations suffisantes sur une image. Le nom d'image est souvent en abrégé et peut – être il n'est pas identifié comme des lettres significatives. Le titre de page peut être trop

général, une page peut avoir plus d'une image dans une page web. En outre, un grand nombre d'images n'ont pas encore des textes alternatifs. En effet, nous récupérons seulement 21% des images qui ont les textes alternatifs.[7]

En fin, avec le système ci-dessus, il nous donne un grand nombre des HTML de pages web. Cependant, on ne sait pas quels HTML qui contiennent des images pertinentes et on ne peut pas chercher dans tous ces HTML.

2.3 Système proposé par Yiming Liu, Dong Xu, Ivor et Luo [5]

Dans ce système, d'abord, l'utilisateur fournit un mot de la requête textuelle. En se basant sur des images Web pertinentes et non-pertinentes récupérées automatiquement, on applique deux méthodes de classification : *k Nearest Neighbor (kNN)* et la décision de souches (*decision stumps*. Et puis, on emploie ces classificateurs pour les photos des consommateurs, et les classer avec la réponse des classificateurs. L'utilisateur peut également fournir des commentaires pertinents pour affiner les résultats de récupération. *Figure 6* ci-dessous notretre la structure du système.

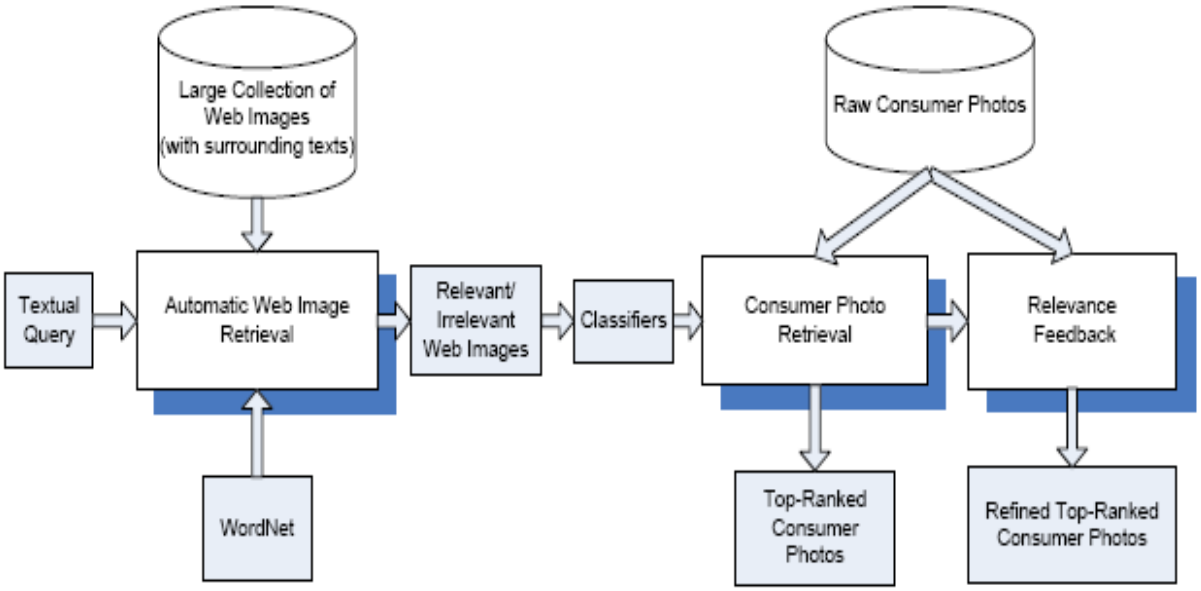
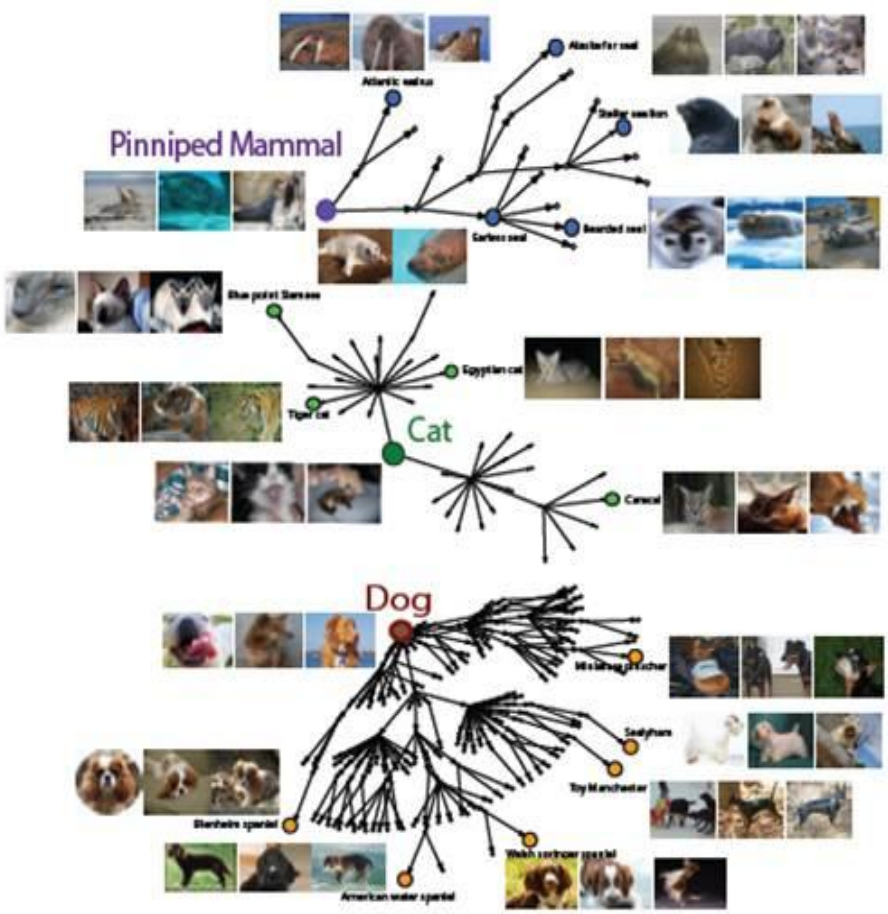


Figure 6 : La structure du système proposé par Yiming Liu, Dong Xu, Ivor et Luo

2.4 ImageNet [6]

- En se basant l'article *ImageNet: A large-scale hierarchical image database*. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li and Li Fei-Fei . CVPR 2009
- Il marche de se baser sur la structure de WordNet.



ImageNet réserve les chercheurs, les enseignements, les étudiants et toutes les personnes se passionnant pour l'image. Il est considéré une base d'image pour la recherche et l'apprentissage. Il est conçu en reposant sur l'ossature de la Structure de WordNet. Il nous permet seulement de chercher des images avec un seul mot-clé en ***anglais*** ayant une sémantique précise par exemple : « *sky* », « *elephant* ». Jusqu'à présent, on peut utiliser seulement des noms comme des mots-clés. Il nous donne des images les plus pertinentes (selon les auteurs). Enfin, la base de donnée d'image est assez de limite, elle concerne l'animal, la véhicule, les fruits, les fleurs.

2.5 Solution proposée

Le but principal de ce TPE est de récupérer des images dans des bases d'images hors ligne stockées dans les équipements numériques sur des requêtes textuelles. L'hypothèse qu'il n'y a pas de texte (pages web, annotations, …) accompagnant les images. Dans le cadre du TPE, on va faire des requêtes textuelles sur des bases d'images hors linge comme CalTech 101/256, Wang. Dans le système que l'on propose, au début, on fera une requête textuelle par un mot quelconque. Ensuite, ce mot sera envoyé à un moteur de recherche d'image comme Google Image, Bing, Yahoo. Puis, on va récupérer un ensemble d'images. A partir des images résultats, on va choisir les images pertinences pour construire un ensemble de caractéristiques d'images représentant le mot-clé. Enfin, ces caractéristiques seront utilisées pour rechercher dans les bases d'images qui nous intéressent comme les bases d'images CalTech 101/256, Wang, etc.

Notre système peut être décrit comme le schéma suivant :

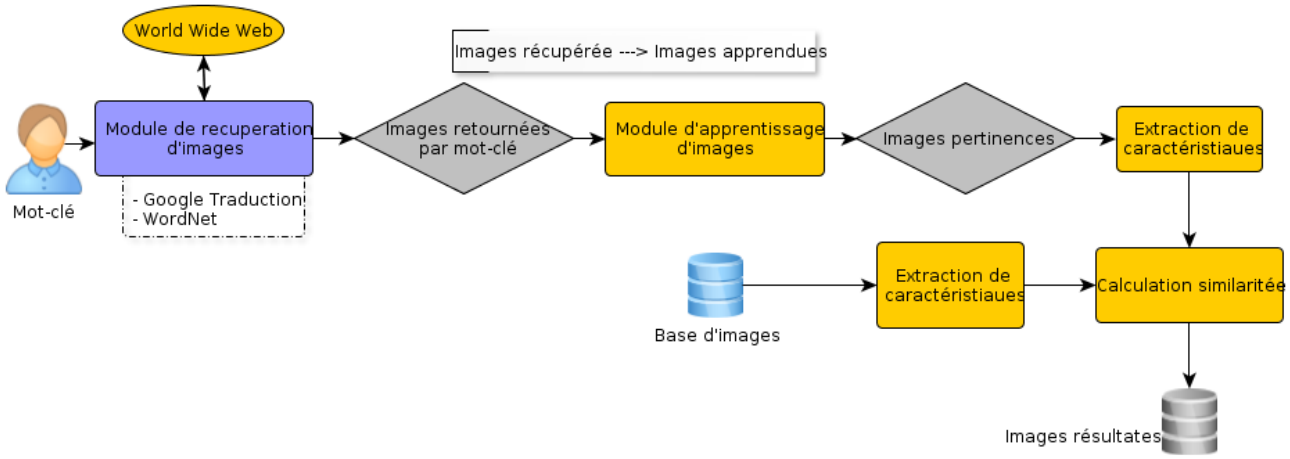


Figure 1 : Recherche d'image par le contenu sur des requêtes textuelles.

2.5.1 **Modèle de texte** : Dans ce système, au début, l'utilisateur fait une requête textuelle, il existe deux cas de mot-clé : le mot-clé contenant un mot et le mot-clé contenant supérieur un mot :

- *Le cas mot-clé contenant un mot* : on va utiliser un modèle appelé « modèle de texte » pour traiter le mot-clé avant d'envoyer à un moteur de recherche. Le modèle de texte sera divisé en deux modèles : modèle de langage et modèle de WordNet. Pour le modèle de langage, ce mot-clé va envoyer à l'outil Google Traduction pour récupérer un mot ayant la même signification en français et en chinois. Alors, avec le modèle de langage, on va obtenir 3 mots-clés à partir le mot initial.

Semblablement, pour le modèle de WordNet, on peut prélever de WordNet des mots ayant la même signification.

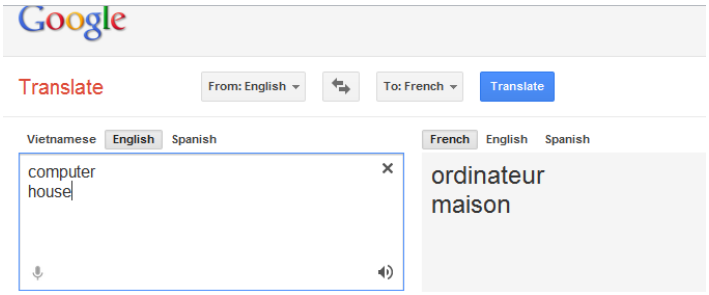


Figure 2 : Google Traduction

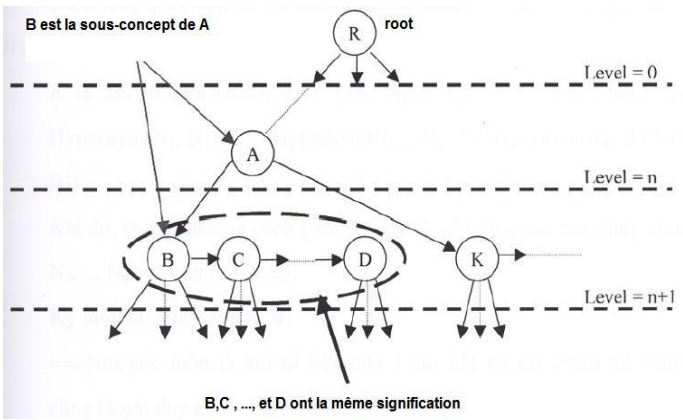


Figure 3 : Structure de WordNet

- *Le cas mot-clé contenant supérieur un mot* : ce mot-clé va envoyer directement à un moteur de recherche d'images.

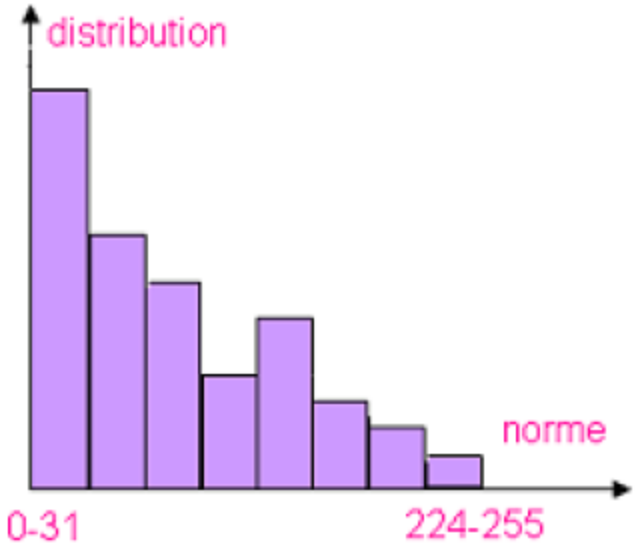
Tous les mots-clés obtenus à partir du modèle de texte seront envoyé à un moteur de recherche d'images comme Google Image, Bing, Yahoo.

2.5.2 Modèle d'apprentissage : On sait que l'ensemble d'images retournées du moteur de recherche d'images contiennent des images pertinences et aussi des images non-pertinences. Dans ce modèle, on va utiliser une description d'une méthode de classification par apprentissage particulière, Les *Machine à vecteurs de support* – les SVM pour classer l'ensemble retournées. Les SVM sont un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes de discrimination de régression. Les SVM sont une généralisation des classifieurs linéaires.

2.5.3 Extraction de caractéristiques

Dans notre système, on choisit la couleur avec des composants : l'intersection d'histogramme, les moments des couleurs, la texture et les moments de Hu.

a) Intersection d'histogramme : Une technique très utilisée pour la couleur est l'intersection d'histogrammes. Avec cette méthode, tout d'abord, on doit calculer l'histogramme d'image. L'histogramme d'une image peut être présenté par un vecteur dont chaque composant est un nombre de pixels de couleur correspondant à son indice. Ensuite, on crée un histobin à partir de l'histogramme de la manière suivante : chaque bin (trou) dans l'histobin est la somme de quelques éléments voisins de l'histogramme. Le nombre de voisins est déterminé par le nombre de bin de l'histobin. On peut voir que l'histobin est plus compact que l'histogramme.



Exemple d'un histobin

Dans notre système, on calcule l'histobin pour composant couleur R, V, B dans l'espace RVB. Après avoir eu l'histobin d'image, la distance entre deux images devient la distance entre deux histobins :

$$D_{Histo}(H,I) = \frac{\sum_i |H_{Histo\ i} - I_{Histo\ i}|}{\sum_i H_{Histo\ i}}$$

Où : H, I : les deux images

- $D_{Histo}(H,I)$: La distance entre l'image H et l'image I en fonction de l'intersection d'histogrammes
- $H_{Histo\ i}$: Le bin i de l'histobin de H
- $I_{Histo\ i}$: Le bin i de l'histobin de I

b) Moments des couleurs : notre système utilise les moments des couleurs : L'espérance (μi), la variance (σi), le moment de troisième ordre (si)

$$\mu_i = \frac{1}{N}\sum_{j=1}^{N} f_{ij} \qquad \sigma_i = \left(\frac{1}{N}\sum_{j=1}^{N}(f_{ij} - \mu_i)^2\right)^{1/2} \qquad s_i = \left(\frac{1}{N}\sum_{j=1}^{N}(f_{ij} - \mu_i)^3\right)^{1/3}$$

Où f_{ij} est la valeur du $i^{ième}$ component de couleur du pixel j et N est le nombre de pixels dans l'image. La distance entre l'image H et l'image I :

$$D_{Moment}(H,I) = \sum_i w_{i1}|\mu_{Hi} - \mu_{Ii}| + w_{i2}|\sigma_{Hi} - \sigma_{Ii}| + w_{i3}|s_{Hi} - s_{Ii}|$$

Où w_{ij} sont les poids correspondants à l'espérance, la variance et le moment de troisième ordre (j = {1, 2, 3}).

c) Texture : La matrice de cooccurrences de niveaux de gris P (g, g') compte les nombres de paires de pixel (m, n) et (m', n') dans une image qui ont une valeur d'intensité g et g' avec une distance d dans une direction α.

Exemple : une région de 15 pixels quantifiés sur 8 niveaux de gris

1	2	1	2	4
2	3	1	2	4
3	3	2	1	1

Matrice de cooccurrences avec d = 1 et α = 0 :

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0	0	0	0	0	0	0	0
1	0	1	2	0	0	0	0	0
2	0	1	0	2	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	1	0	0	1	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0

Il y a 14 caractéristiques statistiques extraites à partir de cette matrice. Mais, ici, dans notre méthode, seulement les 4 caractéristiques les plus appropriées sont utilisées : L'énergie, l'entropie, le contraste et le moment inverse de différence.

- L'énergie T1 : $\sum_i \sum_j P_d^2(i,j)$
- L'entropie T2 : $-\sum_i \sum_j P_d(i,j) log P_d(i,j)$
- Le contraste T3 : $\sum_i \sum_j (i-j)^2 P_d(i,j)$
- Le moment inverse de différence T4 : $\sum_i \sum_j \frac{P_d(i,j)}{(i-j)^2}; i \neq j$

La distance d ici, dans toutes les directions est toujours 1. La distance entre deux images H, I :

$$D_{Texture}(H,I) = \sqrt{(T_{H1} - T_{I1})^2 + (T_{H2} - T_{I2})^2 + (T_{H3} - T_{I3})^2 + (T_{H4} - T_{I4})^2}$$

La distance entre deux images est alors :

$$D(H,I) = W_{HistoRVB} * D_{HistoRVB}(H,I) + W_{MomentRVB} * D_{MomentRVB}(H,I) + W_{Texture} * D_{Texture}(H,I)$$

Où $W_{HistoRVB}$,$W_{MomentRVB}$ $W_{Texture}$ sont les poids correspondants à l'intersection d'histogrammes RVB, la distance entre les moments RVB, la distance entre les textures. Pour faciliter le calcul, je choisis pour chaque poids 1/N où N est le nombre de caractéristiques. Dans ce cas, N = 3.

d) **Les moments de Hu** : 7 valeurs des moments de Hu sont calculées grâce aux formulaires suivants :

$$I_1 = \eta_{20} + \eta_{02}$$
$$I_2 = (\eta_{20} - \eta_{02})^2 + (2\eta_{11})^2$$
$$I_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$
$$I_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$
$$I_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$
$$I_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$
$$I_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2].$$

Où :

$$\eta_{ij} = \frac{\mu_{ij}}{\mu_{00}^{\left(1+\frac{i+j}{2}\right)}}$$

Et :

$$\mu_{00} = M_{00},$$
$$\mu_{01} = 0,$$
$$\mu_{10} = 0,$$
$$\mu_{11} = M_{11} - \bar{x}M_{01} = M_{11} - \bar{y}M_{10},$$
$$\mu_{20} = M_{20} - \bar{x}M_{10},$$
$$\mu_{02} = M_{02} - \bar{y}M_{01},$$
$$\mu_{21} = M_{21} - 2\bar{x}M_{11} - \bar{y}M_{20} + 2\bar{x}^2M_{01},$$
$$\mu_{12} = M_{12} - 2\bar{y}M_{11} - \bar{x}M_{02} + 2\bar{y}^2M_{10},$$
$$\mu_{30} = M_{30} - 3\bar{x}M_{20} + 2\bar{x}^2M_{10},$$
$$\mu_{03} = M_{03} - 3\bar{y}M_{02} + 2\bar{y}^2M_{01}.$$

M_{ij} sont des moments centraux. Ensuite, la distance entre deux images est calculée grâce au formulaire Euclid.

e) **Calcul de similarité [10] :** Dans notre système, on utilise la fonction « *Minkowski-Form distance* ». L'idée de *Minkowski-Form Distance*, c'est que chaque image qui a N caractéristiques est un point dans l'espace de N dimension. Chaque caractéristique est un vecteur f(i) dans cette espace. La distance entre deux images, c'est la distance entre deux points dans cette espace.

$$D(I,J) = \left(\sum_{i=1}^{N} |f_i(I) - f_i(J)|^p\right)^{1/p}$$

Où : D (I, J) : La distance entre l'image I et l'image J

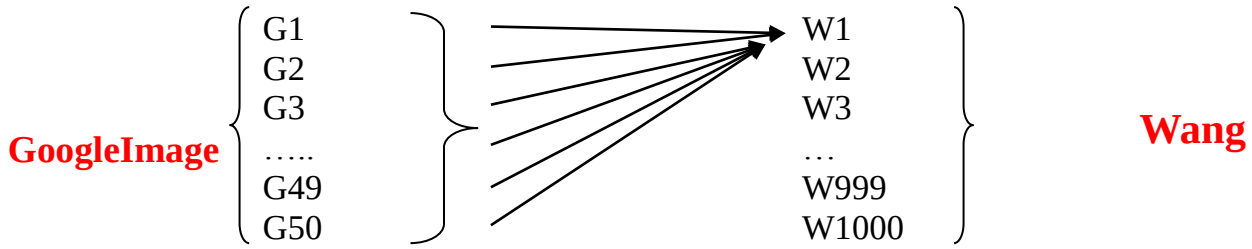
$f_i(I)$: Le vecteur de la caractéristique i de l'image I

$f_i(J)$: Le vecteur de la caractéristique i de l'image I

p = 1, 2 ou ∞

2.5. 4 Algorithme

1. Traitement le mot-clé par le modèle de texte. Des images retournées sont stockées dans un répertoire disque.
2. Appliquant la méthode SVM *one-class* pour classer l'ensemble d'images. Après cette étape, on a un ensemble d'images pertinences, appelées A.
3. Extractions de caractéristiques d'images dans cet ensemble et dans la base d'image locale.
4. Calcul de distance entre toutes les images I_i dans A et une image L_1 dans la base d'images locales. Ensuite, fait le somme en moyenne, on va obtenir la distance D_1 entre l'ensemble A et l'image L_1. De même façon, on va obtenir D_k qui est la distance entre A et l'image L_k.
5. Appliquer QuickSort pour *k* distances ci-dessus, on va obtenir des images qu'on a besoin



Par exemple : *GoogleImage* a 50 images G1, G2, … , G50. *Wang* a 1000 images. D_1 qui est la distance entre GoogleImage et W1 est calculé comme suivante :

$$D_1 = \frac{1}{50} * \{distance\ (G1, W1) + distance\ (G2, W1) + \ldots distance\ (G50, W1)\ \}$$

2.5.5 Méthode d'évaluer le système

Avant l'exécution d'un système de recherche d'informations, on doit mesurer la performance de ce système. Les mesures les plus courantes sont : le temps de réponse une requête et l'espace utilisée. Dans le système de recherche d'images, on a deux autres mesures courantes : le rappel et la précision.

- 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 »

$$Rappel = \frac{|Ra|}{|R|}$$

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

$$Précision = \frac{|Ra|}{|A|}$$

Où :

R : L'ensemble d'images pertinentes dans la base d'images utilisée pour évaluer

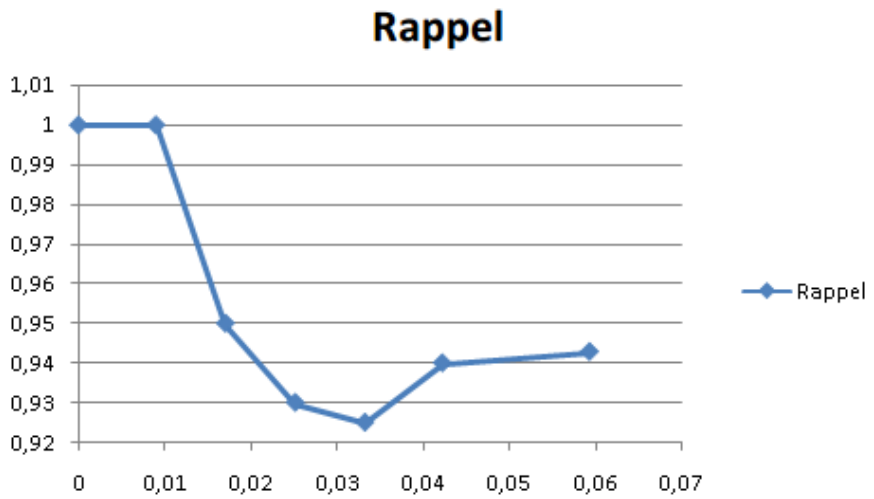
|R| : Le nombre d'images pertinentes dans la base d'images

A : L'ensemble des réponses

|A| : Le nombre d'images dans l'ensemble des réponses
|Ra| : Le nombre d'images pertinentes dans l'ensemble des réponses.

Le rappel et la précision sont très utiles parce qu'ils nous permettent d'évaluer quantitativement la qualité de la réponse globale et la largeur de l'algorithme de recherche. Mais ils ont 4 désavantages. *Premièrement*, l'estimation de la valeur maximum de rappel exige de savoir toutes les connaissances de la base d'images. Quand la base d'images devient de plus en plus grande, ces connaissances ne sont pas disponibles, ce qui veut dire que l'évaluation n'est pas bien estimée. *Deuxièmement*, le rappel et la précision sont reliés. Donc dans quelques cas, les deux mesures ne sont pas suffisantes. Dans ces cas, l'utilisation d'autres mesures qui combinent le rappel et précision pourrait être plus appropriée. *Troisièmement*, ces mesures travaillent bien sur un ensemble de requêtes par lots (non-interactif). Cependant, les systèmes modernes ne travaillent pas dans ce mode. Ils permettent de communiquer avec l'utilisateur donc dans ces systèmes d'autres mesures qui sont plus appropriées doivent être utilisées. *Quatrièmement*, le rappel et la précision sont faciles à définir quand l'ordre des images est linéaire. Ces mesures ne sont pas appropriées pour les systèmes qui ont un ordre faible.

La courbe de rappel et précision :
Le rappel et la précision sont les mesures importantes, mais si on voit seulement une paire de valeurs de rappel et précision, cette paire de valeurs ne peut pas indiquer la performance du système. C'est pourquoi on donne souvent une distribution de rappel et précision sous en forme de courbe. La figure ci-dessous donne un exemple de courbe de rappel et précision. Pour dessiner cette courbe, on doit calculer plusieurs paires de rappel et précision et les interpoler.



Les outils utilisés :

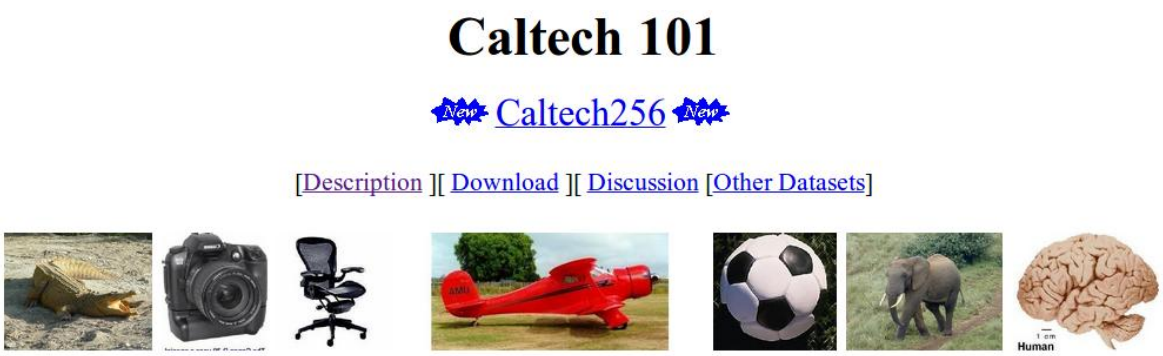
OpenCV (pour Open Computer Vision) est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel. La bibliothèque OpenCV met à disposition de nombreuses fonctionnalités très diversifiées permettant de créer des programmes partant des données brutes pour aller jusqu'à la création d'interfaces graphiques basiques. Elle propose la plupart des opérations classiques en traitement bas niveau des images : la lecture ; l'écriture et l'affichage d'une image ; le calcul de l'histogramme des niveaux de gris ou d'histogramme couleurs ; le lissage ; le filtrage ; la segmentation, etc. Particulièrement, des certains algorithmes classiques dans le domaine de l'apprentissage artificiel sont disponibles : *K-mean, Machine à vecteurs de support (SVM)*, etc. Elle est gratuite et disponible à l'adresse :

http://opencv.willowgarage.com/wiki/ . Dans ce TPE, on va utiliser des méthodes en relation avec l'algorithme SVM, l'histogramme, etc.

Qt est une bibliothèque logicielle orientée objet et développée en C++ par la société Qt Software. Elle offre des composants d'interface graphique (widgets), d'accès aux données, de connexions réseaux, de gestion des fils d'exécution, d'analyse XML, etc. Qt permet la portabilité des applications qui n'utilisent que ces composants par simple recompilation du code source. Les environnements supportés sont les Unix (dont Linux) qui utilisent le système graphique X Window System, Windows et Mac OS X. Qt est notamment connu pour être la bibliothèque sur laquelle repose l'environnement graphique KDE, l'un des environnements de bureau les plus utilisés dans le monde Linux. Cet outil est disponible à l'adresse : http://www.qtsoftware.com/ . Dans ce TPE, on va utiliser cet outil pour développer notre programme.

Les bases d'images utilisées :

Caltech 101 ou Caltech 256 : Caltech 101 contient des images de 101 objets collectées par Fei-Fei Li, Marco Adreetto et Marc Aurolio Ranzato. Avec chaque objet, de 40 à 800 images ont été prises. Chaque image a une taille de 300 x 200 pixels. Ces images sont disponibles à l'adresse : http://www.vision.caltech.edu/Image_Datasets/Caltech101
Caltech 256 est semblable mais elle contient des images de 256 objets d'images (environ 30.000 images). On peut télécharger au site :
http://www.vision.caltech.edu/Image_Datasets/Caltech256/



La base de Wang : La base d'images de Wang est un sous-ensemble de la base d'images Corel. Cette base d'images contient 1000 images naturelles en couleurs. Ces images ont été divisées en 10 classes, chaque classe contient 100 images. L'avantage de cette base est de pouvoir évaluer les résultats. Chaque image dans cette base d'images a une taille de 384 x 256 pixels ou 256 x 384 pixels. Ces images sont disponibles à l'adresse : http://wang.ist.psu.edu/docs/related.shtml .

Chapitre III – Travail pratique

3.1 Réalisation pratique

3.1.1 Indexation des images

Dans l'étape d'indexation des images, le système calcule les caractéristiques comme l'histogramme, les moments de couleur, la texture, les moments de Hu de tous les images dans la base d'images. On va utiliser la base d'image Wang et quelques sous-ensembles de la base d'image Caltech101. Après calculer les caractéristiques, le système écrit ces informations sur un fichier texte s'appelle « **dataWang.txt** » avec la structure suivante :

Numéro d'identification de l'image	Le chemin vers le fichier de l'image	Caractéristiques de l'histogramme (48 valeurs)	Caractéristiques du moment de couleur (9 valeurs)	Caractéristiques de la texture (4 valeurs)	Caractéristiques du moment de Hu (7 valeurs)

Chaque ligne dans le fichier « **dataWang.txt** » représente les informations des caractéristiques. Par exemple, cette ligne suivante représente les informations des caractéristique de l'image « 1**.jpg** »

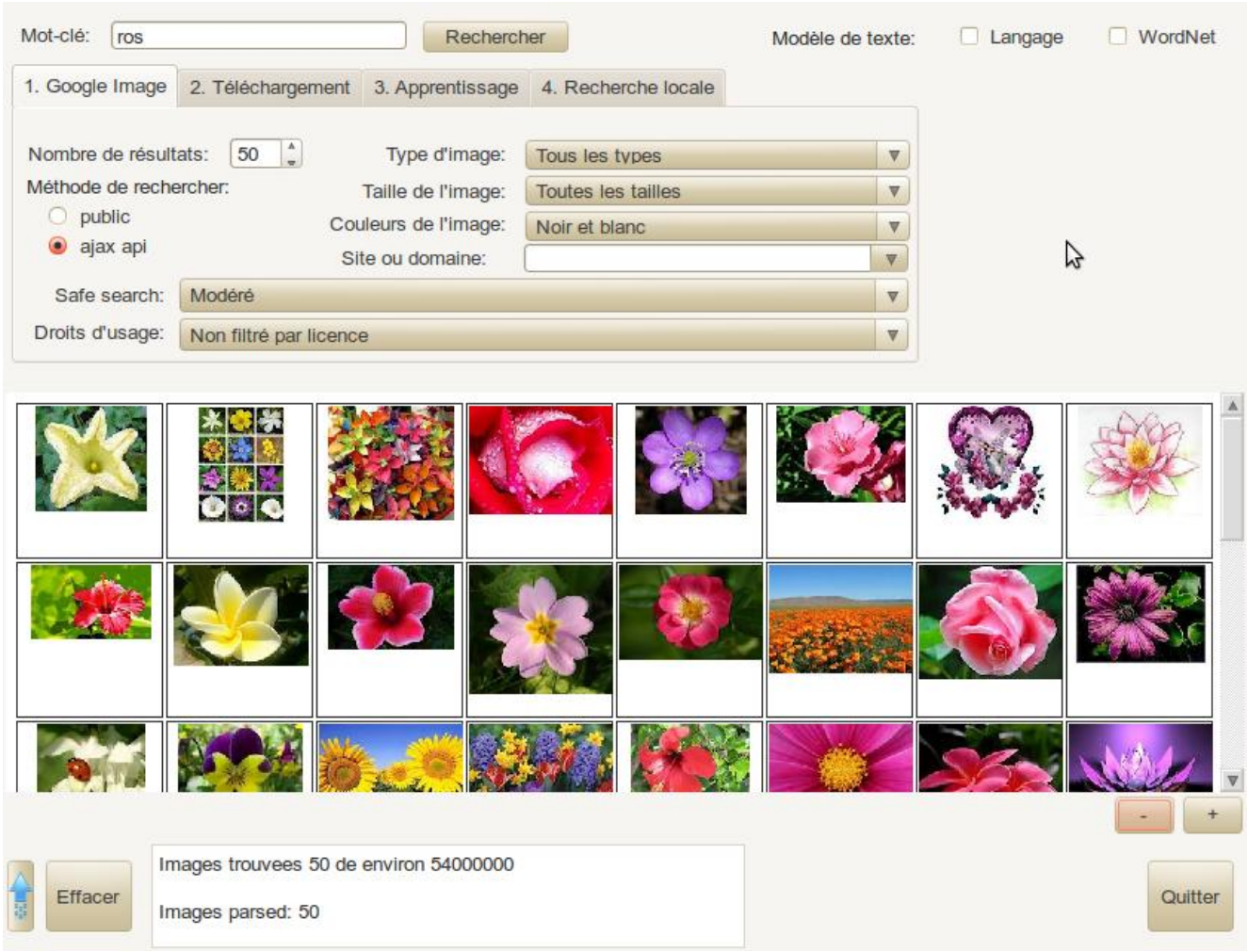
1 /home/clairsang/Bureau/1.jpg 0.0666402 0.123474 0.102153 0.132253 0.133657 0.109294 0.0827637 0.067454 0.0547282 0.0493978 0.0335286 0.0224202 0.0112712 0.00783285 0.00263468 0.000498454 0.0761108 0.118235 0.0761108 0.0727234 0.111582 0.113597 0.116038 0.110016 0.0712687 0.0434977 0.0324198 0.0233765 0.0155741 0.0142212 0.00511678 0.000111898 0.0346578 0.0736287 0.0728353 0.0668742 0.0732218 0.106506 0.101349 0.084198 0.0798543 0.0749308 0.0506185 0.0522868 0.0455933 0.0420532 0.0339966 0.00739543 79.8425 86.9203 111.827 49.306 51.6066 61.0311 42.1921 37.4367 37.9444 8.97605 2.86465 0.550077 14.8379 0.000773271 9.60746e-08 6.68515e-14 9.63125e-14 2.63503e-27 1.56213e-17 -7.26512e-27

La base d'images Wang contient 1000 images. Donc, le fichier « dataWang.txt » a 1000 lignes.

On va utiliser 6 sous-ensembles de la base d'images Caltech 101 : sunflower (85 images), panda (38 images) , elephant (64 images), chair (62 images), dophin (65 images) et airplanes (800 images) . Le fichier « CalTech.txt » a 1114 lignes.

3.1.2 Récupération des images à partir de GoogleImage

Après l'étape d'indexation des images dans la base d'images, on va construire le système automatique de recherche d'images par le contenu basée sur des requêtes textuelles. Dans ce cas, on envoie un mot-clé « rose » à GoogleImage :



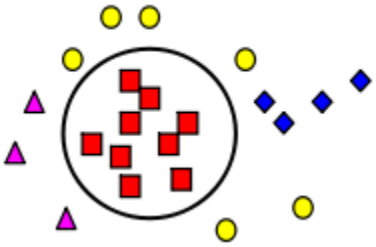
Ensuite, on va télécharger des images vers un répertoire :

3.1.3 L'apprentissage des images

Dans la partie de solution, on a abordé une méthode d'apprentissage des images : SMV one-class. Le but principal de cette étape est de trouver des images pertinentes dans des images qu'on a obtenues à partir de *GoogleImage*. En condition du temps, dans le cadre de ce TPE, on ne va pas implémenter cette méthode d'apprentissage. Mais on peut donner quelques particularités de SVM one-class.

La méthode SVM générale est une méthode d'apprentissage supervision mais SVM one-class est une méthode d'apprentissage sans supervision. SVM one-class est abordé la première

fois dans l'article « *Estimating the Support of a High-Dimensional Distribution* » par Bernhard Scholkopf. Exactement, SVM one-class est utilisé pour faire l'estimation de distribution. Dans le domaine de recherche d'images, on va considérer des images pertinentes qui sont des images les plus proches. C'est –à-dire qu'on va déterminer un hyperplan (déterminer des paramètres) pour des points sont dans une circle avec la radius R (peut-être, c'est une éclipse fermée). Dans ce cas, un point représente une image dans une espace n-dimension. D'autres points (d'autres images) ne s'intéressent pas.



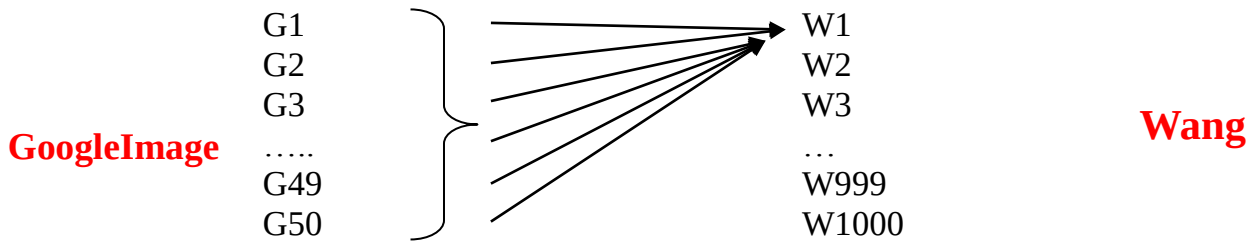
Méthode SVM one-class.

Le grand problème quand on applique SVM one-class est de choisir des paramètres pour trouver la meilleure hyperplane.

3.1.4 Calcul similarité

On n'utilise pas la méthode d'apprentissage et de plus, on reconnait que des premières images à partir de GoogleImage sont souvent des images pertinentes. C'est bien. Dans notre programme, on va utiliser ***vingt premières images*** comme les entrées pour faire calcul similarité. D'abord, on doit extraire des caractéristiques de ces 20 images et sauvegarder dans un fichier « *GoogoleImage.txt* ».

Le calcul similarité entre des images obtenues à partir de GoogleImage et Wang/CalTech se base sur des fichiers textuels : *GoogleImage.txt* et *dataWang.txt/CalTech.txt*



Par exemple : *GoogleImage* a 50 images G1, G2, … , G50. *Wang* a 1000 images. D_1 qui est la distance entre GoogleImage et W1 est calculé comme suivante :

$$D_1 = \frac{1}{50} * \{distance\ (G1, W1) + distance\ (G2, W1) + \ldots distance\ (G50, W1)\ \}$$

Enfin, on va utiliser QuickSort pour arranger 1000 D_i .

Exemple :

Le mot-clé : ***airplanes***

20 images obtenues à partir de GooogleImage :

Ce sont des images locales trouvées dans CalTech101. On peut voir qu'il y a quelques images non-pertinentes. Pour représenter les résultats de la recherche, l'objectif est de donner à l'utilisateur une interface dynamique et rapide. Dans l'interface que j'ai utilisée, les images trouvées sont affichées en une table 2D en ordre de gauche à droite et de bout en bout avec des 50 premières images.

3.2 Résultat et Analyse

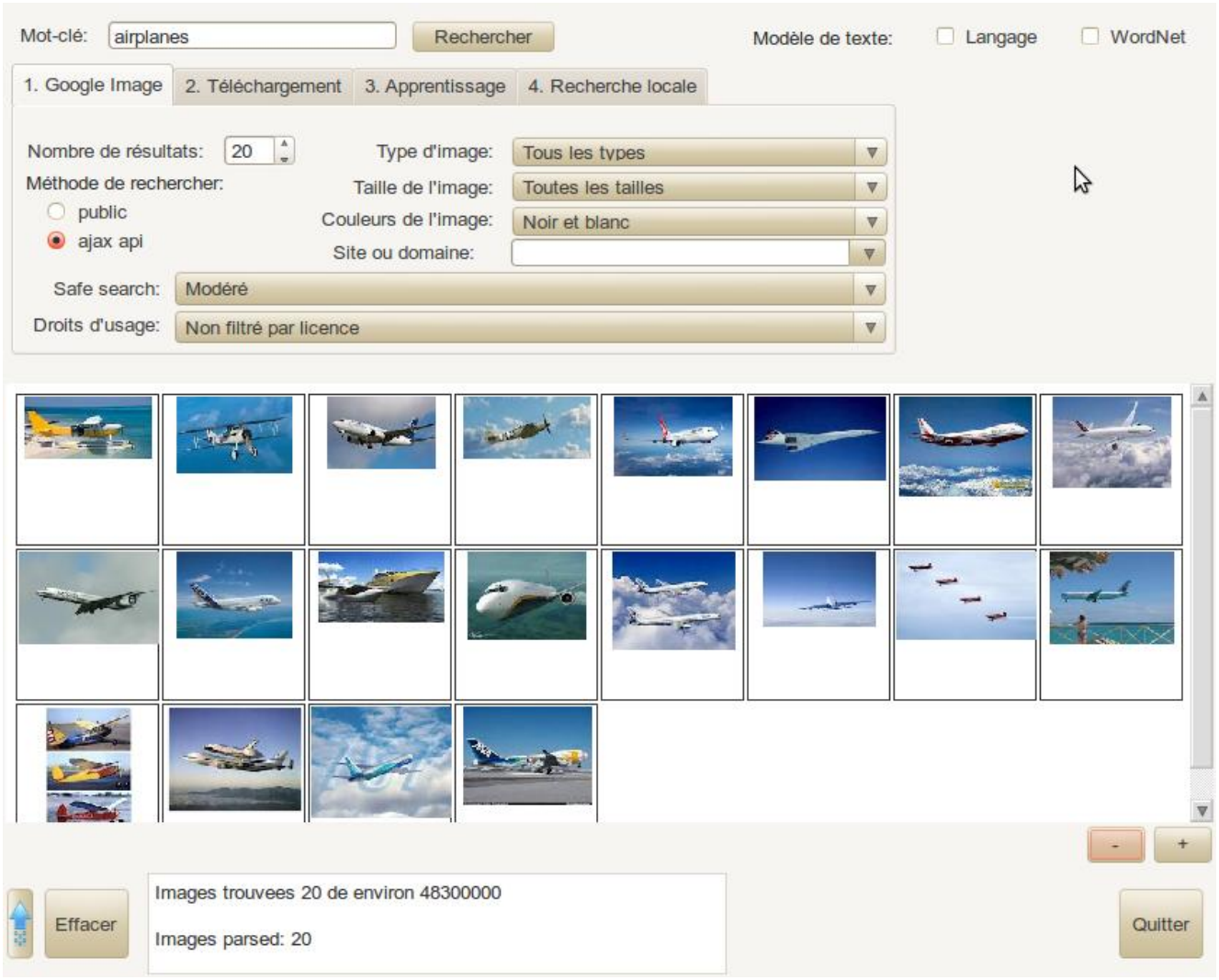
Pour évaluer la qualité de la réponse du système avec une requête, j'utilise la méthode suivante :

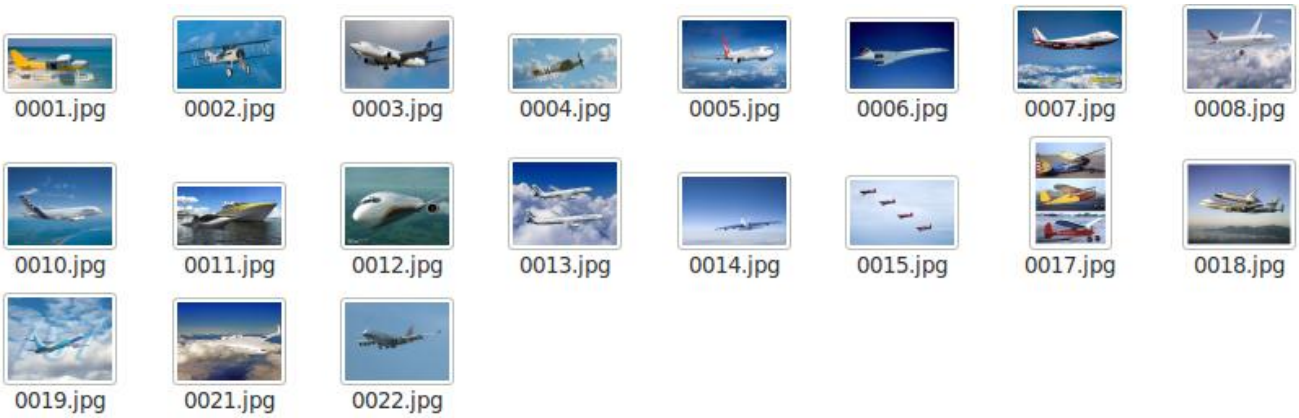
- Prend une requête textuelle qui correspond à un nom de la classe d'images de Caltech 101.
- Le module de récupérer nous permet d'obtenir 20 premières images à partir de GoogleImage.

- Ensuite, le système va donner les résultats comme une liste des images ordonnées par la similarité entre les images de Caltech 101 et 20 images obtenues à partir de GoogleImage.
- On reçoit n premières images dans la liste donnée par le système. Dans ces n images, on compte les images pertinentes. Ensuite, on calcule le rappel et la précision.
- Pour n = 1 à 1114 (on utilise seulement 6 sous-ensembles de CalTech101), on a 1114 pairs de rappel et précision. On faire alors la courbe de rappel et précision.

Dans les n premières images reçues, on peut compter les images pertinentes de façon automatique grâce à l'organisation de la base de Caltech 101. Cette base d'images contient 9145 images. Ces images ont été divisées en 101 classes. Une image est pertinente si elle a le même nom avec le mot-clé (requête textuelle donnée par l'utilisateur).

Exemple 1 : Requête « ***airplanes*** » (800 images dans 1114 images)



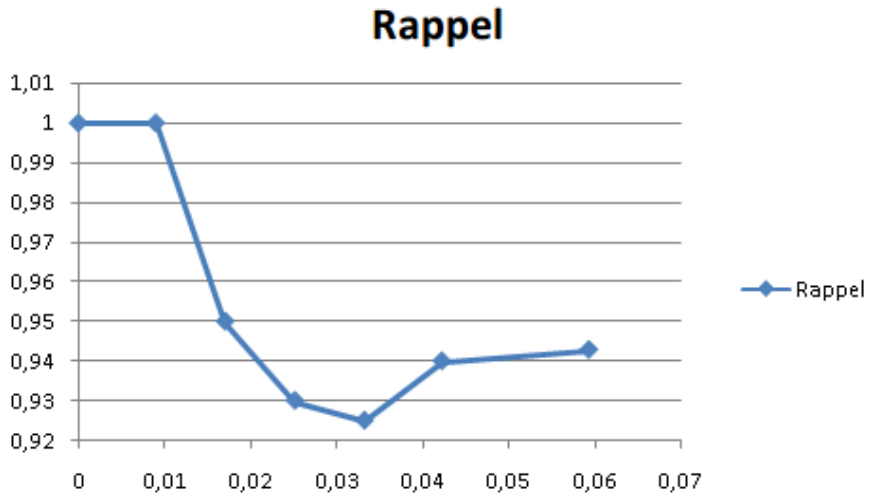


(*Quelques images ne sont pas sauvegardées, donc pour récupérer 20 images, on choisit souvent le nombre d'images sur l'interface graphique qui est de 22.)*

Ce sont 20 images de CalTech101. On peut voir une image non-pertinente. Cette dernière a le fond bleu comme le ciel.

Cette figure représente les images de résultats. A partir des images dans le groupe d'images pertinentes de *GoogleImages*, on peut voir que la couleur principale est les couleurs bleues du ciel, de l'eau de mer et du pond des images.

Le résultat est bien avec 19/20 = 80% images exactes. La courbe de précision et rappel :

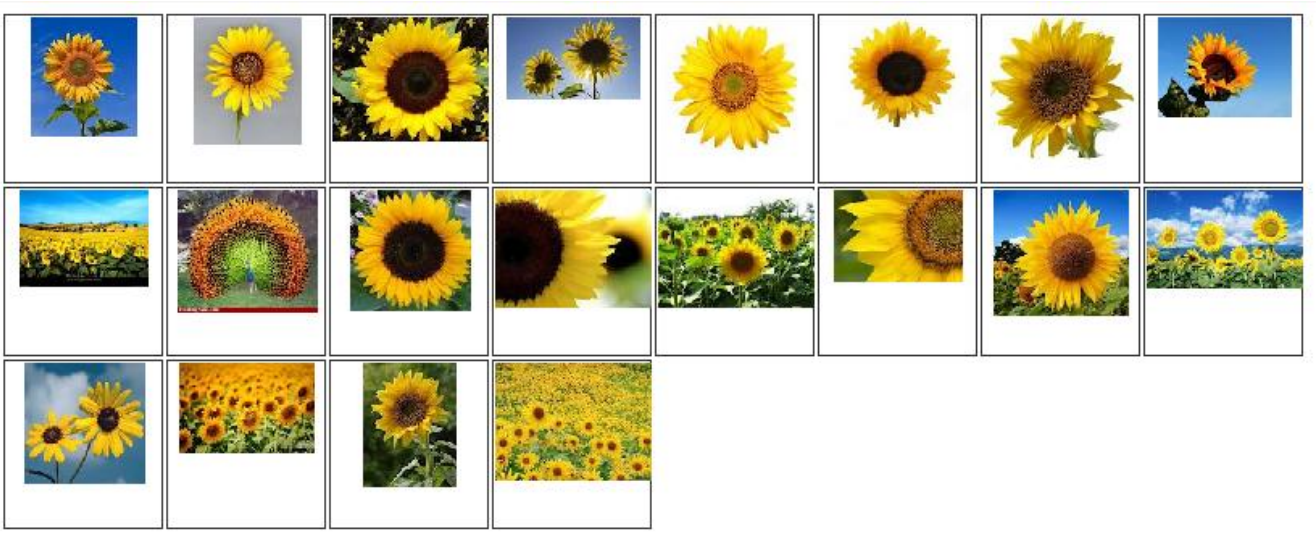


Pour le résultat obtenu ci-dessus, on trouve que les résultats obtenus sont bons pour des objets qui ont peu d'autres formes et peu d'autres couleurs. Dans une base d'images d'un objet, on trouve constamment un couleur principal.

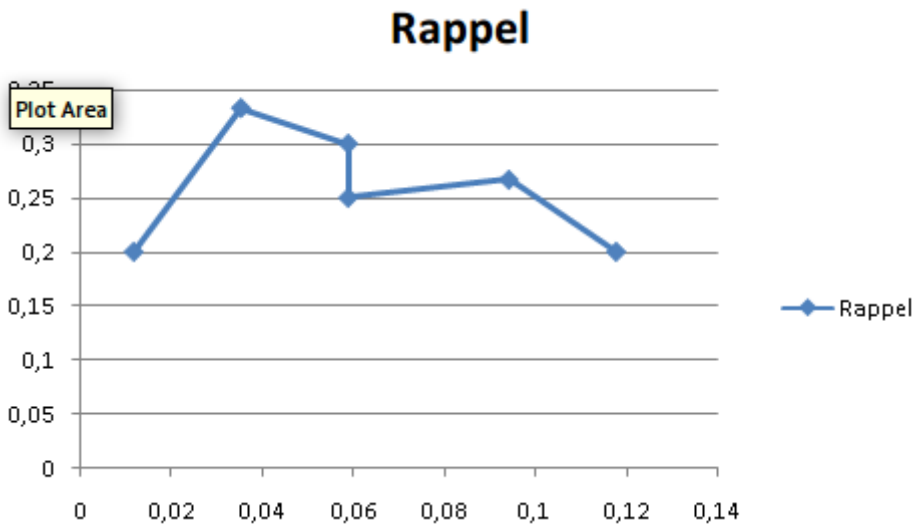
Alors, il existe des cas qui nous donnent des pauvres résultats. Par exemple :

Exemple 2: Requête « **Sunflower »** (85 images dans 1114 images) :

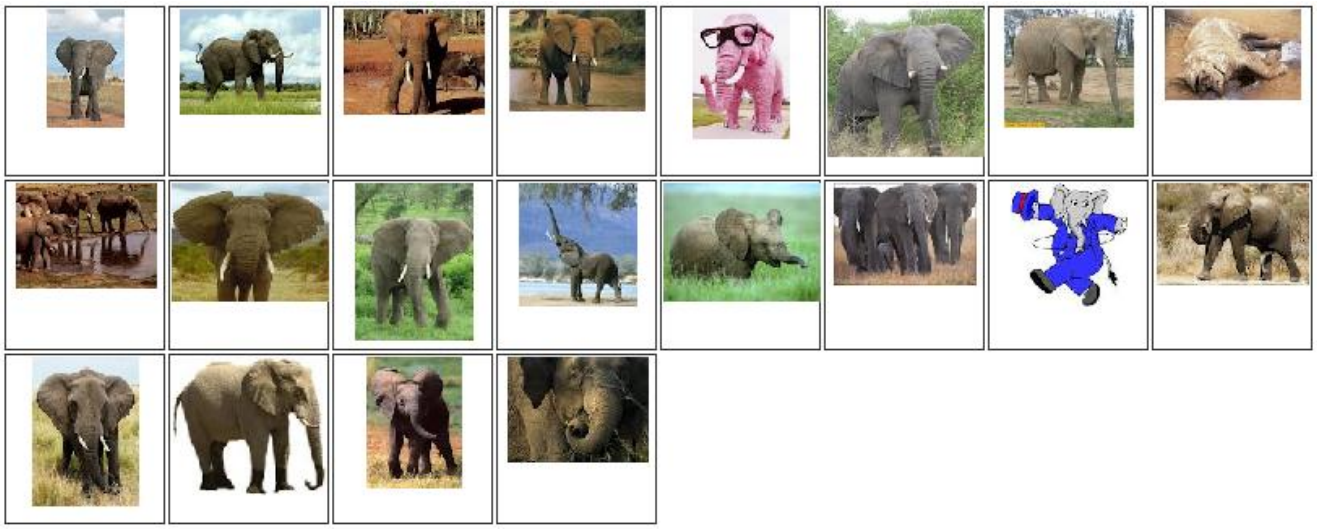
20 images à partir de GoogleImage :



Le résultat n'est pas bon : 5/15 = 33% images exactes.
La courbe de précision et rappel :

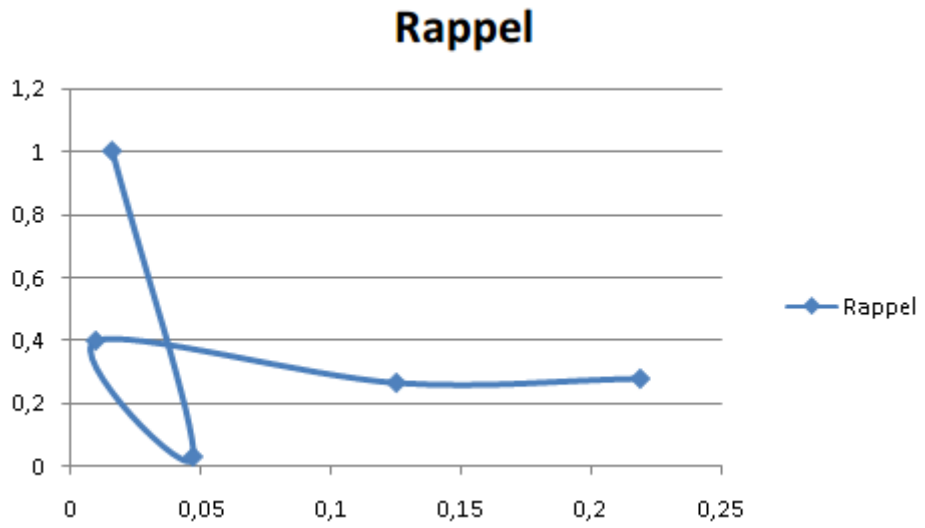


Exemple 3 : « ***elephant*** »
20 images à partir de *GoogleImage* :



Des images locales :
Le résultat est mauvais. 6/20 = 30% images exactes. Avec le résultat de 20 premières images, il est très mal. Parce que le programme utilise des caractéristiques de la couleur (histogramme, moment de couleur, moment de Hu), mais dans le groupe d'images pertinentes de *GoogleImages*, il y a beaucoup de couleurs : le bleu, le jaune, le pink, le noir, etc. On ne trouve pas de couleur principal.

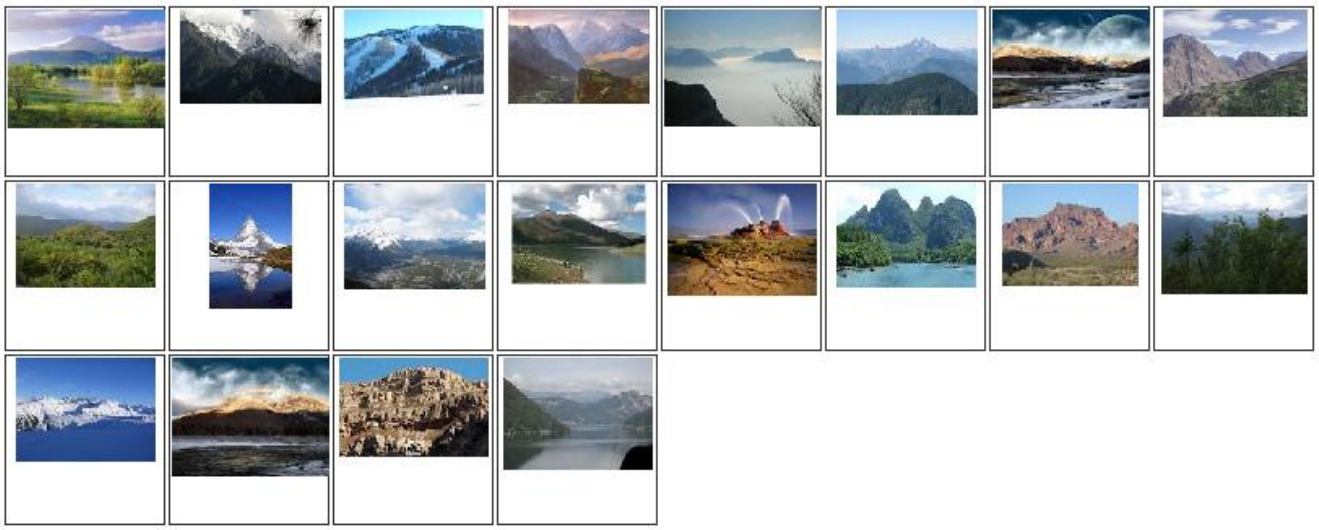
La courbe de précision et rappel :



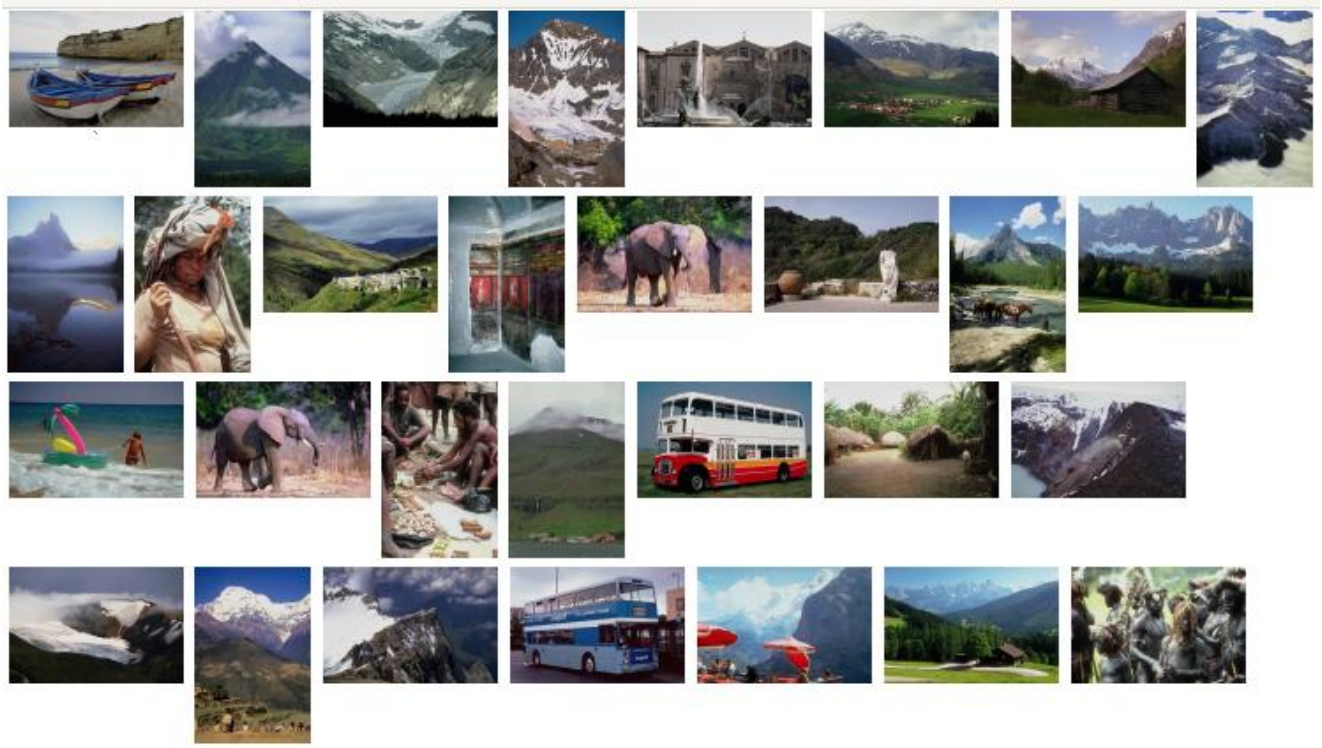
Pour la base d'image Wang.

Exemple 1 : « ***moutains*** »

20 images obtenues à partir de *GoogleImage*.

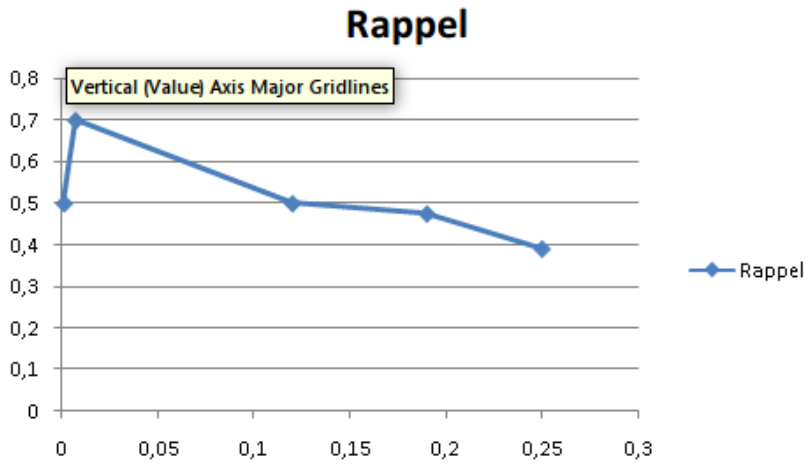


Le résultat dans la base d’image Wang :



Le résultat est assez bon : 12/20 = 60% images exactes.

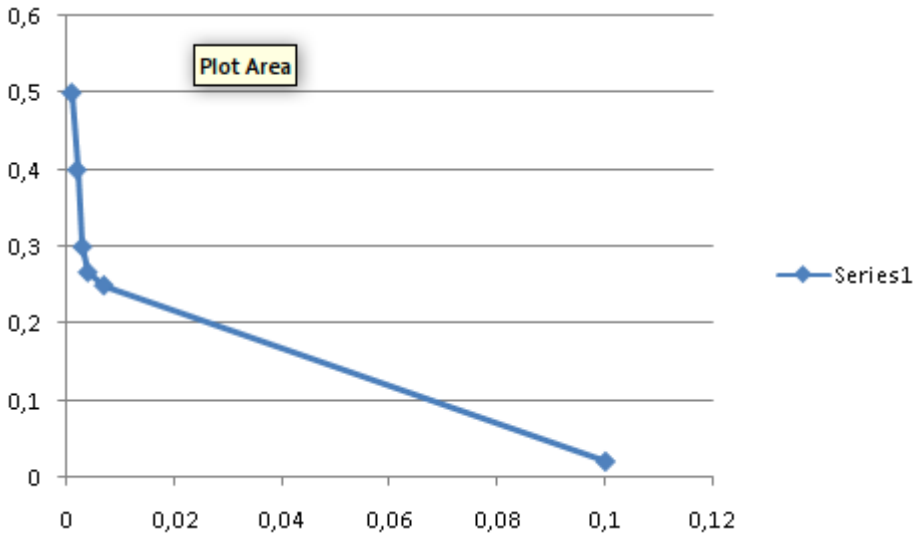
La courbe de précision et rappel :



Exemple 2 : ***mouveuments***
20 images obtenues à partir de *GoogleImage*.

Des images de Wang :

La résultat est aussi mauvais : 5/20 = 40% images exactes.
La courbe de précision et rappel :



Selon les diagrammes ci-dessus, les résultats sont pauvres. Il y a des raisons pour expliquer cette baisse de qualité. Premièrement, c'est le manque de caractéristiques. Dans notre programme qu'on a réalisé, on utilise des caractéristiques de la couleur et on n'utilise qu'une espace de couleur (RGB). On manque des informations. Deuxièmes, c'est le problème de l'histobin. On utilise 48 bins (16 bins pour chaque l'espace de couleur). L'histobin est plus compact que l'histogramme mais ça peut cause des problèmes. Deux images peuvent avoir deux histogrammes totalement différents mais elles ont le même histobin. C'est aussi une raison qui influence le résultat. Troisièmement, c'est le problème sur la

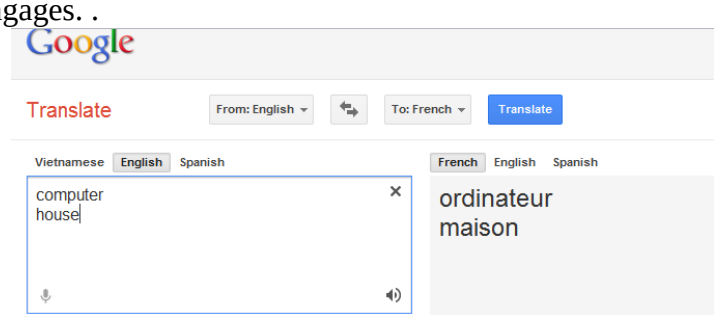
taille d'images. Dans ce travail, je cherche des images par le contenu, alors toutes les choses qui appartiennent au contenu d'images influencent de la recherche des images. La taille d'images est le nombre des pixels d'images et des pixels représentent le contenu d'images. Pour les images récupérées à partir de *GoogleImages* dont les tailles sont plus grandes que les images de Caltech 101 ou Wang. C'est aussi une raison qui influence le résultat. Si l'ensemble d'images de Source et celui de Destination ont la même taille, le résultat va mieux. Quatrièmement, notre programme récupère automatiquement environ 20 images à partir de Google Images. Le nombre des images récupérées, c'est une raison principale qui influence le résultat. Cinquièmement, il existe encore une raison, c'est le système GoogleImages. D'abord, ce système est trop général, très gros. Ensuite, on peut utiliser tous les mots, touts les langages pour GoogleImages, alors on acquiert des différents résultats entre des mots-clés. Sixièmement, la base Caltech contient 101 classes d'images qui ont des nombres d'images différents. Par exemple, la classe « ant » a 42 images, mais la classe "airplanes" a 800 images (comme les résultats ci-dessus, la recherche avec le mot-clé *airplanes* donne le meilleur résultat). C'est une raison qui influence grandement des résultats différents. Enfin, il manque un module d'apprentissage des images, des images obtenues à partir de GoogleImage peuvent être non- pertinentes. Quand on fait la recherche d'images que les entrées sont des images non-pertinentes, des résultats sont très mauvais.

Chapitre IV – Conclusion et Perspective

J'ai construit un système de recherche d'images par le contenu sur des requêtes textuelles. J'ai évalué la performance du système, mais il reste beaucoup de choses à faire. Dans ce chapitre, je vais résumer des résultats obtenus, des aspects limitent de mon programme et à la fois, je vais donner des façons pour pouvoir améliorer ce travail.

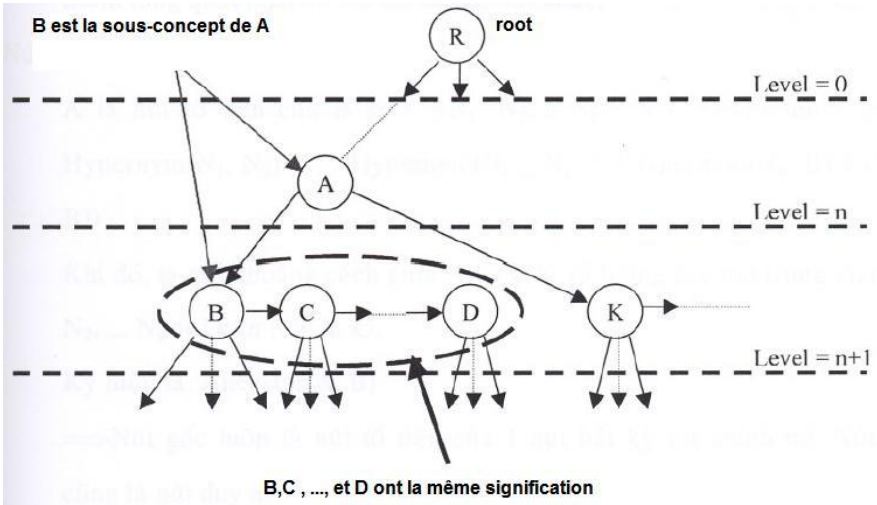
Premièrement, il manque une méthode d'apprentissage des images obtenues à partir de GoogleImage (par exemple : SVM, kNN, etc.). Avec une méthode d'apprentissage, on peut supprimer des images non –pertinentes. Des images à partir de GoogleImage sont plus proches avec le mot-clé. Les résultats de la recherche d'image locale vont mieux. C'est très important.

Deuxièmes, j'ai seulement récupéré des images à partir de GoogleImages avec des requêtes textuelles en anglais. Mon programme peut marcher avec des requêtes textuelles en français, en vietnamien, etc. On peut profiter de GoogleTraduction pour traduire des mots anglais en d'autres langages. .



Troisièmes, il faut ajouter des types de caractéristiques des images, par exemple : forme, MPEG-7. 68 valeurs pour représenter à une image, il ne suffit pas. On peut aussi tester d'autres méthodes de calcul similarité pour améliorer le résultat.

Quatrièmement, on peut implémenter une méthode de texte avant d'envoyer des mots-clés à GoogleImage. Le but est de permettre de faire la recherche d'images avec des mots ayant des significations proches. Pour faire cela, on peut profiter des dictionnaires en ligne, la structure de WordNet.



Référence :

[1] Gang Wang et David Forsyth . *Object image retrieval by exploiting online knowledge ressources.* IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2008.

[2] Keiji Yanai. *Generic Image Classification Using Visual Knowledge on the Web* . Proc. of ACM Multimedia 2003, Berkeley USA, pp. 67-76 (2003/11)).

[3] Schroff, F.; Criminisi, A.; Zisserman, A. "*Harvesting Image Databases from the Web*" , Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference.

[4] Dr. Fuhui Long, Dr. Hongjiang Zhang and Prof. David Dagan Feng. *Fundamentals of content-based image*.

[5] Yiming Liu et al*, Using Large-Scale Web Data to Facilitate Textual Query Based Retrieval of Consumer Photos,* 2009.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li and Li Fei-Fei . *ImageNet: A Large-Scale Hierarchical Image Database* . IEEE Computer Vision and Pattern Recognition (CVPR), 2009

[7] Y. Liu*, D. Xu, Ivor W. Tsang and J. Luo, "*Using Large-Scale Web Data to Facilitate Textual Query Based Retrieval of Consumer Photos*," *ACM Multimedia Conference* (ACM MM), 2009

[8] Dr. Fuhui Long, Dr. Hongjiang Zhang and Prof. David Dagan Feng. *Fundamentals of content-based image*.

[9] TRAN Thi Cam Giang. *Recherche d'images par le contenu basée sur des requêtes textuelles*.Rapport de TPE, promo 15. IFI, 2010.