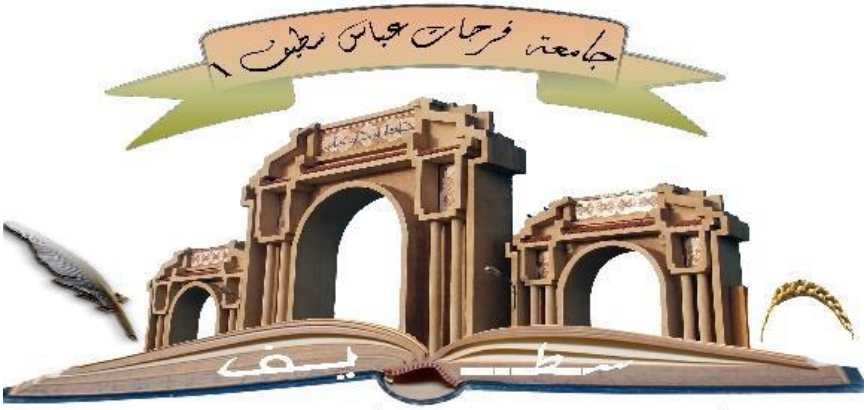
République Algérienne Démocratique et Populaire
Ministre de l'Enseignement Supérieur et de la Recherche Scientifique
Université Ferhat Abbas de Sétif 1
Faculté des Sciences



Thèse de Doctorat

En Sciences

Option : Informatique

Indexation des bases de données images

Présenté par

HADI fairouz

Devant le jury :

Pr. BENHOCINE abdelhamid	Université FERHAT ABBAS SETIF -1	Président
Pr. ALIOUAT zibouda	Université FERHAT ABBAS SETIF -1	Rapporteur
Pr. BENMOHAMED mohamed	Université de CONSTANTINE 2	Examinateur
Dr. BOUDRIES abdelmalek	Université de BEJAIA	Examinateur

Année Universitaire 2020-2021

Résumé

Cette thèse s'inscrit dans la problématique de l'indexation et la recherche d'images par le contenu dans des bases d'images volumineuses. Le travail a pour objectif de minimiser le temps de réponse pour une requête donnée et de combler le vide sémantique. Cette contribution originale consiste en la proposition d'une nouvelle plateforme de recherche d'informations dans des bases de données constituées de fichiers DICOM (Digital Imaging COmmunication in Medicine). Ces fichiers contiennent plusieurs images et informations cliniques contextuelles sur un patient. Notre but est de sélectionner des images de ces fichiers similaires à une image proposée dans une requête, en utilisant le principe de la récupération d'image basée sur le contenu (CBIR). L'objectif principal est d'aider les radiologues à diagnostiquer les images médicales. Le traitement d'images médicales nécessite la manipulation d'une énorme quantité de données, que nous appellerons «Big Data». Les Big Data non structurés peuvent créer des problèmes liés à la latence. Les architectures distribuées basées sur le parallélisme peuvent atténuer cette latence. De plus, la disponibilité de l'image en cas de défaillance d'un serveur dans un système aussi énorme augmente le temps de latence. Pour résoudre les défis de la latence dans le traitement d'images dans un énorme système, nous proposons une plateforme en tant que service pour le Cloud computing, basée sur un arbre de décision. Les serveurs de cette plate-forme sont distribués et fonctionnent de manière parallèle. De plus, un système tolérant aux pannes est proposé pour garantir la disponibilité des images et minimiser la latence de récupération des images en cas de défaillance du serveur.

Mots clés

CBIR, Big Data, Arbre de décision, DICOM, Tolérance aux pannes, Disponibilité, Cloud computing, PaaS.

Abstract

This thesis deals with the problem of indexing and searching for images by content in large image databases. The aim of the work presented in this thesis is to minimize the response time for a given request and to fill the semantic void. An original contribution concerns the proposal for a new platform for information retrieval in databases consisting of DICOM (Digital Imaging COmmunication in Medicine) files. These files contain several images and contextual clinical information about the patient. Our goal is to select images from these files similar to an image proposed in a query, using the principle of Content Based Image Retrieval (CBIR). The main aim is to help radiologists with the diagnosis of medical images. Medical image processing requires to the manipulation of a huge amount of data, which we will call "Big Data." Unstructured Big Data can create some issues related to latency. Distributed architectures based on parallelism can alleviate a latency problem. Also, achieving image availability in case of a server failure in such a huge system increases the latency time. To solve the challenges of latency in image processing in an enormous system, we propose a Platform as a Service for Cloud Computing, based on a Decision Tree. The servers in this platform are distributed and work in a parallel way. Also, a fault tolerant system is proposed to ensure image availability and minimize image recovery latency in the case of a server failure.

Keywords
CBIR, Big Data, Decision Tree, DICOM, Fault tolerance, Availability, Cloud computing, PaaS.

ملخص

تتناول هذه الرسالة موضوع فهرسة الصور والبحث عنها حسب المحتوى في قواعد بيانات الصور الكبيرة. الهدف من العمل هو تقليل وقت الاستجابة لطلب معين وملء الفراغ الدلالي. تتكون هذه المساهمة الأصلية من مقترح لمنصة جديدة للبحث عن المعلومات في قواعد البيانات المكونة من ملفات DICOM (التصوير الرقمي في الطب). تحتوي هذه الملفات على صور متعددة ومعلومات إكلينيكية سياقية عن المريض. هدفنا هو تحديد الصور من هذه الملفات التي تشبه الصورة المقترحة في الطلب، باستخدام مبدأ استرجاع الصور على أساس المحتوى (CBIR). الهدف الرئيسي هو مساعدة أطباء الأشعة في تشخيص الصور الطبية. تتطلب معالجة الصور الطبية معالجة كمية هائلة من البيانات، والتي سنطلق عليها "البيانات الضخمة". يمكن أن تؤدي البيانات الضخمة غير المنظمة إلى حدوث مشكلات تتعلق بوقت الاستجابة. يمكن للبنى الموزعة القائمة على التوازي أن تخفف من وقت الاستجابة هذا. بالإضافة إلى ذلك، فإن توفر الصورة في حالة فشل الخادم في مثل هذا النظام الضخم يزيد من وقت الاستجابة. لحل تحديات وقت الاستجابة لمعالجة الصور في نظام ضخم ، نقدم منصة قائمة على شجرة القرار كخدمة للحوسبة السحابية. يتم توزيع الخوادم على هذه المنصة وتعمل بالتوازي. بالإضافة إلى ذلك ، يتم توفير نظام متسامح مع الأخطاء لضمان توفر الصورة وتقليل وقت الاستجابة لاستعادة الصورة في حالة فشل الخادم.

الكلمات الدالة

استرداد الصور على أساس المحتوى (CBIR) ، البيانات الضخمة ، شجرة القرار ، التصوير الرقمي في الطب (DICOM) ، تحمل الأخطاء ، التوفر ، الحوسبة السحابية ، النظام الأساسي كخدمة (PaaS) .

DÉDICACES

A TOUTE MA FAMILLE.

A MES COLLÈGUES.

A MES AMIES.

Remerciements

En premier lieu, je tiens à exprimer ma reconnaissance et ma gratitude à ma directrice de thèse le Pr. Zibouda ALIOUAT, pour m'avoir guidé avec beaucoup de justesse dans mon travail de recherche. Merci pour votre confiance, votre aide et votre patience.

Je tiens également à remercier les membres du jury pour leur expertise, leurs conseils et leur temps précieux tout au long de la révision de ce document.

Je rends hommage au Pr Boukerram Abdellah, ALLAH yarahmou, qui était initialement mon directeur de thèse. Je me rappellerai de ses qualités humaines et professionnelles, et je prie ALLAH tout puissant de l'accueillir dans son vaste paradis.

Je remercie tous ceux qui, de près ou de loin, ont contribué à l'aboutissement de ce travail.

Merci.

Table des matières

Liste des figures

Liste des tableaux

Liste des abréviations

ACR	American College of Radiology
AID	Automatic Interaction Detection
AMD	Advanced Micro Devices
ANSI	American National Standard Institute
API	Application Programming Interface
ATI	Array Technologies Incorporated
BF	Brute Force
CART	Classification And Regression Trees
CBIR	Content Based Image Retrieval
CC	Cloud Computing
CFDT	Clustering Feature Decision Tree
CHAID	CHi-squared Automatic Interaction Detection
CH-M	Cluster Head Men
CH-MC	Cluster Head Men Colorectal
CH-ML	Cluster Head Men Lung
CH-MLI	Cluster Head Men Liver
CH-MO	Cluster Head Men Other
CH-MP	Cluster Head Men Prostate
CH-MS	Cluster Head Men Stomach
CH-W	Cluster Head Women
CH-WB	Cluster Head Women Breast
CH-WC	Cluster Head Women Colorectal
CH-WL	Cluster Head Women Lung
CH-WO	Cluster Head Women Other
CH-WU	Cluster Head Women Cervical
CUDA	Compute Unified Device Architecture
DICOM	Digital Imaging COmmunication in Medicine
DWT	Discrete Wavelet Transform
EMD	Earth Mover's Distance
FLANN	Fast Library for Approximate Nearest Neighbors
GIF	Graphics Interchange Format
GLCM	Grey Level Co-occurrence Matrix
GLOH	Gradient Location and Orientation Histogram

GPU	Graphical Processing Unit
HDFS	Hadoop Dstributed File System
HSV	Hue –Saturation –Value
IaaS	Infrastructure as a Service
ID3	Iterative Dichotomiser *3*
KNN	K-Nearest Neighbors
k-ppv	k-plus-proches-voisins
LBP	Local Binary Pattern
LSH	Locality Sensitive Hashing
NEMA	National Electrical Manufactures Association
NIST	National Institute of Standards and Technology
OMS	Organisation mondiale de la santé
OpenCL	Open Computing Language
PaaS	Platform as a Service
QBE	Query By Example
QUEST	Quick, Unbiased, Efficient, Statistical Tree
RIC	Recherche d'Image par le Contenu
RVB	Rouge, Vert et Bleu
SaaS	Software as a Service
SGBD	Système de Gestion de Bases de Données
SIFT	Scale Invariant Feature Transform
SLIQ	Supervised Learning In Quest
SQL	Structured Query Language
SURF	Speeded up Robust Feature
SVM	séparateurs à vaste marge
TDMA	Time Division Multiple Access
TH	Transformation de Hough
THAID	THeta Automatic Interaction Detection
TREC	Text REtrieval Conference
TTP	Time Triggered Protocol
URI	Uniform Resource Identifier
XML	Extensible Markup Language

Introduction générale

L'accroissement constant, en nombre et en volume, des bases de données numériques requiert des méthodes d'indexation et des outils de recherche efficaces.

Des bases de données ont alors été construites afin de stocker toutes ces images. Au fil des années, les images de natures très diverses ont gagné en importance occupant parfois jusqu'à plusieurs téraoctets de mémoire. Afin de pouvoir consulter et retrouver au mieux les images, de nombreux chercheurs s'intéressent de près à cette discipline qui est, la recherche d'images. C'est un problème ouvert qui met à l'épreuve des thématiques de recherche diverses :

- Les *bases de données* pour le stockage et l'indexation des images,
- Le *traitement d'images* pour l'extraction de paramètres discriminants.
- Les *statistiques*, les *distances métriques* pour l'évaluation de distances entre images.
- La *classification* pour le classement des images les unes par rapport aux autres.
- L'aspect *réseau* pour la gestion des bases de données à distance en tenant compte des problèmes de sécurité et de transmission de l'information qui en découlent.

Présentation du problème

Avec l'expansion de l'informatique et du multimédia, une problématique nouvelle est apparue : gérer les quantités énormes et croissantes de documents numériques aujourd'hui disponibles. Parmi ces documents, nous nous intéressons plus particulièrement aux images numériques. L'image sous toutes ses formes (photographie, vidéo, graphique ou dessin numérique) est donc aujourd'hui à disposition des professionnels mais aussi du grand public.

Nous distinguons trois grandes origines à ce phénomène :

- ✓ la démocratisation de l'informatique, l'évolution rapide des performances et de la capacité de stockage des machines facilitent le traitement et l'accès à l'information numérique;
- ✓ l'accès aux scanners et autres capteurs de moins en moins chers et de plus en plus performants, tels que les appareils photographiques ou les caméscopes numériques favorise la croissance des banques d'images;
- ✓ les progrès réalisés dans le domaine des réseaux de télécommunication permettent la transmission et le partage de l'information numérique aussi bien localement que mondialement.

Les images fixes et les séquences d'images sont en général compressées puis archivées dans des bases de données, généralistes ou spécialisées, accessibles par les réseaux de télécommunication.

Afin de pourvoir à la réutilisation de ces données, il est donc capital de développer des outils de recherche. Ils sont généralement directement liés aux outils d'archivage, voire de compression des documents numériques.

Contrairement aux documents textuels, pour lesquels des méthodes d'indexation et de recherche existent depuis les années 1970, dans les travaux de Dierk et Salton [DIE 71], les outils d'analyse et d'interprétation de l'image sont souvent en décalage avec le contenu sémantique, de celle-ci souvent riche. Nous admettons communément deux niveaux d'indexation :

- ✓ le premier niveau dit « numérique » fait référence aux caractéristiques primaires ou bas niveaux tel que la couleur, la forme et la texture, etc,
- ✓ le deuxième niveau dit « sémantique » a trait à l'interprétation de l'image.

L'émergence de la recherche d'images par le contenu remonte au début des années 90. Depuis cette date, sont apparus plusieurs systèmes comme QBIC chez IBM [FLI 95] ou bien SurfImage [NAS 98] à l'INRIA. Si la plupart des travaux actuels et passés portent sur le niveau primaire d'indexation, force est de constater que les attentes de l'utilisateur se situent plutôt au niveau sémantique. Dans ce cadre, des systèmes basés sur une description textuelle (mots-clés associés aux images) ont été proposés ; cependant, le mot-clé est une source d'information externe manuellement attachée aux documents. Le procédé d'indexation par mots clés est donc long et coûteux puisque manuel, et délicat de surcroît car la pertinence des termes attribués aux images conditionne la qualité des recherches futures. En outre, il apparaît qu'une liste réduite de mots-clés ne peut couvrir complètement la sémantique souvent riche portée par l'image. L'approche « basée contenu », visant à extraire directement de l'image l'information nécessaire à sa caractérisation, apparaît comme une alternative à l'approche textuelle. Les deux sources d'information, image et texte, peuvent être considérées comme complémentaires et des systèmes hybrides « image-texte » ont été proposés [SCL 99, WES 00].

La recherche d'images par le contenu dans les bases multimédia (Content based image retrieval CBIR ou RIC : Recherche d'Image par le Contenu, en Français) est désormais un domaine de recherche très actif. La recherche par contenu visuel dans les grandes collections d'images, reste une opération clé pour de nombreuses applications, telles que la

reconnaissance d'objets, l'imagerie médicale, la sécurité,…. Donnant naissance à des problématiques nouvelles qui émergent de l'intersection de deux domaines de recherche en informatique: l'analyse d'images et les bases de données. Ces problèmes sont essentiellement liés à l'acquisition, l'indexation, l'interrogation, la comparaison, la classification, et la visualisation.

Les systèmes de recherche d'images par le contenu cherchent à représenter le contenu des images de manière automatique, à l'aide de descripteurs visuels représentant les données multimédia. Les techniques d'indexation d'images permettent l'organisation et la structuration de l'ensemble des descripteurs afin d'effectuer des recherches par similarité visuelle, rapides et efficaces. L'idée clé dans la recherche d'images est la notion de ressemblance, d'où l'utilisation de notion de proximité, de similarité, et de distance entre objets. Généralement, la requête est exprimée sous la forme d'un ou de plusieurs vecteurs qui peuvent être de grande dimension.

Pour cela, les systèmes utilisent des méthodes d'analyse d'images pour décrire automatiquement le contenu visuel des images. Il s'agit d'extraire à partir des images des vecteurs numériques de grande dimension appelés descripteurs et de leurs associer une mesure de similarité. Le stockage de ces descripteurs intéresse aussi bien les professionnels que le grand public. Il en résulte une production permanente et considérable d'images dans des domaines pluridisciplinaire tels que l'imagerie satellitaire, la santé, l'architecture, la communication/publicité/design, l'art et l'éducation, l'industrie pour la surveillance industrielle, ...

Contributions

1) Notre première contribution repose sur l'étude des approches basées sur l'indexation et la recherche textuelle d'images et l'indexation et recherche visuelle d'images. L'objectif principal de cette contribution est d'améliorer le résultat de recherche en combinant la recherche textuelle et la recherche par contenu. Cette contribution permet de combler le problème de vide/fossé sémantique.

   Cette contribution a fait l'objet de la production scientifique suivante :

   International Conference: Hadi Akram, Hadi Fairouz, Boukerram Abdallah. *Indexation et recherche visuo-textuelle des bases de données images*. International Conference on Artificial Intelligence and Information Technology (ICAIIT'2014), Ouargla, Algeria.

2) La seconde et la principale contribution est la proposition d'une nouvelle plateforme efficace en tant que service (PaaS) sur le Cloud public pour le système CBIR. L'objectif principal de cette deuxième contribution est de résoudre les défis de la latence dans la recherche d'images dans un énorme système.

   Cette contribution a fait l'objet d'un article de revue dans le journal " Ingénierie des Systèmes d'Information", International Information and Engineering Technology Association (IIETA) :

   Article : HADI, Fairouz, ALIOUAT, Zibouda, et HAMMOUDI, Sarra. *Efficient Platform as a Service (PaaS) Model on Public Cloud for CBIR System*. Ingénierie des Systèmes d'Information, 2020, vol. 25, no 2, p. 215-225.

Plan de la Thèse

Cette thèse s'organise comme suit :

Le chapitre 1 est consacré à l'indexation et à la recherche d'informations dans une base de données. Nous donnons un aperçu sur les différents systèmes de recherche d'informations dans une base de données. Nous expliquerons en détail le principe des systèmes CBIR et nous exposerons l'état de l'art des travaux de la littérature ayant utilisé les systèmes CBIR.

Le chapitre 2 présente les Clouds computing. Nous étalons les définitions du Cloud, son architecture, ses différents modèles de services et de déploiement ainsi que ses caractéristiques. Nous détaillerons la relation entre le Cloud computing et le Big Data.

Le chapitre 3 introduit les notions de base des arbres de décision. Il présente un bref historique de l'apparition des arbres de décision dans la littérature, Un exemple introductif des arbres de décision, le principe général de la construction d'un arbre de décision, le concept d'apprentissage d'un arbre de décision, l'algorithme ID3, la méthode CART et les forêts aléatoires. Ensuite, une exposition des avantages et des inconvénients des arbres de décision.

Le chapitre 4 est consacré à nos propres contributions. Nous commencerons par la présentation de notre première contribution : *Indexation et recherche visuo-textuelle des bases de données images*. C'est une combinaison entre l'indexation et la recherche textuelle d'images et l'indexation et la recherche visuelle d'images. Puis, nous exposerons notre deuxième contribution : *Efficient Platform as a Service (PaaS) Model on public Cloud for CBIR System*. Dans cette deuxième contribution, nous proposons une nouvelle plateforme de

recherche d'images dans des bases de données constituées de fichiers DICOM (Digital Imaging COmmunication in Medicine).

Nous terminerons par une conclusion générale et des perspectives d'avenir pour enrichir ce travail.

Chapitre 1

Indexation et recherche d'informations dans une base de données

1. Introduction

De nos jours, la recherche d'information est primordiale dans tous les secteurs d'activités : dans le monde industriel, juridique, médical, scientifique, économique et bien sûr informatique. Il faut donc trouver l'information, où elle se trouve, rapidement pour ne pas perdre de temps. Grâce à Internet, on trouve de plus en plus d'information de tous types, il convient de faire le tri et de posséder des outils de recherche performants et d'utilisation facile.

Le but des systèmes de tri automatique (indexation) est de permettre à un utilisateur de trouver, dans des bases d'images, toutes celles qui sont semblables à l'image qui l'intéresse. Un programme d'indexation se conçoit comme un système qui prend en entrée une image de référence et qui retourne un critère de similarité entre l'image de référence et toutes les images de la base. Ceci permet de trier ces images, de la plus similaire à la moins similaire.

On distingue, deux approches principales: l'une emploie des annotations textuelles manuelles et l'autre emploie des descripteurs extraits automatiquement à partir des images. La première approche, basée sur l'annotation textuelle manuelle d'images, est aujourd'hui la plus employée. Mais l'indexation de ces images représente une tâche longue et répétitive pour l'humain, surtout avec les bases d'images qui deviennent de plus en plus grandes, où l'on remarque beaucoup d'erreurs d'indexation, liées au fait que le texte ne correspond pas toujours à l'image.

Pour surmonter les défaites des systèmes de recherche d'images basées sur l'annotation textuelle manuelle, le système de recherche des images basé essentiellement sur le contenu est proposé. Cette deuxième approche a deux directions :

- recherche d'image basée sur le contenu symbolique.
- recherche d'image basée sur le contenu sémantique.

Les méthodes symboliques emploient des descripteurs extraits automatiquement de l'image telle que la couleur, la texture et les formes..... Quoiqu' il est difficile de trouver les

descripteurs puissants pour représenter des images : à cet effet, des méthodes de recherche sémantique sont apparues pour améliorer le résultat de recherche.

Dans ce chapitre, nous donnons une idée sur les bases de données ainsi que leurs architectures. Nous rappelons tout d'abord le principe général de l'indexation et de la recherche d'information. Nous expliquons en détail le principe des systèmes CBIR et nous exposons l'état de l'art des travaux de la littérature ayant utilisé les systèmes CBIR. Nous étalons les différentes méthodes d'extraction des caractéristiques d'images. Nous présentons les diverses mesures de similitude entre descripteurs, ainsi que les mesures pour l'évaluation d'un système CBIR.

2. Les bases de données

L'appellation «gestion de bases de données» [GAR 83] désigne la branche de l'informatique qui étudie le stockage et l'interrogation des données numériques. Une base de données informatique est donc un ensemble d'informations numériques stockées selon un modèle dans le but de les conserver, de les enrichir et de les interroger avec la garantie de l'intégrité de ces données. Ces informations peuvent être de n'importe quel type : texte, image, son ou vidéo car informatiquement, ces données sont représentées par un ensemble de nombres binaires en mémoire. Le modèle de la base de données permet d'organiser les informations et de leur ajouter un sens, une sémantique qui représente les relations entre ces objets et le monde réel.

Le système de gestion de bases de données idéal doit fournir un certain nombre de services [LAN 08] :

– La centralisation de la gestion des données (mais pas des données elles-mêmes qui peuvent être réparties) doit permettre un regroupement logique des informations.

– L'indépendance des applications par rapport à la structure des données doit faciliter l'évolutivité des applications.

– L'environnement de programmation doit être non procédural, l'utilisateur spécifie ce qu'il veut et non la procédure à suivre pour l'obtenir (QUOI mais pas COMMENT).

– L'environnement d'utilisation doit être convivial et doit offrir un meilleur accès à l'information.

– Le niveau de sécurité et l'intégrité des données doivent être garantis (contrôle d'accès, cryptage, transaction, respect des contraintes sur les données. . .).

– Les données et les applications doivent être portables sur différents systèmes (indépendance

vis-à-vis de l'architecture matérielle et logicielle).

En 1975, l'organisme de normalisation américain ANSI (*American National Standard Institute*) a proposé un modèle normalisé de base de données assurant les services ci-dessus [LAN 08].

2.1. Architecture des bases de données

Un système de gestion de bases de données (SGBD) est composé de quatre parties : le niveau physique, le niveau interne, le niveau conceptuel et le(s) niveau(s) externe(s) [LAN 08] :

Le niveau physique : Il est chargé de la gestion physique de la base de données au niveau du matériel et du système d'exploitation. Il assure la gestion des disques, pistes et secteurs ainsi que la gestion des tampons de lecture/écriture en mémoire. C'est le plus bas niveau de l'architecture, la réalité binaire de la base.

Le niveau interne : Le schéma interne fournit une perception plus technique de la base de données. On décrit à ce niveau un ensemble d'objets informatiques (fichiers, index, listes. . .) dont l'organisation et les caractéristiques visent à optimiser les ressources (disques, mémoires, microprocesseur) lors de l'exploitation de la base de données.

Le niveau conceptuel : Le niveau conceptuel décrit les concepts utilisés dans la base. Il est totalement indépendant de la technologie utilisée (matérielle et logicielle) pour gérer la base. Il se compose de deux parties : la structure de données qui comprend l'ensemble des données et des liens pour les applications et les contraintes d'intégrité qui garantissent la cohérence et la vraisemblance des données.

Le niveau externe : Ce niveau représente le niveau application dans lequel les utilisateurs ont accès uniquement à une vue partielle de la base de données (celle qui les intéresse). Il y a un nombre quelconque de schémas externes pour une base de données qui dépend du nombre d'utilisateurs et de leurs droits d'accès en fonction des applications.

Ce découpage en couches permet d'assurer une indépendance à chacune des parties de la base de données. Ainsi, on pourra aisément mettre le même niveau conceptuel ou le même niveau externe sur plusieurs systèmes d'exploitation différents. On sépare ainsi le contenu du contenant (analogie avec XML [1]).

2.2. Le modèle relationnel

Le modèle relationnel [COD 02] est le modèle le plus utilisé dans les bases de données actuelles. Son succès vient du fait qu'il est basé sur la notion mathématique d'algèbre et qu'il possède donc un comportement régi par des règles rigoureuses. Il contient des règles de création, de manipulation et d'interrogation des données assurant un état fiable des données à n'importe quel instant de la vie de la base de données.

Le langage de manipulation et d'interrogation des données est *SQL* (*Structured Query Language*). Il offre une syntaxe très intuitive pour les requêtes et permet de réaliser toutes les opérations de l'algèbre relationnelle (sélection, projection, jointure, . . .).

Les bases de données relationnelles contiennent principalement des données alphanumériques (lettres et chiffres). Il existe pour ces types de données des techniques d'indexation et de recherche très rapides basées sur la structuration des données sous forme d'arbres (B-arbre). Ces techniques bénéficient des nombreuses années de recherche déjà effectuées dans le domaine de la recherche de texte, les principaux résultats sont accessibles dans les actes de la conférence TREC [2] (Text REtrieval Conference) qui est la référence dans cette discipline. C'est grâce à toutes ces techniques que la recherche sur Internet par moteurs de recherche est possible.

Pour les bases de données textuelles, il existe des méthodes de construction permettant de concevoir la base de données dans son intégralité en partant du problème du monde réel pour arriver à la modélisation informatique de ce problème. La méthode MERISE [NAN 01], pour ne citer qu'elle, offre toutes les étapes de la conception d'un système d'information. Malheureusement, il n'existe pas, à l'heure actuelle, une méthode similaire pour les bases d'images en raison de la complexité intrinsèque de ces dernières.

La structure des données textuelles permet de les classer rapidement selon un ordre alphanumérique simple (ordre alphabétique par exemple). Ceci facilite l'indexation et la recherche. Les moteurs de recherche Internet sont basés sur l'indexation des mots contenus dans les pages web et des techniques d'association par synonymes et par relation de proximité entre les mots. Le fonctionnement des moteurs de recherche permet de comprendre les techniques d'indexation mises en œuvre pour l'indexation de textes. Il n'existe pas d'ordre aussi simple sur les images, le problème est de définir les critères de classement des images.

2.3. Les bases de données d'images

Les bases de données gèrent de façon efficace les données de type texte mais sont mal adaptées aux données multimédia. Toutefois, et afin de pouvoir inclure des images dans les bases de données, un type spécial a été ajouté dans les bases de données relationnelles. Il s'agit du type objet binaire (*BLOB*) dans lequel on peut mettre une image, un code exécutable ou n'importe quel objet informatique, quel que soit sa taille. Ce type ne permet pas de résoudre les problèmes de la recherche d'images, mais il permet de stocker les données multimédia : images, vidéos et sons dans la base. La plupart des bases de données ont donc la capacité de stocker des données multimédia, mais sans moyen pour les interroger dans le modèle lui-même.

Une base d'images est donc une base de données contenant des images et/ou leur représentant construite selon un certain modèle dans le but de la stocker, de l'interroger, de l'enrichir et de la partager.

On classe les bases d'images en deux grandes catégories pour la recherche et l'indexation [LAN 08, TOL 06] :

– Les **bases généralistes** sont des bases d'images de sujets très variés comprenant des familles d'images très différentes (par exemple : couchers de soleil, montagne, plage, personnages, véhicules, bâtiments, du web, . . .).

– Les **bases spécialisées** sont des bases dans lesquelles on va trouver des images d'un domaine particulier (images médicales, images satellites, images architecturales, photos de visages ou tableaux d'un musée par exemple).

Bien qu'il y ait une distinction entre bases généralistes et bases spécialisées, les bases spécialisées ne sont pas plus faciles à interpréter que les bases généralistes. Les stratégies de recherche d'images pour ces deux catégories de base sont très différentes. Pour la première, on connaît *à priori* le type d'images que l'on peut y rencontrer, ainsi que le type de recherche que l'on va y mener. Cette connaissance *à priori* permet de développer des techniques d'indexation et de recherche très efficaces. Pour la seconde catégorie, par contre, on ne sait pas ce que contiennent les images, et on ne sait pas sémantiquement, ce que recherche l'utilisateur.

Dans notre première contribution : *Indexation et recherche visuo-textuelle des bases de données images*, (voir chapitre 4), nous avons manipulé les bases de données généralistes. par contre, dans notre deuxième contribution :*Efficient Platform as a Service (PaaS) Model on*

public Cloud for CBIR System. (Voir chapitre 4), des bases spécialisées (images médicales : le standard DICOM) sont utilisés dans notre plateforme.

➢ Les bases de données médicales

L'imagerie médicale a évolué vers le numérique. Dans ce contexte, les échanges de données entre matériels de fournisseurs différents se sont faits de plus en plus nombreux. Le besoin d'un standard a émergé dès le début des années 1980 pour aboutir une dizaine d'années plus tard à un standard universel, DICOM (*Digital Imaging COmmunication in Medicine)*. DICOM peut sembler complexe mais ceci provient des spécificités de l'imagerie médicale et des multiples modalités [CHA 04].

• Le standard DICOM

Le standard DICOM est une norme mondiale de l'imagerie médicale. Elle est définie pour stocker des informations sur un patient avec des images réelles et aussi pour faire une normalisation des échanges (réseau, média). Les détails de l'image individuelle décrits dans les métadonnées DICOM se rapportent généralement aux aspects techniques de l'image (par exemple, les lignes, colonnes, modalité, fabricant) plutôt qu'une inclusion d'un organe particulier ou un diagnostic. Ce standard a été créé par ACR (American College of Radiology) en association avec la NEMA (National Electrical Manufactures Association). DICOM est mis en œuvre dans presque tous les appareils de radiologie, d'imagerie cardiologique et de radiothérapie (rayons X, tomodensitométrie, IRM, échographie, etc.), et de plus en plus dans les appareils d'autres domaines médicaux tels que l'ophtalmologie et la dentisterie. Avec des centaines de milliers de dispositifs d'imagerie médicale utilisés, DICOM est l'une des normes d'imagerie médicale les plus largement déployées au monde. Il existe littéralement des milliards d'images DICOM actuellement utilisées pour les soins cliniques. Depuis sa première publication en 1993, DICOM a révolutionné la pratique de la radiologie, permettant le remplacement du film radiographique par un flux de travail entièrement numérique. Bien qu'Internet soit devenu la plate-forme des nouvelles applications d'information des consommateurs, DICOM a permis des applications d'imagerie médicale avancées qui ont «changé le visage de la médecine clinique». Du service des urgences aux tests de stress cardiaque, en passant par la détection du cancer du sein, DICOM est la norme qui fait fonctionner l'imagerie médicale - pour les médecins et pour les patients [3, ABI 19].

2.4. La recherche d'images

Les premiers systèmes de recherche d'images utilisaient des mots-clés associés aux images pour les caractériser. Grâce à cette association de mots-clés, il suffit d'utiliser les méthodes basées sur le texte pour retrouver les images contenant les mots-clés. Plusieurs moteurs de recherche, par exemple Google [4] et Lycos [5], proposent ces recherches d'images basées sur le texte. Ils s'appuient sur le principe simple que dans une page web, il y a une forte corrélation entre le texte et les images présentes. Le principal problème de ces recherches par mots-clés est que le résultat peut être complètement hors sujet.

La recherche par mot clé dans les bases d'images peut donner de bons résultats mais révèle aussi quelques inconvénients :

– L'association de textes à l'image est une démarche réaliste pour de petites bases de données (taille inférieure à 10 000 images), mais est complètement impensable pour des bases de données de taille supérieure. En effet, le temps passé à l'association de mots-clés et la pertinence des mots-clés restent très subjectifs et très dépendants des personnes qui effectuent l'association.

Une solution consiste à ne pas utiliser les mots-clés et donc à considérer l'image et uniquement l'image pour effectuer les recherches. Cette méthode est la **recherche d'images par le contenu**. En règle générale, les systèmes de recherche d'images par le contenu fonctionnent par comparaison d'un vecteur descripteur d'une image requête avec les vecteurs descripteurs des images de la base selon une métrique donnée.

2.5. La recherche par le contenu

La recherche par le contenu consiste à rechercher les images en n'utilisant que l'image elle-même sans aucune autre information. On ne considère que l'image numérique, c'est-à-dire un tableau de pixels (pour *picture elements*, éléments d'images) à deux dimensions (largeur et hauteur). Une image couleur RVB possède trois composantes tandis qu'une image en niveaux de gris n'en possède qu'une seule.

Un problème posé par cette approche est la quantité très élevée de points contenus dans une image. Par exemple une image couleur de 1600 × 1200 pixels contient 1600 ×1200 ×3 = 5760 000 points. Il est donc nécessaire, d'une part pour réduire le temps de calcul et d'autre part pour comparer des images de taille différentes, de travailler avec un ensemble réduit d'attributs d'images.

Ces **attributs** sont des informations spatiales, colorimétriques, géométriques ou statistiques, extraites de l'image, qui synthétisent au mieux l'information contenue dans celle-ci.

Ces attributs sont regroupés dans un vecteur V_i appelé **vecteur descripteur** de l'image *i*. V_i possède *n* composantes réelles (en général) qui sont les attributs extraits de l'image, $V_i \in R^n$.

Pour décrire au mieux une image, il faut tenir compte des transformations géométriques qu'elle peut subir. Il faut donc trouver des descripteurs d'images invariants par rotation, translation et changement d'échelle pour assurer une indépendance de V_i vis-à-vis de ces transformations.

Au lieu de stocker les images elles-mêmes dans la base (ce qui peut poser des problèmes de taille de données dans les grandes bases d'images), on peut choisir de ne stocker dans la base que le descripteur de l'image et son URI (*Uniform Resource Identifier*), c'est-à-dire le chemin absolu pour la retrouver sur Internet ou sur un disque dur local. Ainsi, on dispose des descripteurs d'images pour comparer les images et on va les chercher seulement si on a besoin de les afficher par exemple. Cela diminue sensiblement la taille de la base de données et donc la vitesse d'accès aux informations stockées dans celle-ci.

Le problème de recherche d'information a amené la création de la norme spécifique MPEG-7 [6] qui définit des descripteurs d'images dans les contenus audio-visuels et permet une recherche par similarité de scènes dans les vidéos. Cette norme fournit aussi un langage et des outils de création et de gestion de schéma de description de contenu audio-visuel. MPEG-7 n'est pas une norme de codage de vidéo comme MPEG-2 et MPEG-4 mais un standard de description de contenu qui permet de décrire des vidéos mais aussi du texte, des sons, des images fixes. MPEG-7 permet par exemple de décrire le contenu de vidéos MPEG-2 ou MPEG-4 pour une recherche plus facile. MPEG-7 est basée sur XML, la norme est aussi extensible, elle permet d'utiliser ses propres descripteurs.

3. L'interprétation des images

Les méthodes de recherche existantes sont basées sur les informations contenues dans l'image. On peut classer ces informations selon un modèle à plusieurs niveaux sémantiques. La *figure 1.1* montre les différents niveaux, de la valeur la plus basse : les pixels de l'image à la valeur la plus haute : la description de la scène. Les pixels de l'image participent à

l'interprétation **bas-niveau** de l'image alors que la description de la scène correspond à l'interprétation **haut-niveau** de l'image.

La **segmentation d'images** permet de trouver les différentes régions qui composent une image. De nombreuses techniques de segmentation [BOL 95] ont été mises au point et offrent un début de solution au problème de reconnaissance de scènes.

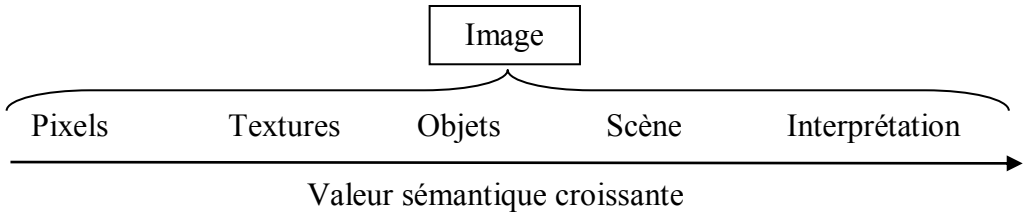


Figure 1.1 : Les niveaux sémantiques dans une image.

Lorsque nous regardons une image, nous pouvons analyser son contenu grâce à nos connaissances. En effet, si on nous présente une image de paysage de montagne, nous reconnaîtrons immédiatement la montagne, les arbres, le chalet. Maintenant, si on nous présente la photo de l'oursin (*figure 1.2*), nous verrons une forme sphérique (car nous percevrons le relief) sans pour autant savoir de quoi il s'agit. On pourrait penser à une vue grossie d'une bactérie, d'un virus ou encore à la vue de dessus d'un champignon. Car nous utilisons notre connaissance pour identifier le contenu d'une image. En revanche, personne n'y verra jamais une voiture, un stylo ou un chat.

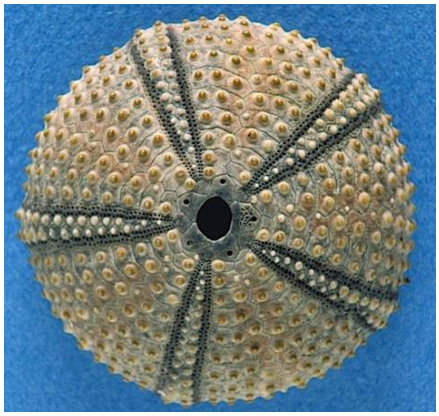


Figure 1.2 : Oursin.

Nous utilisons toujours nos connaissances pour décoder une image. Le problème est d'arriver à simuler le comportement humain à l'aide d'un ordinateur et de trouver des algorithmes capables de reconnaître des objets dans une image. Il existe de nombreuses

méthodes de reconnaissance de formes mais aucune technique actuelle ne permet de reconnaître complètement une scène et les objets qui la composent.

3.1. La sémantique

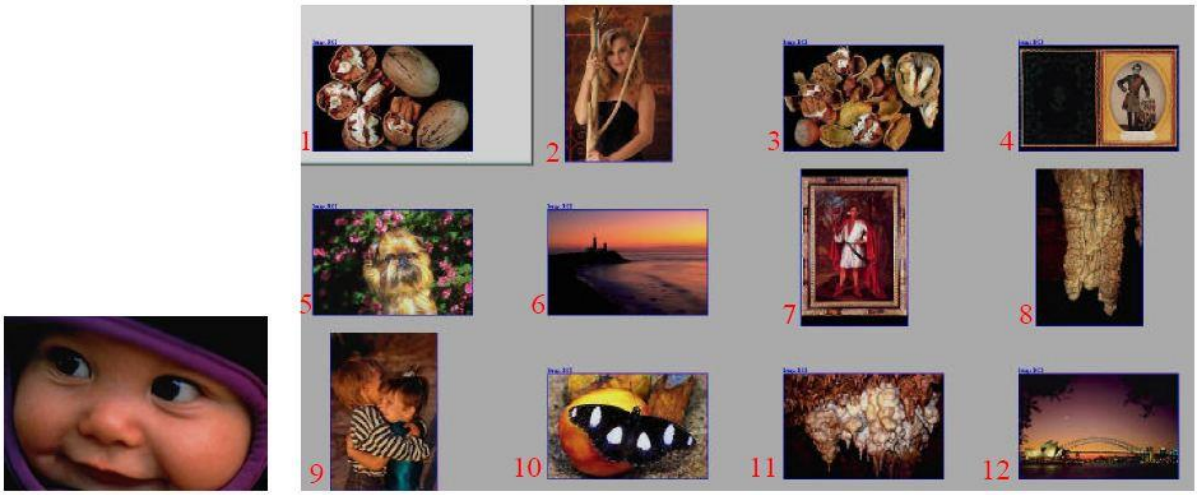
La difficulté dans les systèmes de recherche d'images par le contenu est d'associer une valeur sémantique à une image. À partir des pixels qui représentent une information **bas-niveau**, il est très difficile d'arriver à l'interprétation **haut-niveau** de l'image. La *figure 1.1* montre à quel point ce pas est difficile à franchir puisqu'à l'heure actuelle, reconnaître si un certain animal est présent ou non dans une image est encore un problème difficile à résoudre.

Dans l'étape de segmentation, les pixels sont associés dans des régions de différentes textures. Ces textures définissent les objets qui composent la scène conduisant à l'interprétation de l'image. Ainsi, donner un sens à une image signifie qu'à partir d'une suite de pixels, on va être capable de définir les objets présents dans la scène. Or il n'existe pas de technique de reconnaissance capable de recréer ce processus d'analyse qu'un enfant de quatre ans arrive à faire au premier coup d'œil.

3.2. Le fossé sémantique

Une façon simple afin d'introduire la notion de fossé/vide sémantique est de présenter les résultats de recherches visuelles bas niveaux, c'est-à-dire uniquement basées sur des caractéristiques primaires extraites du contenu des images.

La *figure 1.3* présente les résultats d'une requête portant sur les « portraits d'enfants ». La base de données utilisée contient environ 6000 images variées. Les résultats correspondent aux 12 images les plus « proches » de l'image proposée en exemple. Les caractéristiques extraites des images sont relatives à la couleur et à la texture. On constate que la sémantique des images résultats est très différente de la sémantique de l'exemple. Le système nous retourne une seule image dont la sémantique est proche de celle recherchée (image numéro 9). Ceci signifie que les caractéristiques primaires ne traduisent pas la notion de sémantique contenue dans l'image [FOU 02].



(a) Image exemple (b) Les 12 images les plus similaires à la requête

Figure 1.3 : Illustration 1 du fossé sémantique.

La *figure 1.4* présente les résultats d'une requête de « coucher de soleil » à partir de caractéristiques de couleur. La base contient toujours 6000 images. Ces résultats montrent que la couleur est un attribut discriminant pour les photographies de coucher de soleil. Bien qu'évident, ce fait confirme que certaines requêtes peuvent être satisfaites malgré une approche bas niveau du problème, à condition que des attributs discriminants soient utilisés [FOU 02].

(a) Image exemple (b) Les 12 images les plus « proches »

Figure. 1.4 : Illustration 2 du fossé sémantique.

On s'aperçoit que les résultats de la recherche correspondent bien à des images visuellement similaires mais aussi, on obtient des images qui ne sont plus visuellement similaires et qui n'ont plus rien à voir avec la requête. Ce phénomène est connu sous le nom de **fossé/vide sémantique (*semantic gap*)**.

Ces observations nous amènent à la définition du fossé/vide sémantique donnée par Smeulders et al. [SME 00]:

"*The semantic gap is the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation*".

"*Le fossé sémantique est le manque de coïncidence entre les informations que l'on peut extraire des données visuelles et l'interprétation que les mêmes données ont pour un utilisateur dans une situation donnée*".

L'image est polysémique et ses différentes significations sont à la fois contextuelles, différentielles et révélées par l'action. En premier lieu, la sémantique est contextuelle car elle dépend de l'utilisateur et des conditions particulières dans lesquelles est effectuée la recherche (mode d'interrogation du système, résultats des recherches précédentes, …). En second lieu, l'interprétation de l'image est différentielle, c'est-à-dire qu'elle est manifeste par la comparaison à des images de sémantique identique et d'autres de sémantique différente. En troisième lieu, le sens de l'image est révélé par l'action car du point de vue système, l'interaction utilisateur aide à réduire le fossé sémantique en précisant le but de la recherche. En d'autres termes, la sémantique de l'image est une propriété complexe qui n'est pas propre au signal mais qui peut cependant émerger de l'expertise que l'utilisateur renvoie au système. Il s'agit alors de mettre à jour les caractéristiques visuelles pertinentes.

Il est très difficile de trouver des descripteurs qui permettent de prendre en compte d'une part toutes les représentations possibles d'une même scène et d'autre part la subjectivité des requêtes formulées par les utilisateurs. Dans le domaine de la recherche d'images, on doit tenir compte du fossé/vide sémantique et trouver des techniques permettant de le combler au moins en partie.

Dans notre première contribution : *Indexation et recherche visuo-textuelle des bases de données images* (voir chapitre 4), nous avons proposé un système de recherche *INDEX* qui combine en même temps entre la recherche textuelle et la recherche par contenu pour combler au maximum le problème de vide/fossé sémantique.

Dans notre deuxième contribution : *Efficient Platform as a Service (PaaS) Model on public Cloud for CBIR System* (voir chapitre 4), nous avons proposé une plateforme pour diminuer le problème de vide/fossé sémantique.

3.3. La taille de la base de données

Le travail de recherche dans des bases d'images de plus d'un million d'images est une problématique différente avec des contraintes très fortes en matière de temps de calcul et de pertinence des résultats.

Il y a une autre difficulté à surmonter lorsqu'on travaille avec des descripteurs d'images, c'est celle de la dimension de ces vecteurs. En effet, dans la littérature, les descripteurs extraits des images sont la plupart du temps des vecteurs de taille $n > 100$. Ceci pose le problème de la **malédiction de la dimension (*dimensionality curse*)**.

En effet, les espaces de grandes dimensions possèdent des propriétés mathématiques particulières qui affectent le comportement des méthodes manipulant des données dans ces espaces.

Ainsi, la notion de distance en dimension deux ou trois n'a rien à voir avec la notion de distance dans un espace de dimension 100.

Ces différentes raisons entraînent qu'il faut tester les méthodes employées sur de grandes bases d'images et qu'il faut réduire la dimension des vecteurs descripteurs pour pouvoir donner un sens à la notion de distance qu'on souhaite utiliser.

3.4. Classement des images

Une solution au problème de la taille de la base est d'appliquer la technique «diviser pour mieux régner» (*divide and conquer*) qui consiste dans ce cas précis à diviser la base constituée de millions d'images en regroupements d'images plus petits organisés en familles d'images visuellement similaires.

Ce découpage de la base d'images permet de définir une structure en couches dans lesquelles les images sont regroupées en familles. En terme informatique, cela consiste à construire un arbre (*figure 1.5*) dont les feuilles sont des images et dont les nœuds représentent des familles d'images.

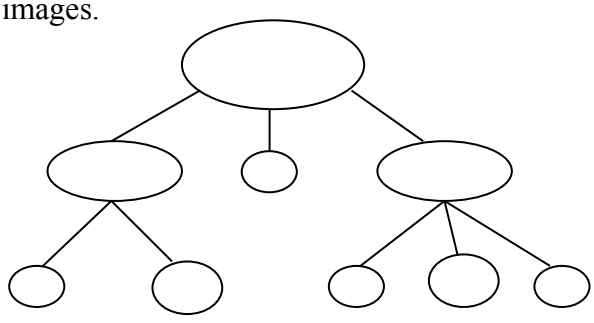


Figure 1.5 : Arbre de recherche qui découpe la base de données.

Le découpage de la base d'images dans la phase hors-ligne permet de ne travailler que sur un sous-ensemble de la base de données et donc d'appliquer avec une grande chance de succès les méthodes de recherche d'images sur ces «petites» bases.

On peut parcourir l'arbre toujours de la même façon, c'est-à-dire utiliser toujours le même critère de recherche au cours de l'exploration. Toutefois une meilleure approche est de définir en plus de la structure arborescente de la base de données une hiérarchie des signatures.

Dans notre contribution : *Efficient Platform as a Service (PaaS) Model on public Cloud for CBIR System.* (Voir chapitre 4), nous avons appliqué la technique «diviser pour mieux régner» pour construire notre plateforme.

4. Principe général de l'indexation et de la recherche d'image

Le problème de l'indexation d'image dans une base de données est complexe. L'index doit être conçu pour faciliter la recherche d'images. Les critères de recherche sont toujours dépendants du domaine d'application or chaque utilisateur a ses propres critères de pertinences, en fonction de ses besoins. Il n'existe donc pas une solution unique à ce problème. On distingue deux approches principales pour définir la requête. La première est la recherche par caractéristiques, où l'utilisateur "décompose" son but et décrit, à travers des primitives, ce qu'il recherche. La seconde approche est la recherche par l'exemple : l'utilisateur donne une image en requête et le système cherche les images les plus similaires au sein de la base de données.

4.1. Les différentes approches de recherche d'images

Les systèmes de recherche d'images existants adoptent l'une des trois grandes approches suivantes: l'approche textuelle qui se base sur l'indexation textuelle et les requêtes à base de mots clés, l'approche par le contenu qui s'appuie sur le contenu visuel des images durant l'indexation alors que la requête est soit une image ou un croquis et l'approche à base des ontologies qui cible des bases homogènes relevant d'un domaine particulier.

4.1.1. Indexation et recherche textuelle d'images

Le texte est une source d'information sémantique qui peut être utilisée pour l'indexation et la recherche des images. Les mots-clés ou les phrases peuvent provenir de l'indexation manuelle ou de données multimédia liées à l'image [FOU 02]. L'indexation des

images par des mots, bien qu'incontournable, reste une gageure en raison de la multiplicité des entités présentes dans l'image. Plusieurs techniques d'indexation ont été présentées dans la littérature [AIG 96, LAN 08, TOL 06]. Une première approche de l'indexation d'images consiste à décrire le contenu visuel sous forme textuelle (utilisation des mots-clés). Ces mots-clés servent comme index pour accéder aux données visuelles associées. L'avantage de cette approche est qu'elle permet de consulter les bases de données en utilisant les langages d'interrogation standard, par exemple SQL.

Cependant ceci nécessite une grande quantité de traitement manuel. De plus la fiabilité des données descriptives n'est pas assurée : elles sont subjectives et elles pourraient ne pas décrire correctement le contenu de l'image. Ainsi les résultats de recherche pourraient ne pas être satisfaisants [QUE 08].

Dès les premières tentatives de description d'une photographie, nous nous trouvons confrontés à l'infinité des éléments qui la composent. Il semble que la charge sémantique d'un mot et la charge sémantique d'une photographie soient incomparables. Définir une image à partir d'un seul terme s'apparente à une véritable réduction de l'image, d'autant plus que le terme qui a un sens pour moi, ne sera pas forcément celui partagé par la majorité. En effet, une image est le support d'une multitude d'interprétations possibles.

Le domaine de recherche et d'indexation d'images est de plus en plus actif. La problématique de la recherche d'images se résume en la recherche des mots, et ce, en se basant sur les attributs textuels des images tels que le nom des fichiers par exemple. Cette approche nécessite une entrée manuelle des mots définissant l'image (légende) et ne peut donc être appliquée au flux, toujours croissant, d'arrivée de nouvelles images.

Dans la première génération des systèmes d'indexation, les images sont représentées par des termes sémantiques de type mots-clés comme par exemple, Google, Yahoo.

A. Indexation textuelle manuelle

L'indexation textuelle manuelle d'images est le plus souvent réalisée par un documentaliste appelé iconographe. Son rôle est de classer et d'indexer les images en les associant à des catégories et à des groupes de mots, souvent extraits d'un thésaurus, permettant de retrouver facilement les images. Son travail est très utile pour les agences de presse, les centres de documentation, les musées... Du fait de l'accroissement du nombre de

photographies personnelles, ce travail est aussi souvent réalisé par les utilisateurs qui souhaitent décrire leurs images personnelles [TOL 06].

B. Indexation textuelle automatique

L'indexation textuelle automatique est effectuée par un système informatique sans aucune intervention humaine. L'indexation automatique des images est une opération nécessaire pour les images destinées au grand public, comme dans le cas du web, même elles sont déjà annotées manuellement. Les parties des moteurs de recherche qui sont responsables de cette opération utilisent le titre, les mots les plus fréquents et pertinents et même les métadonnées de la page où figure l'image [TOL 06]. Une autre façon d'indexation automatique est ce qu'on appelle auto-annotation qui se base sur des méthodes de classification supervisée utilisant un ensemble d'apprentissage où les images sont associées aux classes de mots pour apprendre à prédire des mots sur des nouvelles images [MAW 17, TOL 06].

L'indexation textuelle se fait en termes d'objets, de contenu et de structure. La sémantique elle-même n'est pas inscrite dans l'image, mais se trouve ailleurs. Il faut donc rechercher ces sources extérieures pour avoir accès aux clés de décodage sémantique de l'image.

Dans une image, la sémantique dépend de deux principes:
- du niveau de connaissances et de la perception que possède l'observateur,
- de l'objectif poursuivi par l'utilisateur de cette image lorsqu'il la regarde.

4.1.2. Indexation et recherche d'images à base d'ontologies

La méthode de la recherche d'images consiste à utiliser les ontologies. Cette méthode vise à cerner la sémantique des images en permettant à la fois un niveau important d'expressivité et de calculabilité. Le niveau d'expressivité élevé est assuré par la richesse des concepts de l'ontologie alors que le niveau de calculabilité haut est assuré par le raisonnement sémantique à travers des moteurs d'inférence sur les concepts de l'ontologie. Cette méthode, qui s'inscrit dans le cadre du web sémantique, vient d'être appliquée dans les moteurs de recherche. Une ontologie est vue comme un ensemble structuré de concepts et des relations entre ces concepts destinés à représenter les objets du monde sous une forme compréhensible aussi bien par les êtres humains que par les machines. Une ontologie visuelle est utilisée alors pour exploiter les caractéristiques de bas niveau de l'image. Elle contient un ensemble de

concepts qui permettent de décrire de manière qualitative l'apparence visuelle des concepts sémantiques.

La faiblesse de l'approche de recherche d'images à base d'une ontologie est qu'elle est destinée pour être utilisée dans un domaine particulier. Cette approche est impraticable pour les collections d'images qui ne sont pas spécialisées ou les corpus dont le domaine est inconnu [MAW 17].

4.1.3. Indexation et recherche d'images par le contenu visuel

L'archivage des images et des séquences vidéos aussi bien pour les chaînes de télévision, les journaux, les musées, voire même pour les moteurs de recherche sur Internet ne se fait qu'au prix d'une étape d'annotation manuelle à l'aide de mots-clés. Cette indexation représente une tâche longue et répétitive pour l'humain, surtout avec les bases d'images qui deviennent aujourd'hui de plus en plus grandes. De plus, cette tâche est très subjective à la culture, à la connaissance et aux sentiments de chaque personne. Le besoin de méthodes d'indexation et de recherche directement basé sur le contenu de l'image n'est donc plus à démontrer. Le premier prototype de système fut proposé en 1970 et attira l'attention de la communauté des chercheurs. Les premiers systèmes d'indexation d'images par le contenu sont crées au milieu des années 90 pour des bases de données fournies, spécialisées et pour la plupart fermées.

Les systèmes d'indexation et recherche d'images par le contenu (CBIR) (*Content-Based Image Retrieval systems*) permettent de rechercher les images d'une base d'images en fonction de leurs caractéristiques visuelles. Ces caractéristiques, encore appelées caractéristiques de bas-niveau sont la couleur, la texture, la forme et tout autre caractéristique de l'image qu'on peut imaginer.

Dans notre contribution : *Indexation et recherche visuo-textuelle des bases de données images.* (Voir chapitre 4), nous avons combiné entre l'indexation textuelle manuelle et l'indexation par contenu visuelle pour développer notre système de recherche *INDEX*.

5. Etat de l'art des systèmes CBIR

À partir des années 90, les chercheurs dans le domaine de la vision par ordinateur se posent le problème de l'indexation automatique des images par leur contenu, qui permet la recherche d'images par le contenu (CBIR: Content-Based Image Retrieval). La CBIR cherche une solution alternative semi-automatique au problème de la reconnaissance d'objets en

évitant toute interprétation haut-niveau de la scène. Elle se fonde uniquement sur la ressemblance numérique d'images [QUE 08].

Par la suite, nous décrirons quelques travaux utilisant le système CBIR.

Les auteurs de [DMI 09] ont proposé un système basé sur le Web pour la récupération d'images médicales qui utilisait les fonctionnalités d'Oracle Multimédia pour la récupération d'images. L'objectif principal était de montrer l'utilisation des descripteurs Oracle Multimédia pour la gestion et la récupération des données. Dans ce travail, Dimitrovski et al. Manipulé le format DICOM, qui pourrait contenir des informations supplémentaires concernant la modalité d'image, l'identification du patient et les données d'image brutes .... Dimitrovski et al. Combine à la fois des données basées sur le contenu et des données textuelles. Pour calculer la similitude entre l'image requête et les images dans la base de données, ainsi que pour extraire les caractéristiques d'une image telles que la couleur, la texture et la forme, ils ont utilisé une solution basée sur les méthodes fournies par Oracle. Dans ce travail, les auteurs n'ont pas indiqué les méthodes qu'ils ont utilisées pour l'extraction des caractéristiques (couleur, texture et forme), ni la méthode utilisée pour mesurer la similitude.

Les auteurs de [RAM 13] ont proposé un système CBIR pour une base de données d'images médicales. Le CMBIR assiste les pathologistes dans le diagnostic du patient. La technique d'extraction des caractéristiques de texture basée sur les bords est utilisée pour extraire la texture. Le descripteur d'histogramme de bord est utilisé pour extraire la forme. Les auteurs ont combiné les deux techniques pour produire un seul vecteur caractéristique. Les auteurs ont utilisé la formule de distance euclidienne pour calculer la similitude entre l'image requête et les images de la base de données. L'objectif principal du système était d'améliorer l'efficacité de la récupération d'image par rapport à l'utilisation d'une seule fonction. Parce que ce système combinait la forme et la texture, il fournissait environ 33% - 91% de précision et 33% - 75% de rappel.

Dans les travaux présentés dans [KUM 16], les auteurs ont proposé un système CBIR utilisant des images médicales. Pour mettre en œuvre ce système, les auteurs ont commencé par l'extraction des caractéristiques telles que la texture et l'intensité. Ils ont utilisé l'algorithme de modèle binaire local (LBP : Local Binary Pattern) pour l'extraction de texture.

Les auteurs ont concentré leurs travaux sur les images de l'œil. Pour améliorer la récupération, ils ont également utilisé la distance euclidienne pour calculer les similitudes entre l'image requête et les images de base de données. Parce que ce système combinait la couleur et la texture, il offrait une précision d'environ 40,86% - 89,18% et un rappel de 34,86% - 78,12%.

Dans [KUS 16], les auteurs ont proposé d'extraire les caractéristiques de l'image pour effectuer une classification à l'aide de k-moyennes et d'un algorithme d'arbre de décision. Les auteurs ont utilisé les moments statistiques pour la couleur et la matrice de cooccurrence de Haralick pour la texture. Pour classer une image requête, les auteurs ont extrait ses fonctionnalités et les ont comparées avec les règles construites dans le processus de formation. Malheureusement, cette méthode n'a pas pu récupérer des images similaires.

Les auteurs de [MEE 13, MEE 16] ont proposé un système CBIR implémenté sur le Cloud en tant que services SaaS à l'aide de la plate-forme Windows Azure. Les auteurs ont utilisé la transformée en ondelettes pour extraire les caractéristiques numériques d'images telles que la couleur, la texture et la forme. Les auteurs ont proposé une méthode hybride, appliquant la transformée de Kekre pour les propriétés globales de l'image, et les transformées de Haar, Walsh, DCT et Hartley pour les propriétés locales de l'image. La combinaison de toutes ces transformations a généré de bons résultats. La similitude entre l'image de base de données et l'image requête est réalisée en utilisant la distance euclidienne.

Dans [PAI 18] Paiz-Reyes et al., Ont proposé la récupération d'images GIF (Graphics Interchange Format) dans un environnement informatique Cloud. La collecte des données est classée manuellement en quatre catégories. Les auteurs ont utilisé l'histogramme de couleurs 3D dans l'espace colorimétrique HSV (Hue Saturation Value) pour le descripteur d'image basé sur les couleurs, le LBP (Local Binary Pattern) est appliqué pour le descripteur basé sur la texture et les moments Zernike sont appliqués pour le descripteur basé sur la forme. Les vecteurs d'entités sont enregistrés dans une table de hachage sous forme de dictionnaire. LSH (Locality Sensitive Hashing : hachage sensible à la localité) est utilisé pour l'indexation. Le LSH vise à maximiser la probabilité de "collision" pour des éléments similaires. La distance euclidienne est utilisée pour trouver des images similaires.

Les auteurs de [MAH 19] ont proposé un système CBIR implémenté sur le Cloud en utilisant la plate-forme GPU (Graphical Processing Unit). L'API CUDA (Compute Unified Device Architecture) (Application Programming Interface) est utilisée pour exploiter les cartes NVIDIA. Le framework OpenCL (Open Computing Language) est utilisé pour exploiter les cartes graphiques ATI (Array Technologies Incorporated) / AMD (Advanced Micro Devices). La combinaison des deux descripteurs SIFT (Scale Invariant Feature Transform) avec la SURF (Speeded up Robust Feature) fournit de bons résultats pour l'extraction des caractéristiques de l'image, car ces deux approches sont les plus utilisés pour détecter et faire correspondre les caractéristiques de l'image. Les méthodes SIFT et SURF sont invariantes en termes de rotation, de mise à l'échelle, d'éclairage et de translation, ….. Concernant la comparaison, Mahmoudi et al., Utilisent Matcher BF (Brute Force) et Matcher FLANN (Fast Library for Approximate Nearest Neighbors). Pour trouver des images similaires, l'algorithme KNN (K-Nearest Neighbors) est utilisé.

6. Classification

Nous présentons une classification, illustrées dans le *tableau 1.1*, des travaux décrits dans l'état de l'art des systèmes CBIR. Chaque proposition a son propre principe de fonctionnement, objectif et méthodologie.

7. Description du CBIR

Les méthodes d'indexation comprennent en général les éléments suivants [QUE 08] :

– Une signature ou index du document, qui sert comme caractéristique pour le reconnaître et le comparer avec les autres. Dans le cas de la recherche par caractéristiques, celles-ci ou bien des dérivées de ces caractéristiques sont prises directement comme signatures. Quant à la recherche par l'exemple, il faut déterminer les signatures les plus appropriées pour décrire le contenu des documents d'une façon approchant le mieux possible les critères de l'utilisateur.

– Une métrique de similitude (ou de distance) qui permet de comparer les signatures et d'associer les documents similaires.

– Des algorithmes de recherche qui, basés sur les deux outils précédents, permettent de retrouver rapidement les objets recherchés. Une approche itérative est parfois proposée : à

partir d'une première sélection de documents, on précise les critères, on relance la requête, et ainsi de suite.

– Une interface utilisateur, qui rend transparente la procédure de recherche et facilite l'introduction de la requête.

Les experts humains sont les mieux placés pour construire la signature des documents. Cependant, cette opération est très coûteuse et difficilement réalisable, étant donnée la taille énorme des bases d'images. D'où l'intérêt de l'indexation automatique.

			Meena et al.[MEE 13, MEE 16]	Kusrini et al. [KUS 16]	Paiz-Reyes et al [PAI 18]	Mahmoudi et al. [MAH 19]	Ramamurthy et al. [RAM 13]	Kumar et al. [KUM 16]	Dimitrovski et al.[DIM 09]
Content-based			X	X	X	X	X	X	X
Text-based					X				X
QBE(Query By Example)			X	X	X	X	X	X	X
Global properties of the image	Color	Intensity			X	X		X	Extraction of the attribute without any detail
		Histogram	X			X			
		Color moments (μ, σ,θ)		X					
	Texture	Matrix of co-occurrences		X					Extraction of the attribute without any detail
		LBP			X			X	
		Wavelet transform	X			X			
		Edge based feature extraction					X		
Local properties of the image	Division of an image in the region								
	Segmentation				X				
	Wavelet transform		X			X			
Shape	Boundary-based descriptor						X		Extraction of the attribute without any detail
	Region-based descriptor				X				
Medical images		DICOM							X
		Other					X	X	
General images			X	X	X	X			
Similarity measure	Euclidean distance		X		X		X	X	Without any detail
	K-NN (K- Nearest Neighbor)					X			
Cloud	SaaS		X			X			
	PaaS								
	IaaS								

Tableau 1.1 : Classification des travaux décrits dans l'état de l'art des systèmes CBIR.

7.1. Architecture CBIR

Un système CBIR classique se compose de deux étapes, la phase hors ligne et la phase en ligne comme illustré à la *figure 1.6*.

1) La phase hors ligne (La phase d'indexation)

Dans cette phase, les caractéristiques (couleur, texture et forme) sont automatiquement extraites de l'image et stockées dans un vecteur numérique en tant que «vecteurs de caractéristiques/descripteurs». Durant cette phase, aucun utilisateur n'est connecté au système. Cette phase peut prendre le temps nécessaire pour calculer les descripteurs.

2) La phase en ligne (La phase de recherche)

Dans cette étape, le système analyse une ou plusieurs demandes émises par l'utilisateur. Le système fournit ensuite le résultat sous la forme d'une liste d'images ordonnées, triées en fonction de la similitude entre leurs vecteurs de caractéristiques et les caractéristiques équivalentes de l'image requête. Durant cette seconde étape, le temps de réponse du système est crucial, il faut le réduire au maximum.

Nous identifions trois types de modèles d'indexation : le modèle booléen, le modèle vectoriel et le modèle probabiliste [FOU 02].

Le modèle booléen : les images sont caractérisées par une liste de descripteurs et la requête est une formule logique qui combine à la fois des descripteurs exemples et des opérateurs logiques du type ET, OU, NON, …. En réponse, le système effectue une classification en deux classes correspondant d'une part aux images qui satisfont la requête et d'autre part à celles qui ne la satisfont pas.

Le modèle vectoriel : l'image requête et les images cibles, c'est-à-dire les images de la base, sont représentées par un vecteur dans un espace d'attributs. Ce vecteur correspond à la concaténation des pondérations sur des descripteurs basiques. Une fonction de similarité (ou de dissimilarité) entre vecteurs est utilisée pour classer les images en fonction de leur adéquation à la requête.

Le modèle probabiliste : une probabilité de pertinence de l'image, en réponse à la requête, est attribuée à chacun des descripteurs. Sous hypothèse d'indépendance des descripteurs, il est possible de calculer la probabilité que l'image réponde à la requête de l'utilisateur comme le produit des probabilités précédentes. Ce type de modèle fait appel à une interaction utilisateur

forte par l'intermédiaire, par exemple, de jugements de pertinence émis par l'utilisateur sur les propositions de résultats faites par le système.

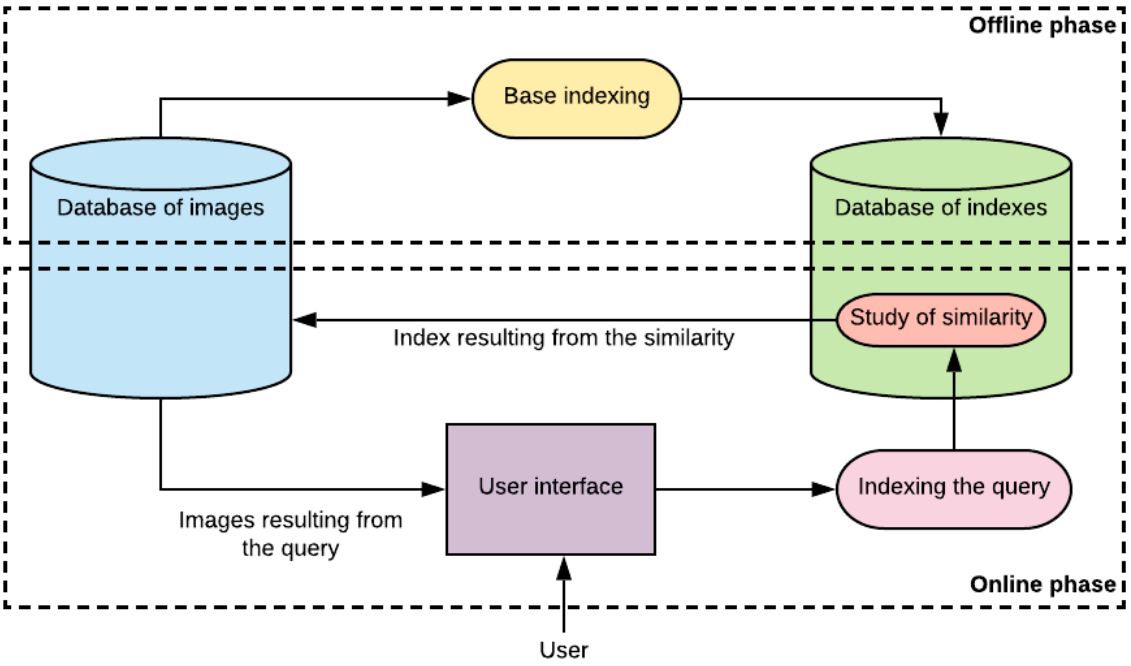


Figure 1.6 **:** Architecture CBIR.

7.1.1. Du côté de l'utilisateur

L'aspect utilisateur est très important dans la création d'un système de recherche d'images. L'utilisateur doit disposer d'un programme de recherche d'images simple d'utilisation, rapide et efficace. Ces contraintes doivent être rigoureusement respectées si on veut fournir un bon système de recherche d'images.

On distingue deux types d'utilisateurs d'une base d'images :

– Le **non-spécialiste** qui cherche une image sans avoir de connaissance particulière du domaine de la base,

– L'**expert** qui connaît parfaitement le domaine de la base d'images et qui connaît (ou qui veut tester) les attributs à utiliser pour classer la base selon ses critères.

Il faut donc adapter le système à des utilisateurs experts ou non du domaine de la collection. C'est-à-dire qu'un utilisateur naïf demandera simplement à rechercher une image dans la base alors qu'un expert pourra modifier les paramètres afin de prendre en compte sa connaissance du domaine.

Dans notre contribution, *Indexation et recherche visuo-textuelle des bases de données images.* (Voir chapitre 4), nous avons proposé un système de recherche *INDEX* pour des

utilisateurs naïfs. Mais, Dans notre deuxième contribution, *Efficient Platform as a Service (PaaS) Model on public Cloud for CBIR System.* (Voir chapitre 4), nous avons proposé une plateforme pour aider les radiologues à diagnostiquer la maladie, c'est-à-dire, notre proposition est destinée aux utilisateurs experts.

7.1.2. Les différentes méthodes de formulation de requêtes

Le contenu visuel des images de la base qui est extrait et décrit est appelé *signatures d'image*. Les signatures des images de la base constituent une *base de signatures*. Pour rechercher des images, l'utilisateur fournit une image exemple (appelée *requête*). Différentes méthodes de formulation de requêtes existes, le traitement de celles-ci dans le système dépend de la manière dont on lui présente l'information. Un modèle de recherche spécifie le mode de représentation de la requête. Nous listons ci-dessous les types de méthodes d'interrogation d'images qui peuvent être classifiés en six catégories [BEN 17]:

• **Requête sur une seule caractéristique** : la sélection des images est basée sur une seule caractéristique, les pourcentages de différentes valeurs prises sont déterminés par l'utilisateur. Un exemple de ce type de requête est de trouver les images contenant 10% de rouge, 30% de vert et 60% de bleu.

• **Requête simultanée sur plusieurs caractéristiques** : la requête utilisateur est une combinaison de plusieurs caractéristiques (couleur, texture et forme). Un exemple de ce type de requête est de trouver les images contenant 10% de rouge et 30% de vert et 60% de bleu de la texture d'arbre.

• **Requête sur la localisation des caractéristiques** : l'utilisateur précise les différentes valeurs prises par les caractéristiques et l'emplacement de ces caractéristiques dans l'image. Un exemple de ce type de requêtes est de trouver les images dont 25% de la couleur rouge est situé à gauche de l'image.

• **Recherche d'image par croquis (query by sketch)** : l'utilisateur dessine sa requête à l'aide d'une interface graphique pour chercher les images qui lui ressemblent dans la base. On distingue deux types de dessin :

- **Le croquis:** l'utilisateur décrit ce qu'il désire en représentant précisément les contours des objets, généralement en une seule et unique couleur, seul l'aspect forme porte l'information.

- Le dessin grossier: un dessin coloré est proposé par l'utilisateur. Il contient une représentation colorée de chaque objet, mais ou (**le «mais ou» rend la phrase lourde**) les contours sont généralement vagues. L'information couleur (les couleurs mêmes mais aussi l'agencement de celles-ci) est donc primordiale, au contraire de la forme qui n'est pas très représentative.

• **Recherche d'images par objets** : l'utilisateur décrit les caractéristiques d'un objet d'une image plutôt que l'image entière, le but est de rechercher un objet précis dans une série d'images. Un exemple de ce type de requête est de trouver les images contenant une personne. On peut aussi combiner plusieurs objets et spécifier les relations spatiales entres les différents objets. Un exemple de ce type d'interrogation est de trouver les images ou une personne est assise à côté d'un arbre.

• **Recherche par l'exemple** : Dans ce cas, le système a besoin de comparer un exemple de même type (image) avec la base pour produire les images similaires. Cette méthode est simple naturelle et ne nécessite pas de connaissance approfondies pour manipuler le système. Elle est donc bien adaptée à un utilisateur non spécialiste.

Parmi ces différents cas de formalisation de la requête, nous avons choisi *la Recherche par l'exemple* comme méthode d'interrogation d'images dans nos deux contributions (voir chapitre 4).

7.2. Extraction des caractéristiques (descripteurs) d'images

La description du contenu des images est une étape essentielle dans les CBIR, car leur performance dépend principalement du choix des descripteurs employés et des techniques associées à leurs extractions. Un descripteur est défini comme la connaissance utilisée pour caractériser l'information contenue dans les images [BEN 17]. Le but n'est pas de coder toute l'information portée par l'image mais de se focaliser sur l'information qui permet de traduire efficacement une similitude proche des besoins exprimés par un utilisateur. Sans dissocier l'extraction des attributs (caractéristiques) de la structuration de l'information sous forme de signature et de l'utilisation de cette représentation pour la recherche, une des clés de l'indexation efficace est l'identification des caractéristiques primaires en accord avec le type et le but des recherches visées par le système. Il n'existe pas d'indexation généralisable à tout type d'application [FOU 02]. Cette étape permet de fournir une représentation du contenu de l'image appelé aussi signature de l'image.

7.2.1. Couleur

La couleur est très souvent le premier descripteur (fonction) utilisé pour la recherche d'images. Bien que la majorité des images soient dans l'espace colorimétrique RVB (Rouge, Vert et Bleu), d'autres espaces tels que HSV (Hue –Saturation –Value) ou les espaces CIELab et CIELuv sont bien meilleurs par rapport à la perception humaine et sont plus fréquemment utilisé [MUL 04]. Quelque soit l'objectif voulu, il faut toujours rechercher la représentation, c'est-à-dire l'espace couleur, qui sera le mieux adapté aux données et à l'algorithme que l'on souhaite utiliser [HOU 10]. Dans des domaines spécialisés, notamment dans le domaine médical, les caractéristiques de couleur ou de niveau de gris ont souvent un pouvoir expressif très limité [MUL 04].

De nombreux attributs de couleur, tels que les histogrammes, les moments, …, peuvent être utilisés pour caractériser la couleur.

7.2.1.1. Histogrammes de couleur

A un facteur de normalisation près, l'histogramme constitue une approximation de la densité de probabilité associée à l'image, vue comme une variable aléatoire. Cette approche statistique est proche de la recherche textuelle pour laquelle les documents sont couramment caractérisés par la fréquence d'apparition de mots-clés [FOU 02]. Un histogramme est un outil statistique qui permet d'estimer la densité de probabilité d'une distribution à partir d'échantillons. L'intervalle des valeurs possibles est divisé en classes, puis pour chacune d'elles on compte le nombre d'échantillons associés [HOU 10].

En 1991, Swain et Ballard [SWA 91] sont les premiers à utiliser l'histogramme pour l'indexation couleur. Par la suite, de nombreux auteurs ont suivi le pas et cette signature reste la plus utilisée dans le cadre de l'indexation. Parmi les travaux, on trouve les histogrammes flous [GRE 01, VER 00] dont l'objectif est de limiter les effets de la quantification des espaces d'attributs.

Les histogrammes de couleur sont faciles et rapides à calculer et robustes en ce qui concerne la rotation et la translation (voir équation 1.1) [SWA 91].

On a:

$c = I(i, j)$.

I: image $(M \times N)$ pixels.

c: couleur appartenant à l'espace colorimétrique C.

h: vecteur avec n composantes$(h_{c1}, h_{c2}, \dots, h_{cn})$.

h_{cj}: le nombre de pixels de couleur c_j dans image I.

$$\sum_{i=1}^{n} h_{ci} = MN \qquad (1.1)$$

Où MN est le nombre de pixels de l'image I.

L'utilisation d'histogrammes pour l'indexation et la recherche d'images pose des problèmes :

- Ils sont de grandes tailles, par conséquent il est difficile de créer une indexation rapide et efficace en les utilisant tels quels.
- Ils ne possèdent pas d'informations spatiales sur les positions des couleurs.
- Ils sont sensibles à de petits changements de luminosité, ce qui est problématique pour comparer des images similaires, mais acquises dans des conditions différentes.
- Ils sont inutilisables pour la comparaison partielle des images (objet particulier dans une image), puisque calculés globalement sur toute l'image.

7.2.1.2. Les moments de couleur

La méthode des histogrammes de couleurs utilise la distribution complète des couleurs et nécessite un espace élevé pour le stockage des données. Plutôt que de calculer la distribution complète, certains systèmes de recherche d'images n'utilisent que des caractéristiques de couleur dominantes telles que: la moyenne de c (symbolisé μ_c), l'écart type (σ_c) et l'asymétrie (θ_c). Ces caractéristiques sont calculées pour chaque composante de couleur par les formules suivantes: équation (1.2), équation (1.3) et équation (1.4), respectivement [DUB 10].

$$\mu_c = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} p_{ij}^{c} \qquad (1.2)$$

$$\sigma_c = \left[\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(p_{ij}^{c} - \mu_c\right)^2\right]^{\frac{1}{2}} \qquad (1.3)$$

$$\theta_c = \left[\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(p_{ij}^{c} - \mu_c\right)^3\right]^{\frac{1}{3}} \qquad (1.4)$$

Où *M* et *N* sont les tailles horizontale et verticale de l'image. Et p_{ij}^{c} : est la valeur de la couleur *c* dans la ligne *i* et la colonne *j* de l'image *I*.

7.2.1.3. Couleurs dominantes

L'utilisation d'histogrammes pour représenter la distribution de couleur présente quelques inconvénients. Du point de vue de l'espace mémoire, les histogrammes à plusieurs dimensions sont « creux », c'est-à-dire que la majorité des cellules ne comptent aucun pixel. Une grande partie de l'espace mémoire est utilisée inutilement. De plus, toutes les classes ont la même taille, alors qu'il serait plus intéressant d'avoir des classes plus petites dans les régions contenant des couleurs très fréquentes, et de grandes classes pour les couleurs moins répandues. Du point de vue des mesures de similarité employées, les mesures traditionnelles effectuent uniquement une comparaison cellule à cellule. Même si les histogrammes sont ordonnés, le voisinage des cellules n'est pas pris en compte quand elles ont des valeurs différentes [HOU 10]. Pour résoudre ces différents problèmes, Rubner et Tomasi [RUB 13] proposent Les signatures par couleurs dominantes. La signature $s = \{ s_i = (m_i, w_i)\}$ est un ensemble de nuages de points. Chaque nuage est représenté par son mode m_i (le mode d'un nuage de point correspond à un maximum local de sa densité de probabilité), et le nombre w_i de pixels qui appartiennent au nuage. Ma et Manjunath [MA 99] proposent de ne conserver de l'histogramme, que les couleurs qui occupent l'aire la plus importante dans l'image (les couleurs dominantes).

Contrairement aux histogrammes, ces signatures ne stockent que les couleurs qui appartiennent à l'image, elles ne stockent pas les cellules vides. Il existe plusieurs algorithmes pour détecter les couleurs dominantes : citons notamment la segmentation d'image, qui consiste à regrouper tous les pixels ayant une couleur proche, ou encore l'extracteur de couleurs dominantes proposée pour le descripteur de MPEG-7 par [DEN 01]. Ce dernier

effectue une classification des couleurs de l'image dans un espace perceptuellement uniforme (habituellement le système L*u*v*), à l'aide de l'algorithme de Lloyd généralisé [HOU 10].

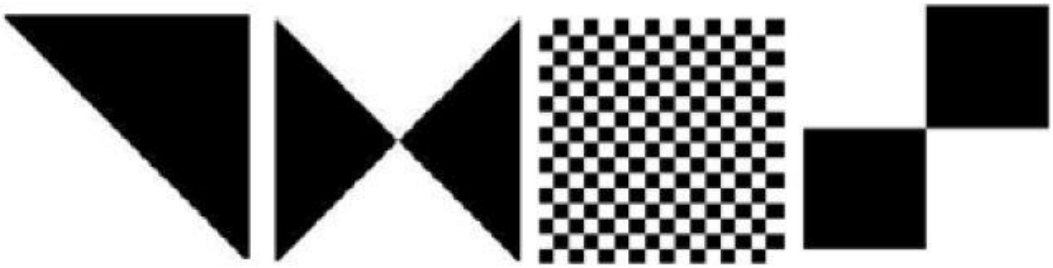


Figure.1.7 : Limite des histogrammes et des couleurs dominantes. Ces quatre images ont le même histogramme et les mêmes couleurs dominantes.

7.2.1.4. Corrélogrammes

Les histogrammes et les couleurs dominantes donnent une bonne estimation de la densité de couleur d'une image, mais ils ont l'inconvénient de perdre toute l'information sur la distribution spatiale des couleurs. Il est impossible de savoir, pour une couleur donnée, ou elle est située par rapport à une autre couleur, ou même de savoir si une couleur est repartie dans toute l'image ou si elle est présente uniquement dans une sous-partie. Ainsi, les images présentées sur la *figure 1.7* sont identiques d'après leur histogramme : chaque couleur a exactement le même nombre de pixels, c'est uniquement leur répartition spatiale qui change. Cette contrainte peut être adoucie, si l'on considère pour chaque couleur non plus le nombre de pixels ayant cette couleur, mais les probabilités pour les différentes couleurs qu'elles soient présentes, à différentes distances. Ces histogrammes améliorés sont appelés des *corrélogrammes* [HOU 10]. Les travaux proposés dans [ORT 97] [HUA 97] sont inspirés des matrices de cooccurrences [HAR 73].

On note $\mathcal{L}$ l'image à indexer, et n son nombre de lignes et de colonnes (on prend l'exemple d'une image carrée pour simplifier les notations). Le terme $\mathcal{L}_c$ représente l'ensemble des pixels ayant la couleur c, et pour un entier quelconque L, on représente par $[L]$ l'intervalle d'entiers de 1 à L. Si on quantifie l'espace de couleurs en m couleurs $c_1, c_2, \ldots, c_m$ la classe c_{ij} d'un histogramme normalisé peut être reformulée en termes de probabilité par : $h_{c_i}(\mathcal{L}) = pr_{p \in \mathcal{L}}(p \in \mathcal{L}_{c_i})$

Pour un pixel donné de l'image, $h_{c_i}(\mathcal{L})$ donne la probabilité que sa couleur soit c_i. Si on discrétise l'intervalle réel $[0,n]$ en d parties, alors pour $(i,j) \in [m]^2$ et $k \in [d]$,

Le corrélogramme est défini par :

$$\gamma_{c_i,c_j}^{(k)}(\mathcal{L}) = Pr_{p_1 \in \mathcal{L}_{c_i}, p_2 \in \mathcal{L}} \left[p_2 \in \mathcal{L}_{c_j} \middle| \, |p_1 - p_2| = k \right] \qquad (1.5)$$

Pour un pixel de couleur c_i de l'image, $\gamma_{c_i,c_j}^{(k)}(\mathcal{L})$ donne la probabilité qu'un pixel situé à la distance k soit de couleur c_j. Le corrélogramme a ainsi $m^2 d$ classes.

7.2.2. Texture

La texture est souvent modélisée comme une structure spatiale constituée de l'organisation de primitives ayant chacune un aspect aléatoire. Une texture peut avoir un aspect périodique ou bien aléatoire [FOU 02]. En raison de la définition imprécise de la texture, ses mesures possèdent une variété encore plus grande que les mesures de couleur. Les mesures de texture tentent de capturer les descripteurs de l'image ou des parties de l'image par rapport aux changements dans une certaine direction et à l'échelle des changements. Ces mesures sont spécialement utiles pour les régions à texture homogène. Les mesures de texture sont invariantes par rapport aux rotations d'image et aux changements d'échelle [MUL 04]. Une texture peut être caractérisée par les attributs de régularité, de contraste et de périodicité du motif. Dans le cadre de la recherche par le contenu, elle permet de distinguer des zones de couleurs similaires, mais de sémantique différente [BEN 17].

7.2.2.1. Méthodes d'analyse de la texture

Les méthodes d'extraction et d'analyse à base de texture peuvent être divisées en quatre classes [TUC 93] :

- *Les méthodes statistiques* analysent la distribution spatiale des valeurs de niveaux de gris par le calcul des indices locaux dans l'image. Les matrices de cooccurrence des niveaux de gris [HAR 73], est l'une des méthodes statistiques de segmentation à base de texture fréquemment citée.
- *Les méthodes géométriques* ou structurelles sont utilisées pour décrire les motifs complexes et déduire les propriétés des textures. Les descripteurs de texture peuvent être extraits, par exemple, par une différence de gaussiennes [TUC 90].
- *Les méthodes fréquentielles* analysent les fréquences de l'image. Les méthodes fréquentielles les plus utilisées sont la transformée de Fourier, les filtres de Gabor [MAN 96] et les décompositions en ondelettes [MAL 89].

- *Les méthodes à base de modèle* cherchent à caractériser statistiquement l'image au moyen de modèles probabilistes en considérant la texture comme une réalisation d'un processus aléatoire. Les méthodes à base de modèles probabilistes sont largement utilisées telles que les champs aléatoires conditionnels (CRF) [LAF 01], les champs aléatoires Markoviens (MRF) [NIC 05], les fractales [FER 03].

7.2.2.2. Représentation statistique des niveaux de gris

Cette catégorie d'approches vise à évaluer la distribution des niveaux de gris contenus dans une texture au moyen de descripteurs statistiques d'ordre deux calculés selon une distance et une orientation entre un pixel central et ses pixels voisins. Parmi les méthodes relevant de cette catégorie, on retrouve régulièrement les matrices de cooccurrence [REG 14].

Matrice de cooccurrence des niveaux de gris

- *Principe général*

Le calcul d'une matrice de cooccurrence de niveaux de gris ou GLCM (*Grey Level Co-occurrence Matrix*) consiste à repérer dans une image le nombre d'occurrences de paires de niveaux de gris séparés par une distance *d* dans une direction définie par un vecteur de déplacement *(dx,dy)* [HAR 73]. Le calcul de la GLCM pour une image *I* de dimension $N \times M$ se formalise comme suit :

$$GLCM_{dx,dy}(i,j) = \sum_{n=1}^{N}\sum_{m=1}^{M}\begin{cases}1, & si\ I(n,m) = i\ et\ I(n+dx,m+dy) = j \\ 0, & sinon\end{cases} \qquad (1.6)$$

Où *i* et *j* sont les niveaux de gris du pixel de référence et du pixel voisin respectivement, *n* et *m* correspondent aux coordonnées des pixels dans l'image *I*.

En pratique, les entrées de la GLCM sont souvent normalisées en les divisant par le nombre de pixels (somme des entrées égale à 1) afin d'éliminer l'influence de la taille de l'image. En outre, la matrice de cooccurrence est généralement construite de manière symétrique en considérant la somme des entrées calculées pour deux déplacements de directions opposées [HAR 73]. Ainsi, pour un déplacement horizontal sur une distance de 1, on intégrera simultanément les occurrences obtenues pour deux vecteurs de déplacement *(dx,dy)* égal à (1,0) et à (0,1).

Bien que les GLCMs fournissent une description riche de la dépendance spatiale, il est peu pratique de les manipuler sous leur forme brute. En 1973, Haralick [HAR 73] a proposé 14 caractéristiques statistiques extraites d'une matrice de cooccurrence. La texture d'une image peut être interprétée comme la régularité d'apparition de couples de niveaux de gris selon une distance donnée dans l'image. La matrice de cooccurrences contient les fréquences spatiales relatives d'apparition des niveaux de gris selon quatre directions :

$$\theta = 0\,, \qquad \theta = \frac{\pi}{4}\,, \qquad \theta = \frac{\pi}{2}\,, \qquad \theta = \frac{3\pi}{4}\,.$$

Une matrice de cooccurrences est définie au moyen d'une relation géométrique $\Re$ entre deux pixels (x_1,y_1) et (x_2, y_2).

La matrice de cooccurrences $P_{d,\theta}(i,j)$ est carrée et de dimension $\Delta*\Delta$, où Δ est le nombre de niveaux de gris présents dans I. Les indices de la matrice de cooccurrences sont donc les niveaux de gris de la texture étudiée.

On définit la matrice de cooccurrences :

$$P_{d,\theta} = P_{d,\theta}(i,j) \qquad (1.7)$$

$P_{d,\theta}(i,j)$ représente le nombre de fois où un couple de points séparés par la distance d dans la direction θ ayant les niveaux de gris I_i et I_j. Pour obtenir de véritables fréquences relatives, il faut normaliser les éléments de la matrice en les divisant par le nombre total de paires de points élémentaires séparés par la distance d dans la direction θ dans toute l'image.

Actuellement, seules les quatre caractéristiques (descripteurs) les plus appropriées sont largement utilisées. L'énergie, l'entropie, le contraste et le moment de différence inverse (IDM) (homogénéité) sont calculés par les formules suivantes équation (1.8), équation (1.9), équation (1.10) et équation (1.11), respectivement.

$$ENE = \sum_{i=1}^{M}\sum_{j=1}^{N}\left(P_{d,\theta}(i,j)^2\right) \qquad (1.8)$$

$$ENT = -\sum_{i=1}^{M}\sum_{j=1}^{N}\left(\log P_{d,\theta}(i,j) P_{d,\theta}(i,j)\right) \qquad (1.9)$$

$$CONT = \sum_{i=1}^{M} \sum_{j=1}^{N} \left((i-j)^2 P_{d,\theta}(i,j)\right) \quad (1.10)$$

$$IDM = \sum_{i=1}^{M} \sum_{j=1}^{N} \left(\frac{P_{d,\theta}(i,j)}{1+(i-j)^2}\right) \quad (1.11)$$

Où *M* et *N* sont les tailles horizontales et verticales de l'image.

$P_{d,\theta}$: Élément de la matrice de cooccurrence.

d: La distance entre les 2 pixels du motif.

θ: La direction.

La signification statistique des descripteurs cités ci-dessus peut être décrite comme suit [REG 14]:

- L'**énergie** exprime le caractère régulier de la texture. De manière générale, une énergie élevée est observée lorsque l'image est très régulière, c'est-à-dire lorsque les valeurs élevées de la GLCM sont concentrées à quelques endroits de la matrice. C'est le cas par exemple pour des images dont la distribution des niveaux de gris soit un aspect constant, soit un aspect périodique. Une image aléatoire ou fortement bruitée produit une GLCM distribuée de manière plus uniforme et présente une énergie faible.
- L'**entropie** est d'autant plus élevée que la diagonale de la GLCM est étalée, le cas extrême étant une GLCM uniforme. En ce sens, l'entropie est l'inverse de l'énergie et caractérise l'aspect irrégulier de l'image, d'où une corrélation forte entre ces deux attributs.
- Le **contraste** est plus élevé pour des GLCMs présentant des valeurs plus larges en dehors de la diagonale, autrement dit pour des images affichant des changements locaux d'intensité.
- L'**homogénéité (**le moment de différence inverse (IDM)**)** évolue à l'inverse du contraste et prend des valeurs élevées si les différences entre les paires de pixels analysées sont faibles. Celle-ci est donc plus sensible aux éléments diagonaux de la GLCM, contrairement au contraste qui dépend plus des éléments éloignés de la diagonale.

Une fois calculés et éventuellement normalisés, l'ensemble des descripteurs sont rassemblés dans un vecteur unique caractérisant chaque pixel ou chaque région de l'image.

Dans le cas d'une analyse pixel-à-pixel, les descripteurs sont souvent représentés sous la forme d'une image pour chaque descripteur calculé.

7.2.2.3. Analyse fréquentielle de la texture

La plupart des méthodes fréquentielles se base sur l'extraction d'attributs à partir de l'image transformée par filtrage. Nous allons présentés : les filtres de Gabor et la transformée en ondelettes.

a) Filtres de Gabor

En 1946, *Gabor* [GAB 46] a proposé un filtre 1D qui optimise la relation entre le temps et la fréquence. *Daugman* [DAU 85] a ensuite développé cette idée pour construire des filtres 2D optimisant la relation entre la localisation spatiale et la fréquence spatiale. Les filtres de Gabor sont très utilisés en indexation, pour la description de la texture. Ils sont notamment utilisés par la norme MPEG-7. Ces filtres sont généralement exploités dans l'espace de Fourier dans le but de caractériser des textures locales [HOU 10]. L'utilisation des filtres de Gabor consiste à calculer la transformée de Fourier sur une partie du signal sélectionnée à l'aide d'une fenêtre bien localisée en temps. Des translations successives de cette fenêtre permettent d'analyser localement le comportement temps-fréquence du signal.

Le filtre spatial (ou directionnel) de Gabor est composé du produit (équation 1.12) d'un filtre passe bas h_y (agissant sur la direction locale θ du filtre) et d'un filtre passe bande h_x (partie réelle d'un filtre de Gabor à une dimension).

$$h(\dot{x},\dot{y}) = \left\{exp\left(-\frac{1}{2}\frac{\dot{y}^2}{\sigma_y{}^2}\right)\right\} \times \left\{exp\left(-\frac{1}{2}\frac{\dot{x}^2}{\sigma_x{}^2}\right) \times \cos\left(2\pi f_0 \dot{x}\right)\right\} = h_x(\dot{x}) \times h_y(\dot{y}) \quad (1.12)$$

$$avec \begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix} = \begin{bmatrix}\cos(\theta) & \sin(\theta)\\ -\sin(\theta) & \cos(\theta)\end{bmatrix}\begin{pmatrix}x\\ y\end{pmatrix}$$

La Figure 1.8, illustre une représentation des bandes passantes fréquentielles (Δf) et directionnelles ($\Delta\Omega$) d'un filtre de Gabor (f_0 représente la fréquence centrale, a et b sont les rayons de l'ellipse centrée en f_0).

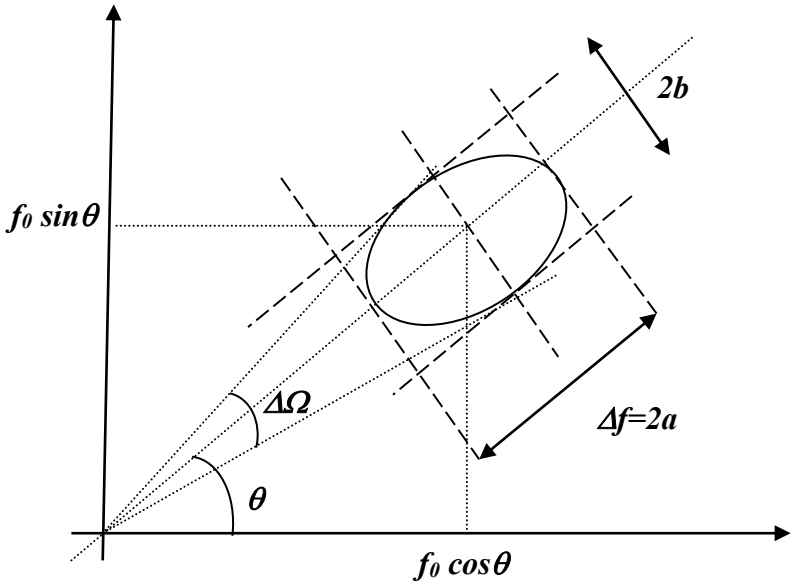


Figure 1.8 : Filtre de Gabor.

Le filtre directionnel de Gabor possède 4 degrés de liberté [MIB 11] :

- L'orientation θ du filtre : ce paramètre fait pivoter l'ondelette autour de son centre.
- La fréquence centrale f_0 que l'on cherche à extraire de l'image : le choix de la fréquence centrale f_0 du filtre est très important car la qualité de l'image finale dépend directement du bon choix de ce paramètre.
- Les écarts types σ_x et σ_y permettant de régler les bandes passantes fréquentielle et directionnelle.

b) Transformée en ondelettes

Malgré la capacité remarquable des filtres de Gabor à décomposer le spectre fréquentiel de l'image, les attributs texturaux extraits de ces filtres peuvent être corrélés en raison de la non-orthogonalité des filtres. Il peut dès lors s'avérer difficile de déterminer si une similarité observée entre échelles d'analyse est due aux propriétés de l'image ou à la redondance inhérente à la représentation. En outre, à chaque échelle d'application des filtres de Gabor, les paramètres définissant ces filtres doivent être modifiés. Ces contraintes sont levées par l'utilisation des ondelettes [MAL 89]. Celles-ci offrent en effet un cadre d'analyse multi-échelles uniforme (une seule paramétrisation pour toutes les échelles) et permettent de décomposer l'image en sous-bandes orthogonales et indépendantes limitant ainsi la redondance d'informations [REG 14].

La transformée en ondelettes continue d'une fonction $f \in L_2(\Re)$ est donnée par l'équation 1.13 :

$$\gamma(s,\tau) = \int f(t)\Psi^*_{s,\tau}(t)dt \qquad (1.13)$$

Où * dénomme le complexe-conjugué. Cette équation montre comment une fonction f est décomposée dans un ensemble de fonctions $\Psi_{s,\tau}$ appelées les ondelettes. Les variables s et τ sont les nouvelles dimensions, échelle et translation, de la transformée en ondelettes. La transformée en ondelettes inverse est donnée par l'équation 1.14.

$$f(t) = \int\int \gamma(s,\tau)\Psi^*_{s,\tau}(t)d\tau ds \qquad (1.14)$$

Les ondelettes (équation 1.15) sont générées à partir d'une ondelette Ψ appelée l'ondelette mère, en appliquant des opérations de changement d'échelle et des translations.

$$\Psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\Psi\left(\frac{t-\tau}{s}\right) \qquad (1.15)$$

s est le facteur d'échelle, τ est le facteur de translation et $\frac{1}{\sqrt{s}}$ est un facteur de normalisation à travers les différents échelles.

En pratique, l'utilisation de la décomposition en ondelettes sur une image discrète à deux dimensions revient à appliquer le produit de filtres passe-haut (H) et passe-bas (L) à une dimension. Dans l'analyse multi-résolutions définie par [MAL 89], une approche de décomposition appelée transformée en ondelettes discrète (DWT : Discrete Wavelet Transform) est proposée. Celle-ci consiste à décomposer l'image en quatre sous-bandes sous-échantillonnées d'un facteur 2 (ondelettes dyadiques) à chaque échelle de décomposition. Ces sous-bandes sont le résultat de combinaisons entre filtre passe-haut et filtre passe-bas : LL, LH, HL, HH. La sous-bande LL ou sous-bande d'approximation est une version moyennée de l'image d'origine alors que les sous-bandes HL, LH et HH ou sous-bandes de détails contiennent les hautes fréquences de l'image respectivement dans la direction de x (horizontale), dans la direction de y (verticale) ou dans les deux directions x et y (diagonale). Pour obtenir le niveau d'échelle de décomposition suivant, la sous-bande LL est à nouveau filtrée et sous-échantillonnée (*Figure 1.9*). Généralement, l'essentiel de l'information texturale étant lissée par l'application du filtre passe-bas, seules les sous-bandes de détails HL, LH et HH sont exploitées dans l'analyse texturale [REG 14].

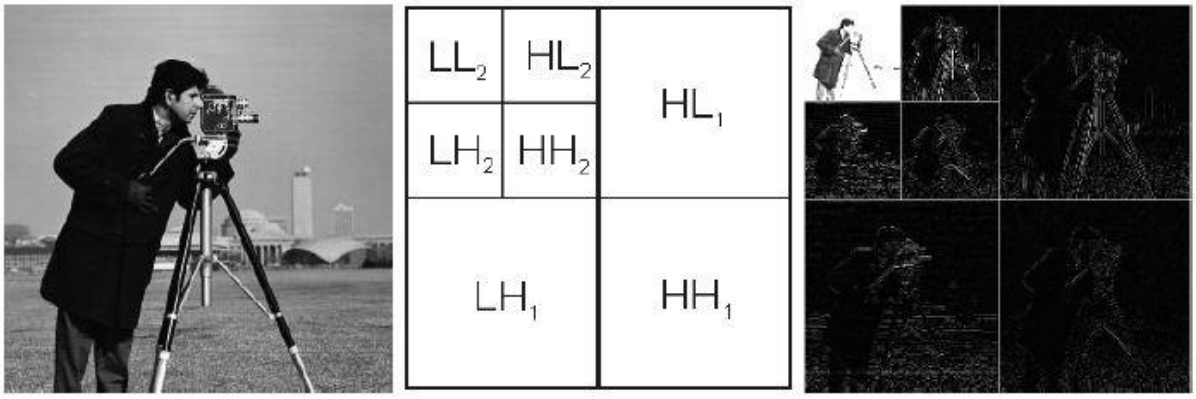


Figure 1.9 : Exemple de décomposition par ondelettes. Image d'origine (gauche), Schéma de décomposition de l'image par DWT avec 2 échelles de décomposition (centre), Résultat de la décomposition (droite).

7.2.3. Forme

Au même titre que les descripteurs/caractéristiques de texture, les attributs de forme sont complémentaires de la description couleur [FOU 02]. La forme est un attribut visuel essentiel et est l'une des caractéristiques/descripteurs de base pour démontrer le contenu d'une image. La précision des descripteurs/caractéristiques de forme est largement basée sur le schéma de segmentation appliqué pour diviser une image en objets significatifs. Quoique, la description des formulaires est une tâche difficile. Une bonne caractéristique/descripteur de forme doit être invariante à la rotation à la translation, à et l'échelle [SUD 18].

Les techniques de modélisation sont regroupées en deux classes [BEN 17, FOU 02, SUD 18]:

1) *l'approche contour* (frontière) décrit une région au moyen des pixels situé sur son contour. Les techniques de description de forme représentatives sont le descripteur de Fourier, les codes de chaîne, le modèle par éléments finis et les approximations polygonales.

2) *l'approche région* considère une région par rapport aux caractéristiques des pixels que cette région contient. Un descripteur basé sur une région spécifie la structure de l'objet à l'intérieur de la frontière et la forme est documentée comme un groupe de primitives (par exemple, quadriques, cercles, rectangles,…). Cette approche fait classiquement référence aux moments invariants [HU 62]. Ces attributs sont robustes aux transformations géométriques comme la translation, la rotation et le changement d'échelle.

7.2.3.1. Les descripteurs géométriques de région

Les descripteurs géométriques de forme permettent de distinguer les différents types de forme que peuvent prendre les objets d'une scène. Ils nécessitent une segmentation en région préalable de l'image. Ils sont ensuite calculés sur les différentes régions de l'image [COI 05, LAFO 06]. La surface relative (ou normalisée) d'une région $\Re_k$ de l'image *I* est le nombre de pixels contenus dans cette région par rapport au nombre total de pixels de l'image [FOU 02] :

$$S_k = \frac{card(\mathcal{R}_k)}{hauteur\ (I) * largeur(I)} \qquad (1.16)$$

Le centre de masse des pixels de la région est définie par :

$$P = (P_i, P_j) = \left(\frac{\sum_{i=\mathcal{R}_k} i/card(\mathcal{R}_k)}{largeur\ (I)}, \frac{\sum_{j=\mathcal{R}_k} j/card(\mathcal{R}_k)}{hauteur\ (I)} \right) \qquad (1.17)$$

La longueur du contour de la région est le nombre de pixels en bordure de la région:

$$l_k = card(contour(\mathcal{R}_k)) \qquad (1.18)$$

La compacité traduit le regroupement des pixels de la région en zones homogènes et non trouées:

$$C_k = \frac{l_k^2}{S_k} \qquad (1.19)$$

Ces descripteurs très simples permettent d'obtenir des informations sur la géométrie des régions de l'image. Il existe d'autres descripteurs de forme, basés sur des statistiques sur les pixels des régions de l'image.

7.2.3.2. Les moments géométriques

Les moments géométriques permettent de décrire une forme à l'aide de propriétés statistiques. Ils sont simples à manipuler mais leur temps de calcul est très long.

On a:

I (M, N): une image.

Le moment d'ordre (p, q) est calculé par la formule suivante.

$$M_{pq} = \sum_{i=1}^{M}\sum_{i=1}^{N} i^p j^q I\,(i,j) \qquad (1.20)$$

La superficie est calculée par la formule suivante.

$$M_{00} = \sum_{i=1}^{M}\sum_{i=1}^{N} I\,(i,j) \qquad (1.21)$$

*Mass Center e*st calculé par les formules suivantes.

$$M_{01} = \sum_{i=1}^{M}\sum_{i=1}^{N} j^q I\,(i,j) \qquad (1.22)$$

$$M_{10} = \sum_{i=1}^{M}\sum_{i=1}^{N} i^p I\,(i,j) \qquad (1.23)$$

$\bar{i} = \frac{M_{10}}{M_{00}}$ and $\bar{j} = \frac{M_{01}}{M_{00}}$

Les moments centrés pour être invariants en translation sont calculés par la formule suivante.

$$\mu_{pq} = \sum_{i=1}^{M}\sum_{j=1}^{N}(i-\bar{i})^p (j-\bar{j})^q I(i,j) \qquad (1.24)$$

Les moments normalisés à mettre à l'échelle invariants sont calculés par la formule suivante.

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{1+\frac{p+q}{2}}} \qquad (1.25)$$

a) Les moments invariants de Hu

À partir des moments géométriques, Hu [HU 62] a défini sept moments invariants aux translations, rotations et changement d'échelle, appelés moments de Hu.

Les moments invariants de Hu [HU 62] décrivent la forme au moyen de propriétés statistiques. Ils sont simples à manipuler et résistent aux transformations géométriques (la rotation, la translation et la mise à l'échelle).Ces 7 moments invariants sont:

• Les invariants du second ordre sont définis par les équations (1.26) et (1.27):

$$\Phi_1 = \eta_{20} + \eta_{02} \qquad (1.26)$$

$$\Phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}{}^2 \qquad (1.27)$$

• Les invariants du troisième ordre sont définis par l'équation (1.28), (1.29), (1.30), (1.31) et (1.32):

$$\Phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - 3\eta_{03})^2 \qquad (1.28)$$

$$\Phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \qquad (1.29)$$

$$\Phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2]\,(3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\,[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \qquad (1.30)$$

$$\Phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} - \eta_{03}) \qquad (1.31)$$

$$\Phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{12} - \eta_{30})\,(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \qquad (1.32)$$

b) Transformée de Hough

Cette transformation a été proposée par P.V.C, Hough dans un brevet déposé en 1960 [HOUG 18] afin de détecter des alignements à l'aide d'un oscilloscope et de deux caméras vidéo. Inaperçue pendant plusieurs années, elle a reçu quelque publicité après les travaux de Rosenfeld [ROS 69] et Duda et Hart [HART 72] au début des années 70, et fait l'objet depuis cette date d'une attention soutenue de la communauté scientifique. Depuis les années 80 elle a quitté les laboratoires de recherche et trouvé des champs d'applications dans de nombreux domaines industriels [MAIT 85].

On regroupe usuellement sous le nom de ‘‘TH’’ des transformations qui permettent de détecter dans des images la présence de courbes paramétriques appartenant à une famille

connue, à partir d'un ensemble de points sélectionnés, appelés points caractéristiques. La TH utilise essentiellement l'information spatiale des points caractéristiques (leur position dans l'image), mais, parfois, tient compte également de l'information contenue dans le signal d'image lui-même (la valeur de la luminance en un point donné). Nous supposerons que ce signal est une fonction scalaire, mais rien ne s'oppose à ce qu'il soit vectoriel; c'est le cas des images en couleur, ou des images multi-spectrales. Enfin, bien que la plupart des images soient bidimensionnelles, nous pourrons appliquer la TH à des images à trois dimensions (en imagerie médicale par exemple, ou sur des séquences d'images animées), voir à quatre dimensions. Nous désignerons par la dimension l'espace de définition de l'image [HOU 10, MAIT 85].

Définition

Soit $\Re^n$ l'espace image, et ξ un ensemble de N points sélectionnés par un prétraitement :

$$\xi = \{M_i, i = 1, \dots, N\} \epsilon \Re^n \qquad (1.33)$$

Un point M de $\Re^n$ est repéré par ses coordonnées x.

Soit $\Omega \subset \mathcal{R}^p$ un espace de paramètres et F une famille de courbes dans $\Re^n$ paramétrée par a :

$$F = \{\{x: f(x,a) = 0, x \in \mathcal{R}^n\}, a \in \Omega\} \qquad (1.34)$$

On appelle transformation de Hough associée à la famille F une transformation qui fait correspondre à l'ensemble ζ une fonction g définie sur Ω.

Il existe donc de nombreuses transformations de Hough, les deux principales sont les suivantes:

- **Transformation de m à 1**

Soit m le nombre minimal de points de $\Re^n$ définissant une courbe de F. Soit $\xi^{(m)}$ l'ensemble de tous les $m - uples\ issus\ de\ \xi$:

$$\xi^{(m)} = \left\{M^{(m)}_{i1} = \{M_{i2}, \dots\dots\dots, M_{im}: M_{ik} \in \xi\}\right\} \qquad (1.35)$$

$$avec : card(\xi^{(m)}) = C_N^m \qquad (1.36)$$

A tout $m - uples\ M_i^{(m)}\ de\ \xi^{(m)}$ est associée une courbe de *F* de paramètre a_i. Soit *C(a)* la fonction caractéristique de $\Re^p$. La transformation de Hough de *m à 1* est définie par :

$$g(a) = \sum_{M_i^{(m)} \in \xi^{(m)}} c(a - a_i) \qquad (1.37)$$

- **Transformation de 1 à m :**

Par tout point M_i de $\Re^p$ passent *m* courbes de *F,* soit l'ensemble des valeurs de *a* telles que :

$$f(x_i, a) = 0 : A_i = \{a_k = f(x_i, a_k) = 0\}$$

La Transformation de Hough de 1 à m est définie par :

$$g(a) = \sum_{M_i \in \xi} \sum_{a_k \in A_i} c(a - a_k) \qquad (1.38)$$

La TH (Transformation de Hough) de *1 à m* conduit, en pratique, à des calculs moins nombreux que la TH de *m à 1*, car elle évite une recherche combinatoire parmi les points. D'autre part, elle se prête bien à des implantations rapides, par sa structure parallélisable. Pour ces raisons c'est actuellement la TH la plus utilisée, et on lui réserve souvent l'usage du nom générique TH [MAIT 85].

7.2.4. Les descripteurs des points d'intérêts

Les points anguleux (ou points d'intérêt, points saillants,...) sont des points « qui contiennent beaucoup d'information » relativement à l'image. Ce sont des points aux voisinages desquels l'image *varie significativement dans plusieurs directions* [HOU 10]. Les points d'intérêt sont des régions de l'image riches en termes de contenu de l'information locale et stables sous des transformations affines et des variations d'illumination. Les points d'intérêts sont aussi plus stables que les régions ou les contours de l'image et leur extraction est plus simple. Les algorithmes de détection de points d'intérêt se focalisent en général sur des points particuliers des contours, sélectionnés selon un critère précis. Ainsi, les coins (corners) sont les points de l'image [BEN 17].

En général, une région d'intérêt possède les caractéristiques suivantes [BEN 17]:

- ✓ Elle a une définition mathématique formelle.
- ✓ Elle a une position précise dans l'image.
- ✓ Elle est riche en informations visuelles locales.
- ✓ Elle est stable face à des variations locales et globales de l'image, i.e., elle conserve les mêmes informations visuelles en cas de variation.

➢ Extraction des caractéristiques des points d'intérêt par SIFT

Une approche représentative pour caractériser les informations des points d'intérêt est la méthode SIFT (*Scale-Invariant Feature Transform*) proposée et développée par David Lowe [LOW 04]. SIFT est un algorithme qui permet de détecter des points d'intérêt et d'extraire des caractéristiques distinctives de ces points pour la reconnaissance d'objet. Les caractéristiques de SIFT sont invariantes à l'échelle et à la rotation, ce qui joue un rôle très important pour la reconnaissance des images a angle de prise de vue très diverses [HOU 10]. En un point particulier de l'image, ce descripteur se présente sous forme d'un histogramme grossier des orientations des gradients contenus dans son voisinage. Ce descripteur se présente sous forme d'un vecteur de dimension 128. Il est obtenu en calculant les modules et les orientations des gradients au voisinage d'un point caractéristique. En sélectionnant 16 régions autour du point d'intérêt de taille chacune 4x4 pixels, en calculant les 16 histogrammes d'orientations des gradients. Chaque histogramme est de taille 8 bins qui représentent les 8 orientations principales entre 0 et 360 degrés [BEN 17].

La détection et l'extraction des caractéristiques sur les points d'intérêt se déroulent en quatre étapes [HOU 10] :

1. Détection d'extrema d'espace-échelle ("scale-space") :

L'image est convoluée avec un noyau gaussien. L'espace-échelle d'une image est donc défini par la fonction :

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \qquad (1.39)$$

Où *I(x, y)* est l'image originale et

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \qquad (1.40)$$

A une normalisation près, cela revient à résoudre $\frac{\partial y}{\partial x} = \Delta I$, où ΔI représente le Laplacien de *I*.

2. *Localisation des points d'intérêt :*

La présélection des points d'intérêt et de leur échelle est faite en détectant les extrema locaux des différences de gaussiennes :

$$D(x, y, k\sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \qquad (1.41)$$

$$D(x, y, k\sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \qquad (1.42)$$

Notons que $D(x, y, \sigma) \approx (k-1)\Delta I \ \ lorsque\ k \to 1$

Les extrema sont recherchés dans de petits voisinages en position et en échelle (typiquement 3 x 3 x 3).

Une étape d'interpolation a pour but d'améliorer la localisation des points d'intérêt en espace et en échelle. Puis une analyse des rapports des valeurs propres de la matrice hessienne ''2 x 2'' permet d'éliminer les points d'intérêt situés dans des zones insuffisamment contrastées ou sur des bords présentant une courbure trop faible.

3. *Choix de l'orientation des descripteurs :*

Cette étape consiste à assigner à chaque point une orientation. Cette orientation correspond à l'orientation majoritaire des gradients spatiaux d'intensité calculés dans un voisinage du point d'intérêt à l'échelle préalablement déterminée. Un point d'intérêt peut se voir associer plusieurs orientations. Cela entraîne par la suite une redondance des descripteurs.

4. *Calcul des descripteurs :*

Finalement, pour une position, une échelle et une orientation données, chaque point d'intérêt se voit associer un descripteur. Pour chaque image, la norme du gradient spatial *m(x,y)* et l'orientation du gradient spatial *θ(x,y)* correspondants à cette échelle sont calculées :

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \qquad (1.43)$$

$$\theta(x, y) = \tan^{-1}\left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}\right) \qquad (1.44)$$

Le descripteur est constitué d'histogrammes d'orientation du gradient spatial d'intensité pondérés par la norme du gradient spatial.

En effet, le voisinage du point d'intérêt dont la taille dépend de l'échelle subit un découpage 4 x 4 en blocs. Pour chaque bloc, un histogramme à 8 niveaux de quantification résume les orientations du gradient spatial d'intensité à l'intérieur du bloc. Le descripteur SIFT est donc un vecteur à 4 x 4 x 8 = 128 coordonnées.

D'autres descripteurs ont été proposés dans la littérature [BEN 17]:

- *Shape context* [BEL 02], le principe de cet algorithme est d'extraire d'une image les point en décrivant les contours, et d'obtenir pour chacun de ces points le contexte de forme en déterminant la distribution relative des points les plus proches au moyen d'un histogramme de distribution de coordonnées log-polaires.
- *PCA-SIFT* [KE 04], est un vecteur de gradients d'image dans les directions x et y calculé à l'intérieur de la région de support. La région du gradient est échantillonnée en 39 × 39 positions, générant un vecteur de dimension 3042. Cette dimension est réduite à 20 par la méthode d'analyse en composantes principales.
- *Gradient Location and Orientation Histogram* (GLOH) [MIK 05], est une extension du descripteur SIFT dont la robustesse et le caractère distinctif ont été améliorés.
- *Le descripteur SURF* (Speeded Up Robust Features) présenté pour la première fois par Herber Bay en 2006 [BAY 06]. SURF est un algorithme de détection de points caractéristiques d'une image et un descripteur associé. Il est également plus performant en terme de rapidité que le détecteur SIFT; les résultats fournis étant comparables au regard de la qualité des points caractéristiques détectés. Son avantage réside également dans son invariance aux rotations et aux changements d'échelles. Tel SIFT, SURF permet la détection à la fois de points d'intérêt et de leurs descripteurs associés.

7.3. La similarité

La notion de similarité entre deux images est une notion difficile à définir. Il faut en effet préciser sur quel(s) critère(s) cette notion de similarité se base. La *figure 1.10* présente un exemple où on demande à l'utilisateur de classer les quatre images proposées en familles d'images similaires.

Exemple :

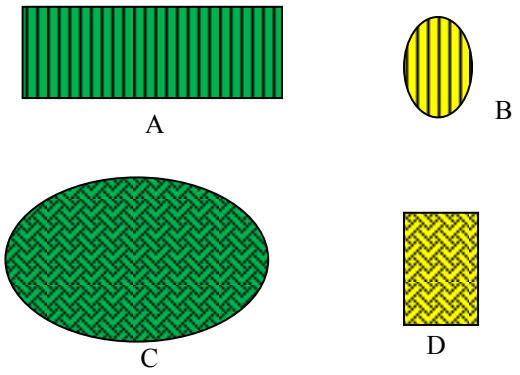


Figure 1.10 : Exemple de base à classer en familles d'images similaires.

Il y a plusieurs réponses acceptables à cette question selon le **critère de similarité** retenu. Si on décide de retenir la couleur comme critère, les deux familles sont (A,C) et (B,D). Si c'est la texture que nous retenons, on aura (A,B) et (C,D). Si c'est la forme que nous choisissons comme critère, les deux familles deviennent (A,D) et (B,C). Enfin, si nous choisissons la taille, le regroupement devient (A,C) et (B,D) comme dans le premier cas. Il est aussi possible de définir une composition de ces critères avec un résultat qui dépend de leur importance relative.

Cet exemple est volontairement caricatural mais il indique qu'il faut tenir compte du critère de similarité lors de la recherche. Autrement dit, il faut créer un vecteur descripteur qui contient les informations selon un ou plusieurs critère(s) choisi(s) en fonction des besoins de l'utilisateur.

La similarité de deux images est un problème mal posé dans lequel des contraintes additionnelles sont nécessaires pour la régularisation. Il y a des choix à faire a priori pour résoudre ce problème. Il faut donc définir un critère de similarité avant de comparer deux images.

Dans la définition de la similarité et dans l'extraction des attributs des images, l'expert du domaine joue un rôle très important. Il choisit les caractéristiques des objets et propose un modèle de comparaison des images. Il apporte son expérience du domaine comme connaissance a priori de la similarité entre les images.

7.3.1. Mesure de similitude entre descripteurs

Une fois la base de données est indexée en fonction des caractéristiques telles que la couleur, la texture et la forme, les images les plus pertinentes sont recherchées. Dans la

plupart des CBIR, la distance entre les caractéristiques de deux images est utilisée pour mesurer leur similitude. Plus la valeur de la distance est petite, plus la différence entre les images est petite, et donc c'est la plus similaire des images.

Généralement, la requête est exprimée sous la forme d'un ou de plusieurs vecteurs de grande dimension, on cherche alors les plus proches voisins de la requête, et cela durant la phase en ligne, où le système calcule la correspondance entre le descripteur de l'image-requête et chaque descripteur des images de la base.

La mesure de similarité vérifie généralement les propriétés suivantes [HOU 10]:

- La perception : une faible distance dans l'espace caractéristique indique deux images semblables.

- Le calcul : la mesure de distance se calcule rapidement pour une faible latence.

- La stabilité : le calcul de distance ne doit pas être affecté par une modification de la taille de la base.

- La robustesse : la mesure devra être robuste aux changements des conditions d'acquisition d'image.

De ce fait le calcul, établi par une mesure de similarité a pour résultat une liste ordonnée des images de la base. La première image de cette liste est celle qui est considérée par le système comme la plus pertinente, c'est-à-dire celle qui répond le mieux à la requête image. La complexité de calcul d'une distance doit être raisonnable parce que dans un système CBIR, cette tache s'exécute en temps réel. D'autres paramètres entrent enjeu tel la dimension de l'espace caractéristique, la taille de la base, ...

Dans la littérature, Il existe un grand nombre de mesures de similarité (dissimilarité) entre les images de la base et l'image-requête. Certaines mesures sont spécifiques aux histogrammes ou aux distributions. Certaines sont des distances (des mesures qui respectent les axiomes des espaces métriques : symétrie, réflexivité, non-négativité et l'inégalité triangulaire).

Considérons deux vecteurs caractéristiques $X = (x_1, x_2, x_3, \ldots, x_n)$ et $Y = (y_1, y_2, y_3, \ldots, y_n)$.

- **Définition des espaces métriques**

Maurice Fréchet (1878-1973) définit un espace métrique E comme un ensemble non vide doté d'une application d, appelée distance, de $E \times E$ dans R^+ vérifiant les axiomes suivants :

$\forall\, x, y, z \in E$

1. $d(x,y) = 0 \Leftrightarrow x = y$ ($Identité$)
2. $d(x,y) = d(y,x)$ ($Symétrie$)
3. $d(x,y) + d(y,z) \geq d(x,z)$ ($Inégalité\ triangulaire$)

L'axiome de l'inégalité triangulaire est calqué sur la structure métrique du plan muni de la distance euclidienne usuelle, définie dans un repère orthogonal utilisant la même unité sur chaque axe par :

$$d(A,B) = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2} \qquad (1.45)$$

Les métriques de Minkowski (ou normes *Lp*) sont les distances géométriques les plus courantes.

7.3.1.1. Distances de Minkowski

L'approche la plus simple pour mesurer la similitude entre deux images correspond aux distances de Minkowski. Cette distance est calculée entre les vecteurs descripteurs/caractéristiques, elle est basée sur la norme L_r qui est définie par l'équation (1.46):

$$L_r(V_1, V_2) = \left[\sum_{i=0}^{n} |V_1(i) - V_2(i)|^r\right]^{1/r} \qquad (1.46)$$

Où V_1, V_2: vecteurs descripteurs / caractéristiques.

Lorsque r = 1 ou r = 2 ou r =∞ , cela s'appelle respectivement la distance de Manhattan (the city block distance), la distance euclidienne et la distance de Tchebychev (la distance du maximum), comme indiqué ci-dessous dans l'équation (1.47), équation (1.48) et l'équation (1.49) en conséquence:

$$L_1(V_1, V_2) = \sum_{i=0}^{n} |V_1(i) - V_2(i)| \qquad (1.47)$$

La distance de Manhattan est plus appropriée pour mesurer la similarité entre les données multi-variées.

$$L_2(V_1, V_2) = \left[\sum_{i=0}^{n} |V_1(i) - V_2(i)|^2\right]^{1/2} \quad (1.48)$$

La distance euclidienne est couramment utilisée dans des espaces à deux ou 3 dimensions, elle donne notamment de bons résultats si l'ensemble des données présentes des classes compactes et isolées.

$$L_\infty(V_1, V_2) = \left[\sum_{i=0}^{n} |V_1(i) - V_2(i)|^\infty\right]^{1/\infty} \quad (1.49)$$

Cette distance est adaptée aux données de grandes dimensions.

7.3.1.2. Distances entre histogrammes

Les techniques géométriques, telles que la distance euclidienne, peuvent être appliquées sur des histogrammes. Il existe cependant des mesures spécifiques aux histogrammes. L'*intersection d'histogrammes* a été proposée par Swain et Ballard [SWA 91], Cette mesure est l'une des premières distances utilisée dans la recherche d'image par le contenu, mesurant la partie commune entre deux histogrammes. Etant donné deux histogrammes h_1 et h_2 :

$$D_{Intersec} = \frac{\sum_{i=1}^{n} min(h_1(i), h_2(i))}{\sum_{i=1}^{n} h_2(i)} \quad (1.50)$$

Deux images présentant une intersection normalisée d'histogrammes proche de 1 sont considérées comme similaires. Cette mesure n'est pas une distance, car elle ne respecte pas l'axiome de symétrie. Il existe des travaux pour l'amélioration de l'intersection d'histogrammes: Smith [SMI 97] propose une version symétrique. Stricker [STR 95] propose le calcul de distance entre histogrammes cumulés. Cette approche est plus robuste que la distance sur les histogrammes classiques [SWA 91], car elle est moins sensible aux

changements (par exemple, aux changements de couleurs), mais elle suppose un ordre sur les composantes des histogrammes, ce qui n'est pas le cas des histogrammes de couleurs.

7.3.1.3. Distance quadratique

L'inconvénient des mesures telles que la distance de Minkowski ou l'intersection d'histogrammes est qu'elles traitent les éléments du vecteur de caractéristique d'une manière équitable (elles comparent les composantes des vecteurs une par une, sans prendre en compte les autres composantes). Pour remédier à ce problème, on peut utiliser la *distance quadratique* [HAF 95]. Cette distance favorise les éléments les plus ressemblants. Elle est définie ainsi :

$$D_Q = \sqrt{(f_1 - f_2)^T A (f_1 - f_2)} \qquad (1.51)$$

Où

$A=[a_{ij}]$ est la matrice de similarité. Elle permet de pondérer le poids des composantes voisines en fonction de leur distance à la composante considérée. Cependant, cette mesure a une complexité quadratique. Un cas particulier de distances quadratiques est la *distance de Mahalanobis* où A correspond à la matrice de covariance des données d'apprentissage. Cette distance nécessite donc la connaissance d'un ensemble de données d'apprentissage. Lorsque A est la matrice identitaire alors D_Q est équivalente à la distance euclidienne.

a_{ij} Représente la distance entre deux éléments des vecteurs f_1 et f_2. Hafner et al [HAF 95] propose la formule suivante pour construire la matrice A.

$$a_{ij} = 1 - \frac{d_{ij}}{\max(d_{ij})} \qquad (1.52)$$

Les propriétés de cette distance la rendraient proche de la perception humaine de la couleur, ce qui en fait une métrique attractive pour les systèmes de recherche d'images couleur par le contenu.

7.3.1.4. Earth Mover's Distance *(EMD)*

La mesure *Earth Mover's Distance (EMD)* [RUB 13] est basée sur la minimisation du coût nécessaire pour transformer une distribution en une autre distribution. Elle peut être appliquée pour calculer la similarité entre deux distributions ou entre deux ensembles de distributions. Elle porte ce nom car pour la comprendre on peut voir l'une des distributions comme une collection de planètes associées à leur masse et l'autre comme une collection de

trous dans le même espace. C'est l'une des seules distances qui permet de travailler sur des histogrammes qui n'ont pas forcément le même nombre de *bins*.

Nous donnons sa définition dans le cas où l'on souhaite calculer la distance entre deux ensembles de distributions. Soit $x = \{(\vec{x}_1, \omega_{x,1}), (\vec{x}_2, \omega_{x,2}), \dots, (\vec{x}_n, \omega_{x,n})\}$ et $y = \{(\vec{y}_1, \omega_{y,1}), (\vec{y}_2, \omega_{y,2}), \dots, (\vec{y}_m, \omega_{y,m})\}$ deux ensembles de distributions où chaque $k^{ième}$ distributions $\vec{z}_k$ est associé à un poids $\omega_{z,k}$, alors la distance *EMD* est définie par :

$$EMD(x,y) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m}\delta(\vec{x}_i,\vec{y}_j)W_{i,j}}{\sum_{i=1}^{n}\sum_{j=1}^{m}W_{i,j}} \qquad (1.53)$$

où $\delta(\vec{x}_i,\vec{y}_j)$est la distance (par exemple euclidienne) entre les distributions $\vec{x}_i$ et $\vec{y}_j$, et $W_{i,j}$ est le coût (appelé «flux») nécessaire pour se déplacer de la composante $\vec{x}_i$ à la composante $\vec{y}_j$. La matrice W est le résultat de la minimisation de la fonction de coût $C(X,Y,W) = \sum_{i=1}^{n}\sum_{j=1}^{m}\delta(\vec{x}_i,\vec{y}_j)W_{i,j}$ avec les contraintes suivantes :

$$\forall(i,j)W_{i,j} \geq 0 \ , \ \forall i \ \sum_{i=1}^{n}W_{i,j} \leq \omega_{x,i} \ , \ \forall j \sum_{j=1}^{n}W_{i,j} \leq \omega_{y,j}$$

et $$\sum_{i=1}^{n}\sum_{j=1}^{m}W_{i,j} = \min\left(\sum_{i=1}^{n}W_{i,j}, \sum_{j=1}^{m}W_{i,j}\right)$$

EMD peut travailler sur des histogrammes et n'a donc pas besoin qu'une quantification des vecteurs soit effectuée. Cette mesure est intéressante pour calculer la distance entre deux images segmentées où chaque blob serait représenté par une distribution visuelle. Cependant, *EMD* est très coûteuse au niveau du temps de calcul.

7.3.1.5. Distance de Mahalanobis

Cette distance prend en compte la corrélation entre les distributions des classes [MAHA 30]. Elle est ainsi définie par :

$$D_M = \sqrt{(f_1 - f_2)^T\, C^{-1}\, (f_1 - f_2)} \qquad (1.54)$$

Où *C* est la matrice de covariance. Dans les cas où les dimensions des caractéristiques sont indépendantes, *C* ne comporte que des variances et la distance de *Mahalanobis* se simplifie sous la forme :

$$D_M = \frac{\sum(f_1(i) - f_2(i))^2}{c_i} \qquad (1.55)$$

Si C est la matrice identité, D_M est la distance euclidienne.

7.3.1.6. Similarité multi-vectorielle

Lorsque l'image n'est plus indexée par un seul vecteur mais par un ensemble de vecteurs se rapportant aux différents espaces de caractéristiques (couleur, texture, forme, …), se pose le problème de la fusion d'informations issues de modèles distincts.

La première possibilité consiste à concaténer les différentes signatures et à utiliser une des mesures classiques de comparaison entre vecteurs. Les sous vecteurs sont préalablement centrés et réduits afin de se ramener à une échelle de valeurs commune aux différents attributs. Il est également possible d'adapter la fonction de similarité à la différence d'échelle, comme le permet la distance de Mahalanobis [MAHA 30], par une analyse des signatures de la base d'images.

Selon la dénomination de Rui et Huang [RUI 96], le modèle de similarité qui résulte de ce traitement par concaténation est appelé « modèle à plat » (Flat model).

La seconde possibilité consiste à traiter chaque espace d'attributs indépendamment. La similarité globale est calculée en fusionnant les scores de similarités relatifs à chaque espace de caractéristiques. On parle alors de « modèle hiérarchique » (hierarchical model) [RUI 96].

La fusion des similarités est classiquement réalisée par une simple combinaison linéaire (somme pondérée). Les coefficients de la somme règlent le poids relatif de chaque attribut dans la similarité globale. Ils peuvent être réglés manuellement ou bien encore automatiquement par des expériences psycho-visuelles ou par interaction avec l'utilisateur.

Lorsque l'image est indexée par un ensemble d'invariants calculés autour de points d'intérêt, il s'agit encore du cas multi-vectoriel. La technique la plus simple pour effectuer le calcul de similarité requête-cible consiste à associer un espace de vote à chaque image de la base, espace dans lequel est accumulé le nombre d'invariants similaires à la requête.

7.3.1.7. Distances entre distributions

La similarité entre distributions consiste à déterminer si deux distributions peuvent être issues de la même distribution de probabilités.

Notons que les mesures suivantes (test du $X2$, divergence de Kullback-Leibler et Jensen Différence Divergence) doivent être qualifiées de mesures de similarités plutôt que de distance car elles n'en vérifient pas toutes les conditions.

a. La distance de X^2

Le *test statistique du* $\boldsymbol{X^2}$ teste l'hypothèse que les échantillons observés x_i sont tirés de la population représentée par les y_j. On en déduit la distance suivante entre les distributions:

$$D(X,Y) = \frac{\sum_{i=1}^{n} (x_i - \hat{z}_i)^2}{\hat{z}_i} \qquad (1.56)$$

où $\hat{z}_i = (x_i - y_i)/2$

C'est une des mesures de similarité parmi les plus rapides, elle donne de bons résultats sur les grands ensembles de données.

b. La divergence de *Kullback-Leibler*

La divergence de ***Kullback-Leibler*** (noté divKL) exprime l'entropie relative de la distribution $\vec{x}$ par rapport à la distribution $\vec{y}$:

$$divKL(\vec{x},\vec{y}) = \sum_{i=1}^{n} x_i \ log_2 \frac{x_i}{y_i} \qquad (1.57)$$

Cette mesure n'est pas une distance, car elle n'est pas symétrique $divKL(\vec{x},\vec{y}) \neq divKL(\vec{y},\vec{x})$. Si l'on souhaite une distance, on peut utiliser la distance de *Kullback-Leibler* (notée *dKL*) [KUL 51] définie par :

$$dKL(\vec{x},\vec{y}) = divKL(\vec{x},\vec{y}) + divKL(\vec{y},\vec{x}) \qquad (1.58)$$

En statistique bayésienne, la divergence de *Kullback-Leibler* peut être utilisée pour mesurer la «distance» entre la distribution *a priori*, et la distribution *a posteriori*.

c. Divergence de Jeffrey (JD)

La divergence de *Jeffrey* est défini par :

$$D_{JD} = \sum_i f_1\ (i) \log \frac{f_1(i)}{\hat{f}_i} + \ f_2(i) \log \frac{f_2(i)}{\hat{f}_i} \qquad (1.59)$$

où $\hat{f}_i = (f_1(i) + \ f_2(i))/2$.

A la différence de la mesure *KL*. *JD* est symétrique et plus stable.

d. Distance de Kolmogorov Smirnov

Cette distance est appliquée aux distributions cumulées $f^c\ (i)$:

$$D_{KS} = \max_i\ |f_1^c\ (i) - f_2^c\ (i)| \qquad (1.60)$$

e. Distance de Cramer Von Mises

La distance de *Cramer Von Mises* s'applique également sur des distributions cumulées, elle est définie par :

$$D_{CVM} = \sum_i (\ f_1^c(i) - f_2^c(i)\ )^2 \qquad (1.61)$$

f. Distance de Bhattacharya

La distance de Bhattacharya exploite la séparabilité entre deux distributions gaussiennes représentées par leur covariance Σ:

$$D_B = \frac{1}{8}\ (\mu_1 - \mu_2\ )^T\ \Sigma^{-1}\ (\mu_1 - \mu_2\ ) + \frac{1}{2}\ln\frac{\det(\Sigma)}{\sqrt{\det(\Sigma_1)\ \det(\Sigma_2)}} \qquad (1.62)$$

où $\Sigma = 0.5 \times (\Sigma_1 + \Sigma_2\ )$ La séparabilité entre classes est estimée par la distance des moyennes et des matrices de covariance de chaque classe.

7.4. Évaluation des performances

Les mesures les plus courantes pour évaluer un système de recherche d'images par le contenu, sont le temps de réponse et l'espace utilisé. Plus le temps de réponse est court, plus l'espace utilisé est petit, et plus le système est considéré bon. En plus de ces deux mesures, nous nous intéressons à des mesures telles que la précision et le rappel. Ce sont les mesures les plus couramment utilisées pour évaluer les performances des algorithmes de récupération d'image. La précision de la récupération est définie comme la fraction des images récupérées qui sont effectivement pertinentes pour la requête par rapport au total retourné (voir l'équation (1.63)); Le rappel est la fraction des images pertinentes renvoyées par la requête par rapport au nombre total de données pertinentes images dans la base de données (voir équation (1.64)) [BEN 17, JOB 14].

Appelons A l'ensemble des images pertinentes (au sens d'une requête donnée) de la base et B l'ensemble des images retournées par le système. Nous définissons les critères suivants :

$$Précision = \frac{|A \cap B|}{|B|} = \frac{Nombre\ d'images\ pertinentes\ retrouvées}{Nombre\ total\ d'images\ retrouvées\ de\ la\ base\ de\ données} \quad (1.63)$$

$$Rappel = \frac{|A \cap B|}{|A|} = \frac{Nombre\ d'images\ pertinentes\ retrouvées}{Nombre\ total\ d'images\ pertinentes\ dans\ la\ base\ de\ données} \quad (1.64)$$

La précision et le rappel sont compris entre 0 et 1. Ces deux mesures permettent de rendre compte de la pertinence de la réponse du système à la requête de l'utilisateur. On trace la courbe précision en fonction du rappel qui donne la pertinence des réponses du système aux requêtes [LAN 08]. Le système est utilisé en recherche-par-similarité : une unique image exemple est proposée au système et ce dernier retourne les k images les plus proches de cette requête au sens de la fonction de similarité employée. La précision et le rappel sont estimés pour un nombre fixé d'images retournées par le système. En faisant varier ce paramètre, il devient possible de décliner nombre de mesures de qualité différentes [FOU 02].

Un bon système de récupération doit avoir des valeurs élevées de précision et de rappel.

8. Conclusion

Nous avons présenté dans ce chapitre une étude détaillée des systèmes de recherche d'images par contenu (CBIR), constitué de deux phases principales: la phase hors ligne (l'extraction des caractéristiques), et la phase en ligne (recherche).

Dans ce présent chapitre nous avons parcouru en premier lieu les différents types des bases de données, spécialement l'usage de chacune. En second lieu, nous avons présenté le principe général de l'indexation et de la recherche d'image. Et puis, nous avons exposé un état de l'art des systèmes CBIR. Ensuite, nous avons présenté les différentes distances employées pour mesurer la similarité entre les images.

Dans le chapitre suivant, nous détaillerons les notions du Cloud computing.

Chapitre 2
Le Cloud computing

1. Introduction

La technologie de l'internet se développe de manière exponentielle depuis sa création. De nos jour, une nouvelle "tendance" à fait son apparition dans le monde des IT (Technologies de l'information et de la communication), il s'agit du Cloud computing. Divers fournisseurs comme Google [4], Amazon [7], IBM [8] offrent des services à la demande sur une base de paiement à l'utilisation. Aujourd'hui, le besoin de services en ligne tels que l'espace de stockage, les logiciels, les plates-formes,… augmente rapidement. Le Cloud computing est un modèle pour permettre l'accès à la demande et sur le réseau à un pool partagé de ressources informatiques configurables (par exemple, réseaux, serveurs, stockage, applications et services). Les services Cloud peuvent être fournis en tant que SaaS (Software as a Service), PaaS (Platform as a Service) ou IaaS (Infrastructure as a Service) et ils peuvent être déployés en tant que Clouds privés, communautaires, publics ou hybrides [MEL 11].

Le Cloud computing offre plusieurs avantages comme la facilité de déploiement, aucune maintenance, une évolutivité rapide et efficace, une technique de virtualisation, et une indépendance géographique, le Cloud évolutif a révolutionné le régime informatique et commercial à notre ère. Mais cela pose plusieurs défis en matière de sécurité des données et des problèmes de confidentialité, et de nombreuses entreprises sensibles à la sécurité ont tendance à se détourner des services du Cloud pour cette même raison.

Dans ce chapitre, nous étalons les définitions du Cloud, son évolution, son architecture, ses différents modèles de services et de déploiement ainsi que ses caractéristiques. Nous citons quelques avantages et inconvénients de Cloud computing. Nous détaillons la relation entre le Cloud computing et le Big Data.

2. Le Cloud computing

Le concept de Cloud computing (CC) a reçu une attention considérable dans la littérature universitaire et technique au cours des dernières années. La littérature existante fait état de divers avantages que le Cloud computing (CC) peut apporter aux organisations, notamment la simplicité, la rentabilité, la réduction de la demande de main-d'œuvre qualifiée

et l'évolutivité. Cependant, la littérature avertit également les adoptants de faire attention aux risques potentiels associés à la mise en œuvre, la gestion et l'utilisation des services du Cloud computing [GAZ 13]**.** Le Cloud computing consiste à accéder à des données et des services sur un serveur distant à la demande des consommateurs. Cette idée n'a rien de nouveau et son principe de base reste le même depuis plusieurs décennies [ALZ 16]. Le terme « Cloud computing » est un terme anglais. De ce fait, on rencontre également des francisations : *informatique en nuage*, *informatique dématérialisée* et *infonuagique*….

➢ Qu'est ce que le Cloud Computing?

Le Cloud computing n'est pas toujours clairement défini. Le Cloud computing est un service par abonnement où vous pouvez obtenir un espace de stockage en réseau et des ressources informatiques. Une façon de penser au Cloud computing est de considérer votre expérience avec le courrier électronique. Votre client de messagerie, s'il s'agit de Yahoo !, Gmail, Hotmail, etc., s'occupe du logement de tout le matériel et des logiciels nécessaires pour prendre en charge votre compte de messagerie personnel. Lorsque vous souhaitez accéder à votre messagerie, vous ouvrez votre navigateur Web, accédez au client de messagerie et connectez-vous. La partie la plus importante de l'équation est l'accès à Internet. Votre e-mail n'est pas hébergé sur votre ordinateur physique; vous y accédez via une connexion Internet et vous pouvez y accéder n'importe où. Si vous êtes en voyage, au travail ou dans la rue pour prendre un café, vous pouvez consulter vos e-mails tant que vous avez accès à Internet. Votre e-mail est différent d'un logiciel installé sur votre ordinateur, tel qu'un programme de traitement de texte. Lorsque vous créez un document à l'aide d'un logiciel de traitement de texte, ce document reste sur l'appareil que vous avez utilisé pour le créer, sauf si vous le déplacez physiquement. Un client de messagerie est similaire au fonctionnement du Cloud computing. Sauf qu'au lieu d'accéder uniquement à votre messagerie, vous pouvez choisir les informations auxquelles vous avez accès dans le Cloud [HUT 11].

Définition 1:

Le National Institute of Standards and Technology (NIST) définit le Cloud computing comme suit : "*Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models* "[MEL 11].

« Le Cloud computing est un modèle permettant un accès réseau omniprésent et pratique à la demande à un pool partagé de ressources informatiques configurables (par exemple, réseaux, serveurs, stockage, applications et services) qui peuvent être rapidement provisionnées et publiées avec un effort de gestion minimal ou interaction des prestataires de services. Ce modèle Cloud est composé de cinq caractéristiques essentielles, de trois modèles de service et de quatre modèles de déploiement» [MEL 11].

Définition 2 :

D'autres Définitions tirée du Grand dictionnaire terminologique de l'Office québécois de la langue française,

« Modèle informatique qui, par l'entremise de serveurs distants interconnectés par Internet, permet un accès réseau, à la demande, à un bassin partagé de ressources informatiques configurables, externalisées et non localisables, qui sont proposées sous forme de services, évolutifs, adaptables dynamiquement et facturés à l'utilisation » [9].

Définition 3 :

« L'infonuagique, c'est en fait l'informatique vue comme un service et externalisée par l'intermédiaire d'Internet. Elle fait référence à l'utilisation de la mémoire et des capacités de calcul des ordinateurs et des serveurs répartis dans le monde entier et reliés par Internet. Les ressources informatiques mises en commun et rendues ainsi disponibles à distance peuvent être, entre autres, des logiciels, de l'espace de stockage et des serveurs » [9].

Il s'agit donc d'une délocalisation de l'infrastructure informatique.

3. Évolution du Cloud computing

Pour le Cloud computing, nous ne pouvons pas octroyer son paternité précisément**.** Certains l'attribuent au père fondateur d'internet : *Arpanet*, technologie sans laquelle l'informatique dans les nuages n'existerait pas. D'autres pensent que la discipline n'est apparue que plus tard, dans les années 2000, telle une solution à un problème rencontré par *Amazon* [7]: le surdimensionnement de son parc de serveurs hors période de fêtes. Pour rentabiliser ces machines, il décida de louer ses serveurs à d'autres entreprises à la demande. La première personne à employer l'expression de Cloud computing fut le professeur *Ramnath Chellappa* de l'université du Texas à Austin en 1997. *Salesforces*, en 1999, fut le premier, à transformer ce concept en business avec le logiciel de gestion de la relation client éponyme. *Amazon* lui emboîte le pas en 2002. C'est sans doute grâce à *IBM* que tout le monde en parle

aujourd'hui. En effet, en 2007, il décida de faire de ce concept l'une des lignes de force de sa stratégie. Vers 2009, de grandes entreprises comme Google, Microsoft, HP et Oracle avaient commencé à fournir des services de Cloud computing. Aujourd'hui, chaque personne utilise les services du Cloud computing dans sa vie quotidienne. Par exemple, Google Photos, Google Drive et iCloud, etc. À l'avenir, le Cloud computing deviendra le besoin fondamental des industries informatiques [10, SRI 18].

La *figure 2.1* schématise l'évolution du Cloud computing.

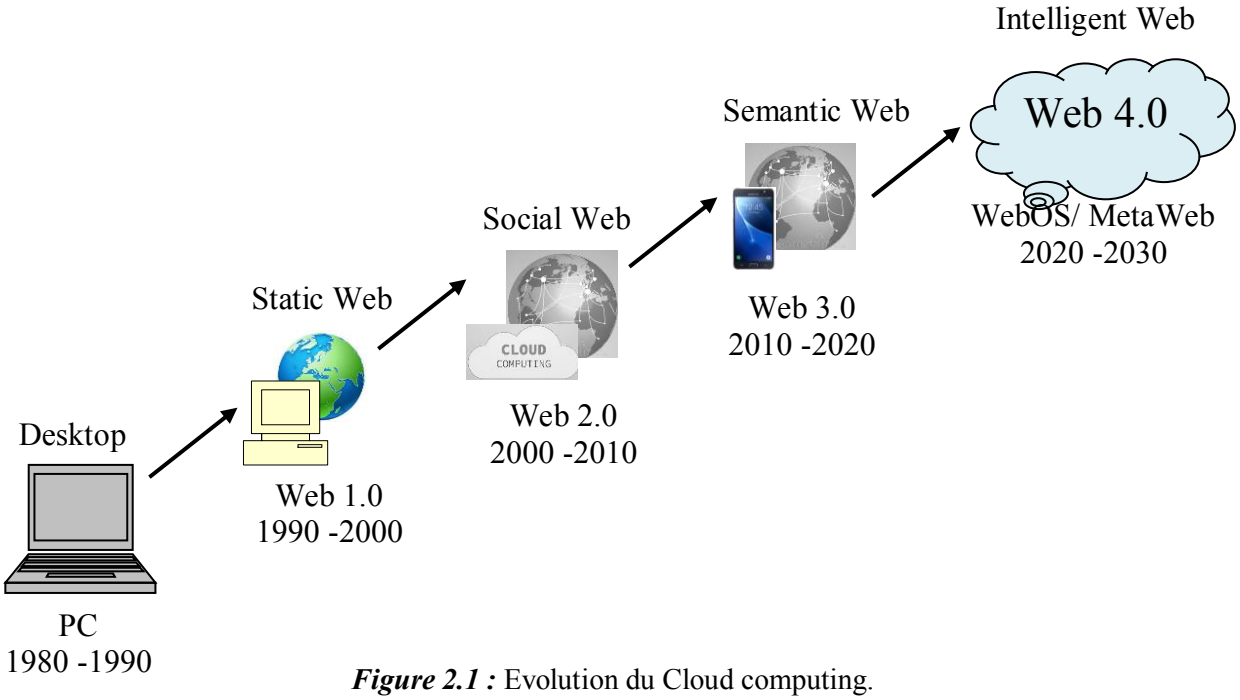


Figure 2.1 : Evolution du Cloud computing.

4. L'architecture du Cloud computing

Le Cloud computing est composé de cinq caractéristiques essentielles, de trois modèles de service et de quatre modèles de déploiement. La *figure 2.2* montre l'architecture du Cloud.

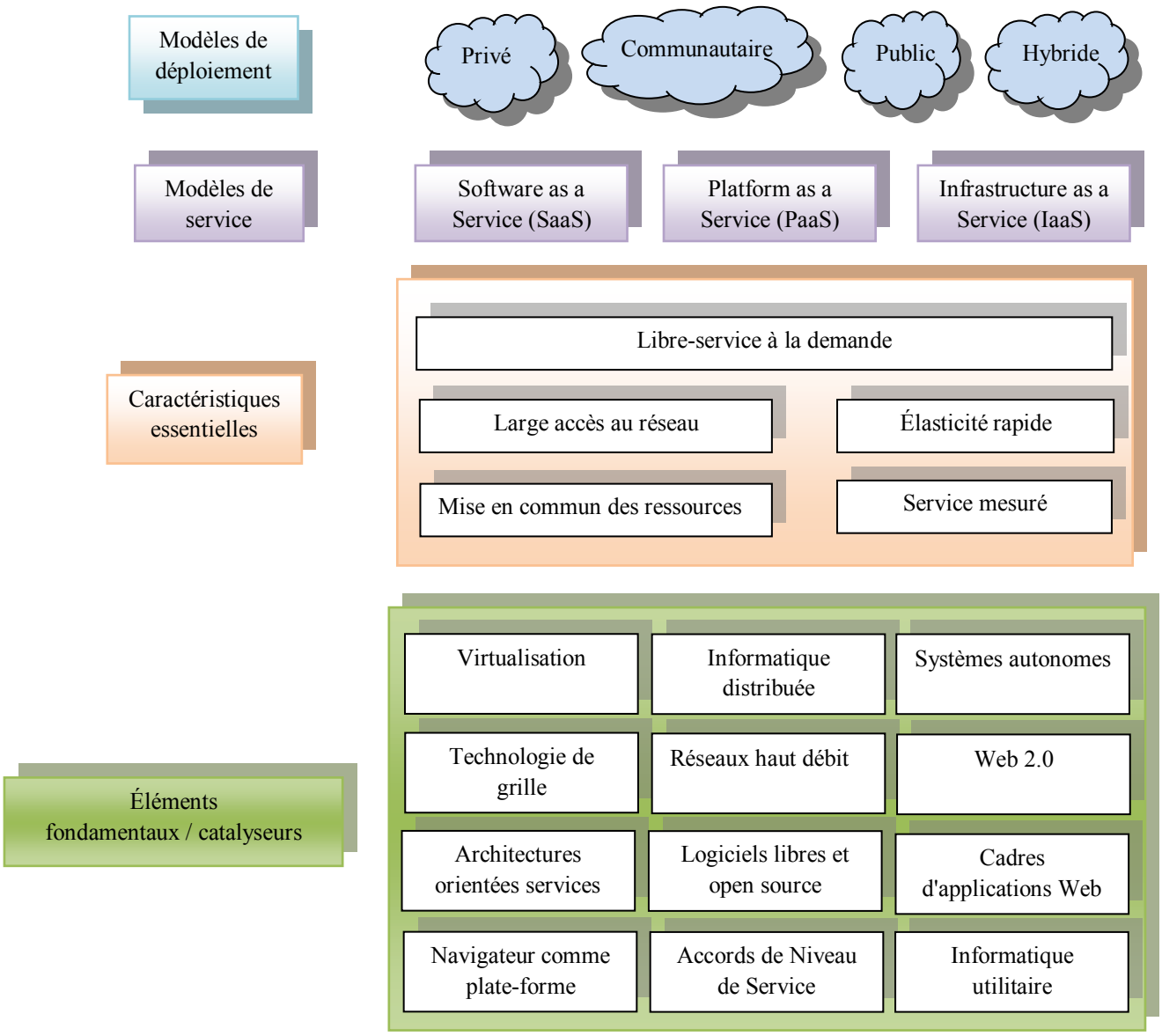


Figure 2.2 : Architecture du Cloud.

4.1. Les différents modèles de services de Cloud computing

Il existe trois modèles de service pour le Cloud Computing (voir *figure 2.3*), selon la façon dont un service est fourni à l'utilisateur, le degré de contrôle que l'utilisateur a sur les ressources et le type de ressources que l'utilisateur a demandé. Ceux-ci sont [10, ALZ 16, MEL 11]:

- ***L'Infrastructure as a Service (IaaS)*** fournit des serveurs virtuels et physiques ainsi que des ressources réseau et de stockage à la discrétion des consommateurs. Le consommateur

n'a pas accès à l'infrastructure Cloud sous-jacente comme le type et la puissance des serveurs, mais il peut mettre en service autant de ressources qu'il le souhaite en termes de stockage et de puissance de calcul. Il a également le choix d'exécuter le système d'exploitation de son choix. par exemple : les clusters Amazon EC2 et Microsoft Azure (qui fournit à la fois Linux et Windows VM). De même, le stockage en nuage est un exemple spécial d'IaaS où les consommateurs ne sont concernés que par l'espace de stockage.

- ***La Platform as a Service (PaaS)*** offre à l'utilisateur la possibilité de développer, déployer et gérer sur l'infrastructure Cloud des applications créées ou acquises par le consommateur et créées à l'aide de langages de programmation, de bibliothèques, de services et d'outils pris en charge par le fournisseur. Le fournisseur de services PaaS contrôle généralement des ressources telles que la puissance de stockage et de calcul. Le consommateur n'a aucun contrôle sur l'infrastructure sous-jacente comme les serveurs, le stockage, le système d'exploitation et le nombre de processeurs, mais contrôle les applications déployées et éventuellement les paramètres de configuration de l'environnement d'hébergement d'applications. Par exemple : les services d'hébergement Web tels que Microsoft Azure Web Services [11], Amazon Web Services [12] et la plate-forme d'hébergement d'applications [13].

- ***Le Software as a Service (SaaS)*** : les consommateurs ont la possibilité d'accéder et d'utiliser les applications du fournisseur de services fonctionnant sur une infrastructure Cloud. Les applications sont accessibles à partir de divers périphériques clients via une interface client léger, telle qu'un navigateur Web (par exemple, un courrier électronique basé sur le Web) ou une interface de programme. Les utilisateurs s'abonnent normalement à ces services sur une base mensuelle ou annuelle, et ils ont peu de contrôle sur ces applications. Par exemple : Microsoft Office 365 [14], Microsoft Skype [15], Google Apps [16] et Sales force [17].

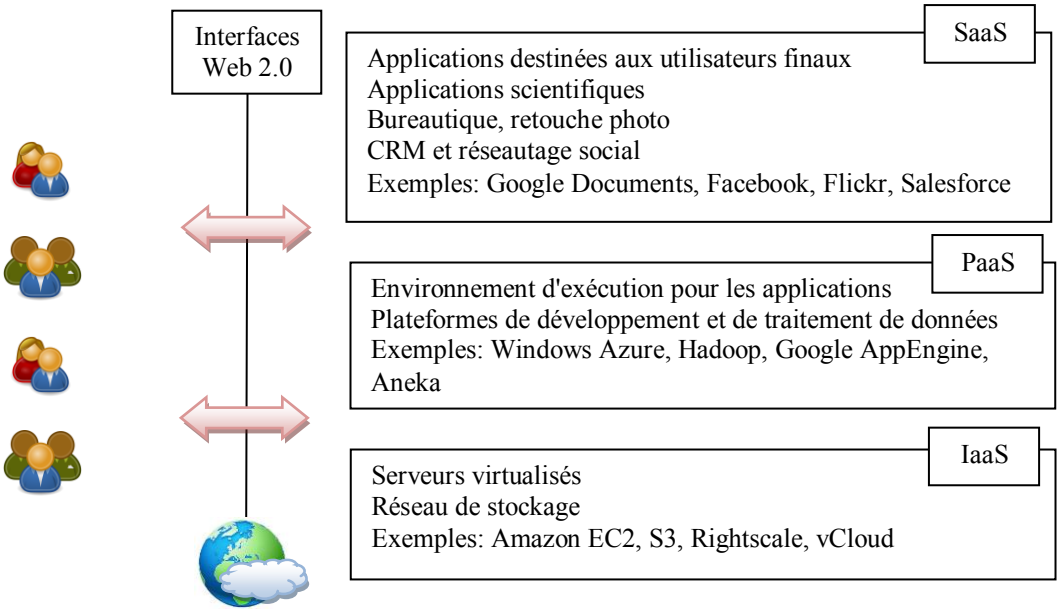


Figure 2.3 : Les trois modèles de service Cloud.

4.2. Modèles de déploiement du Cloud computing

Il existe quatre modèles de déploiement du Cloud Computing(voir *figure 2.4*). Ceux-ci sont [ALZ 16, MEL 11]:

- ***Cloud privé :*** l'infrastructure Cloud est provisionnée pour une utilisation exclusive par une seule organisation comprenant plusieurs consommateurs (par exemple, des unités commerciales). Il peut être détenu, géré et exploité par l'organisation, un tiers ou une combinaison de ceux-ci, et il peut exister dans ou hors des locaux. Le principal avantage de ce déploiement est que la sûreté et la sécurité des données de l'entreprise et d'autres informations vitales sont assurées car personne du monde extérieur n'a accès au Cloud.
- ***Cloud communautaire :*** dans le Cloud communautaire, l'infrastructure Cloud est partagée par une communauté d'organisations qui ont des objectifs communs et travaillent sur des projets similaires (par exemple, mission, exigences de sécurité, politique et considérations de conformité). Donc, dans ce cas, il est logique d'avoir un environnement partagé accessible à toutes les personnes concernées avec des privilèges spécifiques qui leur sont attribués. L'infrastructure peut être détenue, exploitée et gérée par une ou plusieurs des organisations de la communauté, un tiers ou une combinaison de ceux-ci, et il peut exister dans ou hors des locaux.
- ***Cloud public:*** le Cloud public est ouvert à l'accès au grand public et il peut appartenir à une entreprise, une université ou une institution gouvernementale. Un Cloud public

appartient au fournisseur de services Cloud à ses propres locaux. La plupart des fournisseurs de SaaS utilisent ce déploiement pour servir leurs consommateurs, par exemple tous les fournisseurs de stockage Cloud utilisent un modèle de Cloud public pour allouer du stockage à partir d'un pool de ressources partagées. C'est également le modèle le plus commun rencontré par les gens, et la plupart des gens pensent que c'est le seul modèle Cloud.

- ***Cloud hybride:*** le modèle de Cloud hybride est une composition de deux ou plusieurs infrastructures Cloud distinctes (privées, communautaires ou publiques) qui restent des entités uniques, mais sont liées par une technologie standardisée ou propriétaire qui permet la portabilité des données et des applications (par exemple, l'éclatement du Cloud pour l'équilibrage de charge entre des Clouds). Ces modèles sont généralement utilisés lorsque les entreprises ont besoin d'une infrastructure sécurisée pour leur stockage, mais d'autres tâches peuvent être effectuées sur des infrastructures Cloud publiques ou communautaires.

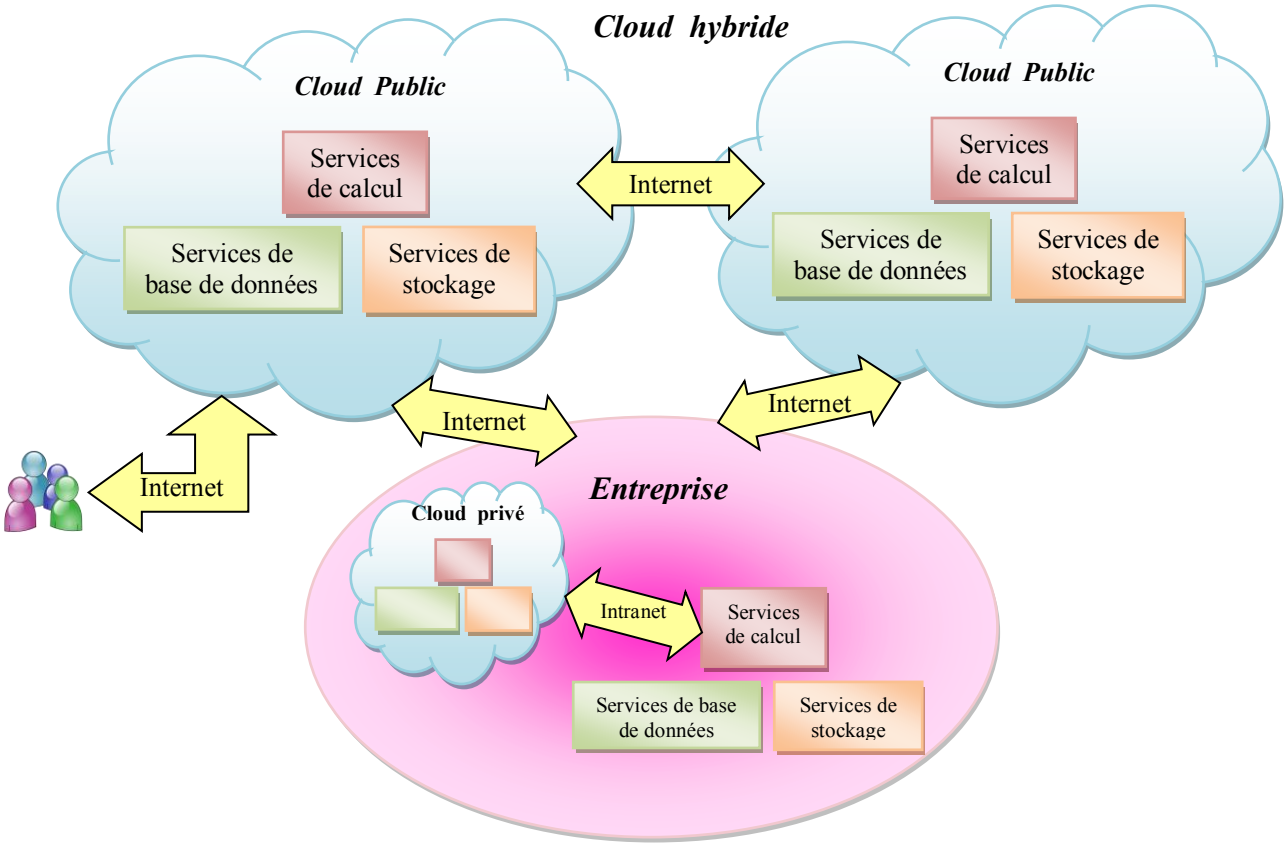


Figure 2.4 : Modèles de déploiement.

4.3. Les Cinq caractéristiques essentielles du Cloud computing

L'institut national des normes et de la technologie (*NIST :* National Institute of Standards and Technology) a définit cinq caractéristiques principales que chaque technologie Cloud devrait intégrer. Elles sont [ALZ 16, MEL 11]:

***Libre-service à la demande* :** Un consommateur peut fournir unilatéralement des capacités informatiques, comme le nombre de clusters Cloud, l'heure du serveur et le stockage réseau, selon les besoins, sans l'intervention d'un fournisseur de services humains.

Large accès au réseau: les services Cloud devraient être facilement disponibles via des mécanismes Internet standard sur tous types d'appareils tels que les téléphones mobiles, les tablettes, les ordinateurs de bureau, les ordinateurs portables, les postes de travail, etc.

Mise en commun des ressources : Les ressources informatiques du fournisseur sont regroupées pour servir plusieurs consommateurs à l'aide d'un modèle multi-locataire, avec des différentes ressources physiques et virtuelles attribuées et réaffectées dynamiquement en fonction de la demande des consommateurs de manière sécurisée (Des exemples de ressources : le stockage, le traitement, la mémoire et la bande passante réseau).

Élasticité rapide: les ressources doivent être provisionnées et libérées à la demande, et à tout moment, le consommateur doit disposer exactement de la quantité de ressources dont il a besoin pour son produit. En substance, le consommateur devrait être en mesure d'augmenter ou de réduire les ressources, supprimer ou ajouter des utilisateurs, prévoir plus de machines ou de stockage de manière transparente, et pour lui, les ressources devraient sembler infinies, dont toute quantité peut être provisionnée à tout moment de temps.

Service mesuré : Les systèmes Cloud contrôlent et optimisent automatiquement l'utilisation des ressources en tirant parti d'une capacité de mesure à un certain niveau d'abstraction approprié au type de service (par exemple, stockage, traitement, bande passante et comptes d'utilisateurs actifs). L'utilisation des ressources peut être surveillée, contrôlée et signalée, offrant une transparence pour le fournisseur et le consommateur du service utilisé.

5. Les avantages du Cloud computing

Le Cloud computing présente de nombreux avantages tels que [10, ALZ 16, GUP 17]:

Évolutivité et flexibilité accrues: les Clouds peuvent allouer dynamiquement des ressources informatiques pour leurs consommateurs, à la demande ou via la configuration directe du

consommateur du Cloud. Cela permet aux consommateurs du Cloud de faire évoluer leurs ressources informatiques basées sur le Cloud pour s'adapter aux fluctuations et aux pics de traitement automatiquement ou manuellement. Les consommateurs de services ont la possibilité d'externaliser des parties de l'infrastructure. Il est également flexible pour les consommateurs qui souhaitent passer des dépenses en capital aux dépenses de fonctionnement.

Récupération des données : Lorsque les entreprises commencent à s'appuyer sur les services de Cloud computing, elles n'ont plus besoin de programmes complexes de récupération des données. Les fournisseurs de Cloud computing se chargent de la plupart de ces tâches et ils le font plus vite.

Mises à jour logicielles automatiques : les fournisseurs de Cloud computing se chargent de la maintenance du serveur, y compris des mises à jour de sécurité, permettant ainsi à leurs clients d'allouer le temps et les ressources économisés à d'autres tâches plus stratégiques.

Investissements et frais réduits : Les services de Cloud computing sont généralement fournis selon un modèle de « paiement à l'utilisation ». Pour la plupart des services Cloud, il n'y a pas de coûts de déploiement et d'achat initiaux. Le Cloud réduit les coûts d'investissement du matériel et des logiciels. Il ne nécessite que des frais de propriété. Cela conduit à une réduction du coût d'installation qui est payé par le client au moment de l'installation.

Collaboration accrue : Le Cloud computing améliore la collaboration en permettant à l'ensemble des employés, où qu'ils se trouvent, de synchroniser leurs actions et de travailler sur des documents et des applications partagées simultanément. Ils peuvent aussi suivre leurs collègues et leurs enregistrements afin de recevoir des mises à jour critiques en temps réel.

Contrôle des documents : Le Cloud computing conserve tous les fichiers dans un emplacement central et tout le monde travaille à partir d'une copie centralisée. Les employés peuvent même utiliser la messagerie instantanée lorsqu'ils effectuent des modifications. L'ensemble de ce processus renforce la collaboration, ce qui augmente l'efficacité et améliore les résultats financiers d'une entreprise.

Accessibilité facile: On peut accéder aux applications comme utilitaires sur Internet.

Installation facile: Il est simple d'utiliser et de configurer tous les services. Il ne nécessite pas d'installer de logiciel spécifique pour accéder ou manipuler une application Cloud. L'application peut être mise à jour à tout moment sans avoir à se soucier de la gestion des ressources et des autres problèmes liés à la configuration et à la gestion de l'infrastructure.

Disponibilité et fiabilité: les services Cloud sont toujours disponibles pour les utilisateurs tant qu'ils sont connectés à l'Internet. Le client peut accéder à ses données de n'importe où, au fur et à mesure de ses besoins. Les fournisseurs de Cloud offrent généralement des ressources informatiques résistantes qui sont facilement accessibles en tant qu'utilitaires sur Internet. L'environnement Cloud est assez fiable pour les raisons : les fournisseurs des services Cloud utilisent une sauvegarde complète, l'architecture modulaire du Cloud et aussi sa capacité d'équilibrage de charge.

Aucun coût de maintenance: les fournisseurs de services Cloud n'exigent pas l'installation d'applications sur PC, ce qui réduit les coûts de maintenance. Le consommateur n'a qu'à payer pour les services qu'il a utilisés, et quand l'usure des serveurs ou les lecteurs de données échouent, c'est le service prestataire qui doit supporter les frais de remplacement. Et en cas de catastrophe naturelle, les Clouds ont la fonction de sauvegarde, ainsi les données peuvent être conservées en toute sécurité dans un environnement Cloud.

Meilleures performances: dans un environnement de Cloud computing, l'utilisateur n'a pas besoin d'installer de logiciel lourd sur son propre ordinateur. Cela conduit à augmenter les performances de l'ordinateur.

Capacité de stockage illimitée: à l'aide de services Cloud, le client peut utiliser la capacité de stockage illimitée. Si les besoins de stockage du client augmentent, le client doit payer un peu plus pour utiliser une plus grande capacité de stockage fournie par le serveur Cloud. Ainsi, avec un faible coût d'installation, un utilisateur peut utiliser une capacité de stockage illimitée.

Facilité de mise à l'échelle: les services Cloud peuvent facilement être mis à l'échelle vers le haut et vers le bas selon le désir du consommateur. Par exemple, la capacité de stockage dans le Cloud peut être augmentée jusqu'à To ou elle peut être aussi faible que certains GB.

6. Les inconvénients de Cloud computing

Le Cloud computing a beaucoup d'avantages. Pourtant, certaines entreprises n'ont pas intérêt à l'utilisation du Cloud, pour des raisons techniques et légales. Voici quelques inconvénients que présente le Cloud computing [10, ALZ 16]:

Sécurité des données*:* le plus grand point d'interrogation sur les services Clouds sont la sécurité des données. Plusieurs points sont à étudier. – La sécurité et la confidentialité des données : si le fournisseur de service assure des tests portant sur sa sécurité informatique et si ces tests sont faits de façon régulières. – La sécurité vis-à-vis du stockage : si les données sont

gardées dans un seul disque, ou si elles sont entre plusieurs unités de stockage. – La sécurité des locaux : sont-ils inaccessibles pour des personnes malveillantes ? Certaines applications comme Twitter et Facebook sont très sujets aux attaques. Autrefois, il y a eu une fuite massive de données dans le stockage ICloud, où les comptes ICloud de plusieurs célébrités ont été piratés et leurs photos personnelles ont été publiées en ligne. C'est le plus gros inconvénient du Cloud - vous mettez vos données en ligne où d'autres personnes peuvent potentiellement y accéder en cas de brèche. C'est la primordiale raison que de nombreuses entreprises hésitent à l'utiliser.

Temps d'arrêt: la plupart des services Cloud restent disponibles 24/7 mais pour certains services il y a des temps morts programmés. Cela peut être dû à une maintenance périodique. Parfois, le fournisseur de services ne s'engage que pour une période de temps limitée par jour.

Contrôle limité: les consommateurs ont très peu de contrôle sur leurs produits dans le Cloud. Le plus de contrôle dont ils disposent se trouve dans l'IaaS (Infrastructure as a Service) modèle, dans lequel ils peuvent provisionner des machines virtuelles entières et les personnaliser en fonction de leurs besoins. En SaaS, ils ont le moins de contrôle car ils ne peuvent configurer que certaines parties de l'application mais ils n'ont aucun contrôle sur quoi que ce soit d'autre.

Dépendance du réseau: un autre inconvénient majeur du Cloud est sa dépendance à Internet, ou certain autre réseau local en cas de Cloud privé. Même si Internet est devenu omniprésent dans le monde développé, il trouve encore ses marques dans les pays en développement. Donc, utiliser le Cloud pour vos produits signifie essentiellement que vous ignorez cette partie de la population mondiale qui est sans Internet, et pour certains produits, ils peuvent être une population importante. De plus, malgré les services de données 3G et 4G dans les Smartphones, les gens ne sont pas encore très friands de l'utilisation des données mobiles et essentiellement pour les applications à forte intensité de données, ils préfèrent utiliser le Wi-Fi, qui n'est pas toujours disponible partout.

Aucune responsabilité légale pour les vendeurs: Même si les fournisseurs de services Cloud hébergent les données des utilisateurs et d'autres informations sensibles et assurent la meilleure sécurité disponible, ils déclinent toute responsabilité en cas de manquement potentiel. Les données transférées dans le Cloud ne sont pas forcément présentes sur le territoire national : elles peuvent l'être, comme elles peuvent être dans un autre pays. Par conséquent, sauf mention contraire du prestataire de service, on ne sait pas précisément à quel endroit sont stockées les données. De plus, on n'a aucun accès physique à ces données.

7. Exemples du Cloud computing

Voici quelques exemples concrets du Cloud computing:

1. Clusters Amazon EC2- serveurs virtuels pour le stockage et l'informatique [18].
2. Microsoft Azure - fournit plus de 50 services cloud (PaaS et SaaS) [11].
3. Microsoft Office 365 [14] et Google Docs – SaaS [19].
4. OneDrive [20], Google Drive [21], iCloud [22], Dropbox [23]– Cloud Espace de rangement.

8. Le Big Data

Le taux de croissance des données générées et stockées a augmenté de façon exponentielle. Cette explosion de données crée des opportunités pour de nouvelles façons de combiner et d'utiliser les données pour trouver de la valeur, ainsi que des défis importants en raison de la taille des données gérées et analysées. Un changement important concerne la quantité de données non structurées. Historiquement, les données structurées ont généralement été au centre de la plupart des analyses d'entreprise et ont été gérées grâce à l'utilisation du modèle de données relationnelles. Récemment, la quantité de données non structurées, telles que des micro-textes, des pages Web, des données de relation, des images et des vidéos, a explosé, et la tendance indique une augmentation de l'incorporation de données non structurées pour générer de la valeur. L'avantage central de l'analyse Big Data est la capacité de traiter de grandes quantités et différents types d'informations. Le Big Data n'implique pas que les volumes de données actuels sont simplement «plus gros» qu'auparavant, ou plus grands que les techniques actuelles ne peuvent les gérer efficacement. Le besoin d'une plus grande performance ou efficacité se produit sur une base continue. Cependant, le Big Data représente un changement fondamental dans l'architecture nécessaire pour gérer efficacement les ensembles de données actuels [CHAN 15, MEL 11].

Définition:

Selon l'Institut national des normes et technologie (NIST), *"Big Data refers to the inability of traditional data architectures to efficiently handle the new datasets. Characteristics of Big Data that force new architectures are:* ***Volume*** *(i.e., the size of the dataset);* ***Variety*** *(i.e., data from multiple repositories, domains, or types);* ***Velocity*** *(i.e., rate of flow); and* ***Variability*** *(i.e., the change in other characteristics). These characteristics—*

volume, variety, velocity, and variability—are known colloquially as the 'Vs' of Big Data. While many other V's have been attributed to Big Data, only the above four drive the shift to new parallel architectures for data-intensive applications, in order to achieve cost-effective performance. These Big Data characteristics dictate the overall design of a Big Data system, resulting in different data system architectures or different data life cycle process orderings to achieve needed efficiencies" [CHAN 15, HAS 15].

"Le Big Data fait référence à l'incapacité des architectures de données traditionnelles à gérer efficacement les nouveaux ensembles de données. Les caractéristiques du Big Data qui forcent de nouvelles architectures sont: ***Volume*** *(c'est-à-dire la taille de l'ensemble de données);* ***Variété*** *(c'est-à-dire, données provenant de plusieurs référentiels, domaines ou types);* ***vitesse*** *(c.-à-d., Débit); et* ***Variabilité*** *(c'est-à-dire le changement d'autres caractéristiques). Ces caractéristiques - volume, variété, vitesse et variabilité - sont connues sous le nom de «Vs» du Big Data (*voir *figure2.5). Alors que de nombreux autres V ont été attribués au Big Data, seuls les quatre ci-dessus entraînent le passage à de nouvelles architectures parallèles pour les applications gourmandes en données, afin d'obtenir des performances rentables. Ces caractéristiques Big Data dictent la conception globale d'un système Big Data, ce qui se traduit par différentes architectures de système de données ou différents ordres de processus de cycle de vie des données pour atteindre les efficacités nécessaires"* [CHAN 15, HAS 15].

Cette définition se réfère à la générosité des données numériques de divers sources de données dans le cadre de Digital Earth, qui se concentrent sur les aspects géographiques des méga données de l'information sociale, Observation de la Terre (EO : Earth observation), service d'observation des capteurs (SOS : sensor observation service), cyber-infrastructure (CI), médias sociaux et informations commerciales [HAS 15].

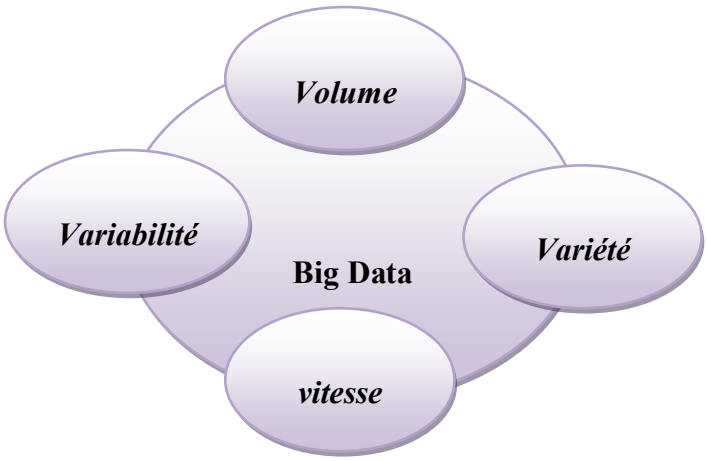


Figure 2.5 : les Quatre Vs de Big Data.

8.1. Cycle de vie de Big Data

Le cycle de vie des données comprend les quatre étapes suivantes [CHAN 15]:

a) *Collecte*: cette étape rassemble et stocke les données dans leur forme d'origine (c'est-à-dire les données brutes).
b) *Préparation*: cette étape implique la collecte de processus qui convertissent les données brutes en informations nettoyées et organisées.
c) *Analyse*: cette étape implique les techniques qui produisent des connaissances synthétisées à partir d'informations organisées.
d) *Action*: cette étape implique des processus qui utilisent les connaissances synthétisées pour générer de la valeur pour l'entreprise.

Dans l'entrepôt de données traditionnel, le processus de traitement des données a suivi l'ordre ci-dessus (c.-à-d. collecte, préparation, stockage et analyse). Le modèle relationnel a été conçu de manière à optimiser l'analyse prévue. Les différentes caractéristiques du Big Data ont influencé les changements dans l'ordre des processus de traitement des données [CHAN 15].

Voici des exemples de ces changements [CHAN 15]:

a) *Entrepôt de données:* un stockage persistant se produit après la préparation des données.
b) *Système de volume Big Data:* les données sont stockées immédiatement sous forme brute avant la préparation; la préparation se produit lors de la lecture et est appelée «schéma en lecture».
c) *Application de la vélocité du Big Data:* la collecte, la préparation et l'analyse (alerte) se produisent à la volée, et peuvent éventuellement inclure un résumé ou une agrégation avant le stockage.

8.2. Classification des Big Data

Les Big Data sont classées en différentes catégories pour mieux comprendre leurs caractéristiques. La *figure 2.6* montre les nombreuses catégories de Big Data. La classification est importante en raison des données à grande échelle dans le Cloud. La classification est basée sur cinq aspects [HAS 15] :

a) Les sources de données (data sources).
b) Le format du contenu (content format).

c) Les magasins de données (data stores).

d) La mise en scène des données (data staging).

e) Le traitement des données (data processing).

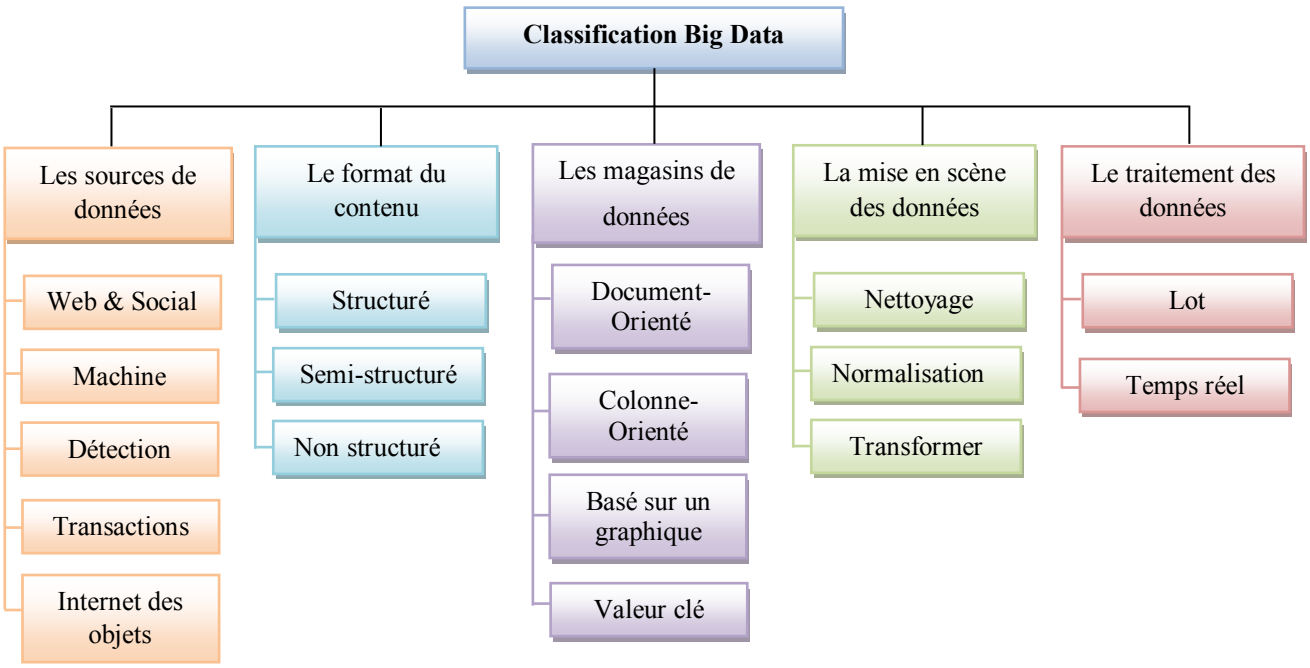


Figure 2.6 : Classification des Big Data.

8.3. La relation entre le Cloud computing et le Big Data

Le Cloud computing et le Big Data sont conjoints. Le Big Data offre aux utilisateurs la possibilité d'utiliser l'informatique de base pour traiter les requêtes distribuées sur plusieurs ensembles de données et renvoyer les ensembles résultants en temps opportun. Le Cloud computing fournit le moteur sous-jacent grâce à l'utilisation de Hadoop [24], une classe de plates-formes de traitement de données distribuées. Le Cloud computing fournit une solution évolutive et rentable au défi du Big Data. L'utilisation du Cloud computing dans les Big Data est illustrée à la *figure2.7* [HAS 15, ODR 13].

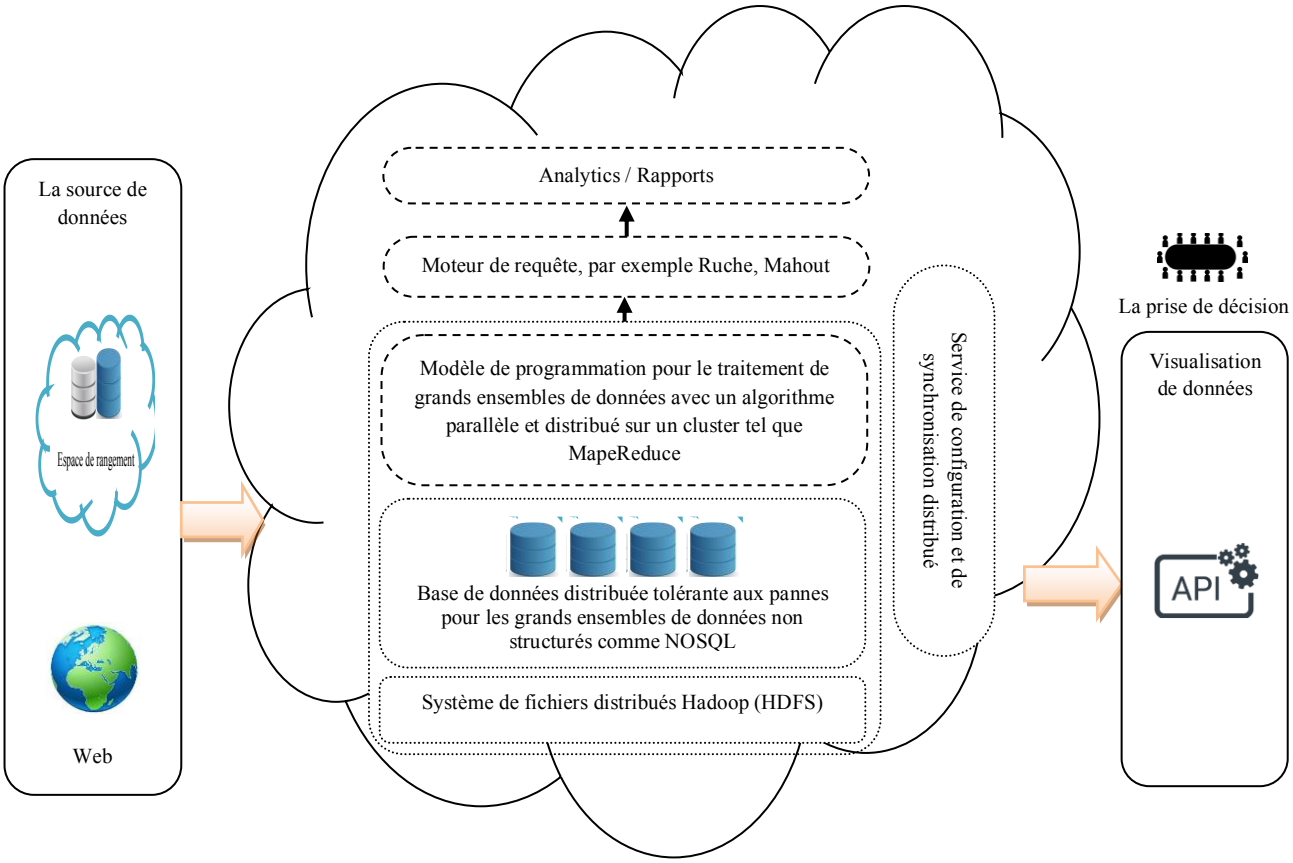


Figure 2.7 : Utilisation du Cloud computing dans les Big Data.

8.4. Exemples de Cloud computing et de Big Data

Le Cloud computing est utilisé dans de nombreux environnements et déployé dans de nombreux domaines. Voici quelques exemples d'utilisation de Cloud computing pour le Big Data [ALM 16, ANS 16, SHA 18, ZHA 15]:

a) *Soins de santé (* Health care)*:* les technologies modernes dans l'environnement des soins de santé ont produit des quantités de données rapides, énormes et hétérogènes qui devaient être traitées et utilisées de manière utile. Le Cloud computing et l'analyse des Big Data aident les différents hôpitaux à gérer ces Big Data médicales issues des rapports des patients, des examens cliniques et d'autres sources, en collectant et en traitant les données collectées de manière gérable pour aider à améliorer les services offerts par les centres de santé à travers les opportunités offertes par le Cloud computing.

b) *Éducation et apprentissage:* la transformation des méthodes traditionnelles d'apprentissage en apprentissage en ligne a conduit à la mise en ligne d'énormes

quantités d'informations. Ces informations doivent être traitées et transformées en connaissances et ressources pédagogiques qui peuvent ensuite être utilisées sur le Cloud pour réduire les problèmes de coûts.

c) *Développement économique:* les gouvernements intelligents de nombreux pays savent comment utiliser l'analyse des Big Data pour promouvoir leurs industries. Les services peuvent être fournis aux citoyens à un coût beaucoup plus faible que les méthodes traditionnelles. Autant de données que possible sur les citoyens peuvent être collectées et les services nécessaires leur sont offerts de la manière la plus simple.

9. Conclusion

Le Cloud computing est l'une des plus passionnantes technologies au cours de la dernière décennie. Il est une technologie émergente qui offre des ressources informatiques distribuées, inégalées et à des coûts d'infrastructure et d'exploitation abordables. Il ouvre le monde de l'informatique à une gamme plus large d'utilisations et augmente la facilité d'utilisation en donnant accès via n'importe quelle connexion Internet. Cependant, cette facilité accrue s'accompagne également d'inconvénients. Vous avez moins de contrôle sur qui a accès à vos informations et peu ou pas de savoir où elles sont stockées. Vous devez également être conscient des risques de sécurité liés au stockage des données sur le Cloud. Le Cloud est une cible importante pour les individus malveillants et peut présenter des inconvénients car il est accessible via une connexion Internet non sécurisée. Si vous envisagez d'utiliser le Cloud, assurez-vous d'identifier les informations que vous publierez dans le Cloud, qui aura accès à ces informations et ce dont vous aurez besoin pour vous assurer qu'elles sont protégées. De plus, il faut connaitre vos options en termes de type de Cloud qui conviendra le mieux à vos besoins, quel type de fournisseur vous sera le plus utile et quelle est la réputation et les responsabilités des fournisseurs que vous envisagez avant de vous inscrire. Néanmoins, le Cloud est là pour rester dans un avenir prévisible et il serait prudent pour de nombreuses entreprises de l'adopter.

Dans ce chapitre, nous avons présenté les définitions de base du Cloud computing. Nous avons étalé son évolution, son architecture, ses différents modèles de service et de déploiement ainsi que ses caractéristiques. Nous avons cité les avantages et les inconvénients du Cloud. Nous avons détaillé la relation entre le Cloud computing et le Big Data.

Le prochain chapitre sera consacré aux arbres de décision.

Chapitre 3

Les arbres de décision

1. Introduction

Les arbres de décision ou méthodes de partitionnement récursif ont été introduits dès les années 60. Un arbre de décision peut analyser les informations contenues dans une vaste source de données et identifier des règles et des relations précieuses. Habituellement, les arbres de décision sont déployés à des fins de prédiction / classification [BAT 18]. Les arbres de décisions sont des techniques très populaires par leur efficacité et leur simplicité dans le domaine de la classification supervisée. Le succès des arbres de décision réside en grande partie dans leur lisibilité, par opposition aux algorithmes de "boîte noire" tels que les réseaux de neurones [QUE 08]. L'un des plus grands avantages est que les utilisateurs n'ont pas besoin de connaître beaucoup de connaissances de base dans le processus d'apprentissage, ce qui est également le plus gros inconvénient [YUA 18]. Ils fournissent une représentation graphique du modèle facilement interprétable. Le modèle final est constitué d'un nœud racine et des nœuds intermédiaires, des branches et des feuilles. La racine est le point d'entrée à l'arbre. Les feuilles représentent les valeurs classes à prédire. Les branches représentent les résultats de test relatif à chaque nœud. Pour effectuer une classification, l'arbre est parcouru de la racine aux feuilles selon une série de tests à chaque niveau de l'arbre. Aujourd'hui très largement utilisés dans différents domaines ils représentent un classifieur standard, facilement modulable selon les applications.

Dans ce chapitre un bref historique de l'apparition des arbres de décision dans la littérature, Un exemple introductif pour mieux comprendre la notion des arbres de décision, le principe général de sa construction est présenté, le concept d'apprentissage d'un arbre de décision est bien détaillé, l'algorithme ID3 (Iterative Dichotomiser *3*), la méthode CART (Classification And Regression Trees) et les forêts aléatoires sont explorés. Ensuite, une exposition de ses avantages et de ses inconvénients.

2. Bref historique

Les premiers travaux s'apparentant plus ou moins aux arbres de décision furent présentés par Belson en 1956 dans un cadre statistique de *régression* [BELS 56]. Un peu plus tard, les premiers algorithmes d'arbres de décision firent leur apparition, le premier fut lui

aussi proposé par les statisticiens Morgan et Sonquist en 1963[MOR 63], qui, les premiers, ont utilisé les arbres de régression dans un processus de prédiction et d'explication (AID – Automatic Interaction Detection). Il s'en est suivi toute une famille de méthodes, étendues jusqu'aux problèmes de discrimination et classement, qui s'appuyaient sur le même paradigme de la représentation par arbres (THAID : THeta Automatic Interaction Detection - Morgan et Messenger, 1973 [MOR 73] ; CHAID : CHi-squared Automatic Interaction Detection - Kass, 1980 [KAS 80]).

Cependant ce n'est qu'en 1984 que les arbres de décision devinrent réellement populaires avec le travail de Breiman et al. [BRE 84]. Ce travail, intitulé *CART* (Classification And Regression Trees), s'inscrit lui aussi dans un cadre purement statistique.

En apprentissage automatique, la plupart des travaux s'appuient sur la théorie de l'information. C'est Quinlan qui en 1986 fut le premier informaticien à s'intéresser de près aux arbres de décision avec l'algorithme *ID*3 (Iterative Dichotomiser *3*) [QUI 86] qui, lui même, rattache ses travaux à ceux de Hunt (1962) [HUN 62]. Quinlan a été un acteur très actif dans la deuxième moitié des années 80 avec un grand nombre de publications où il propose un ensemble d'heuristiques pour améliorer le comportement de son système. Son approche a pris un tournant important dans les années 90 lorsqu'il présenta la méthode C4.5 qui est l'autre référence incontournable dès lors que l'on veut citer les arbres de décision [QUI 14]. Il existe bien une autre évolution de cet algorithme, C5.0, mais étant implémentée dans un logiciel commercial, il n'est pas possible d'en avoir le détail. Grâce à ce travail, c'est toute une communauté scientifique (celle de l'*apprentissage automatique* qui est essentiellement composé d'informaticiens) qui bénéficia de cet outil très puissant que représentent les arbres de décision.

De très nombreux autres algorithmes d'arbres de décision furent ensuite proposés tel que ***CHAID*** [KAS 80], ***SLIQ*** (Supervised Learning In Quest, où Quest est le projet d'exploration de données du centre de recherche IBM Ahnaden) [MEH 96], ***QUEST*** (Quick, Unbiased, Efficient, Statistical Tree) [LOH 97], ***CFDT*** (Clustering Feature Decision Tree) [XU 11].

Malgré tout *CART* et *C*4.5 restent à ce jour les deux algorithmes de construction d'arbres de décision les plus reconnus et utilisés. Alors que le premier ne produit que des arbres binaires (chaque nœud ne donne naissance qu'à deux branches) et comprend une méthode d'élagage dont l'efficacité n'est plus à démontrer, le second est quant à lui plutôt adapté aux attributs qualitatifs (chaque nœud sur un tel attribut donne naissance à un nombre

de branches égal au nombre de modalités de l'attribut) et produit des arbres moins profond (un attribut ne peut apparaître qu'une fois le long d'un chemin partant du nœud initial à une feuille).

On peut remarquer que très souvent les statisticiens ont tendance à privilégier *CART* alors que leurs confrères informaticiens se tournent plus facilement vers *C*4.5.

3. Un exemple introductif

Nous reprendrons et déroulerons un exemple qui est présenté dans l'ouvrage de Quinlan [QUI 14]. Le fichier est composé de 14 observations avec quatre attributs : Ensoleillement, Température, Humidité, Vent et l'attribut à prédire Jouer. Il s'agit d'expliquer le comportement des individus par rapport à un jeu {jouer, ne pas jouer} à partir des prévisions météorologiques (*Tableau 3.1*).

Numéro	Ensoleillement	Température (°F)	Humidité (%)	Vent	Jouer
1	Soleil	75	70	Oui	Oui
2	Soleil	80	90	Oui	Non
3	Soleil	85	85	Non	Non
4	Soleil	72	95	Non	Non
5	Soleil	69	70	Non	Oui
6	Couvert	72	90	Oui	oui
7	Couvert	83	78	Non	oui
8	Couvert	64	65	Oui	oui
9	Couvert	81	75	Non	oui
10	Pluie	71	80	Oui	Non
11	Pluie	65	70	Oui	Non
12	Pluie	75	80	Non	Oui
13	Pluie	68	80	Non	Oui
14	Pluie	70	96	Non	Oui

Tableau 3.1 : Données "weather" [QUI 14].

L'arbre de décision correspondant est le suivant (*figure 3.1*) :

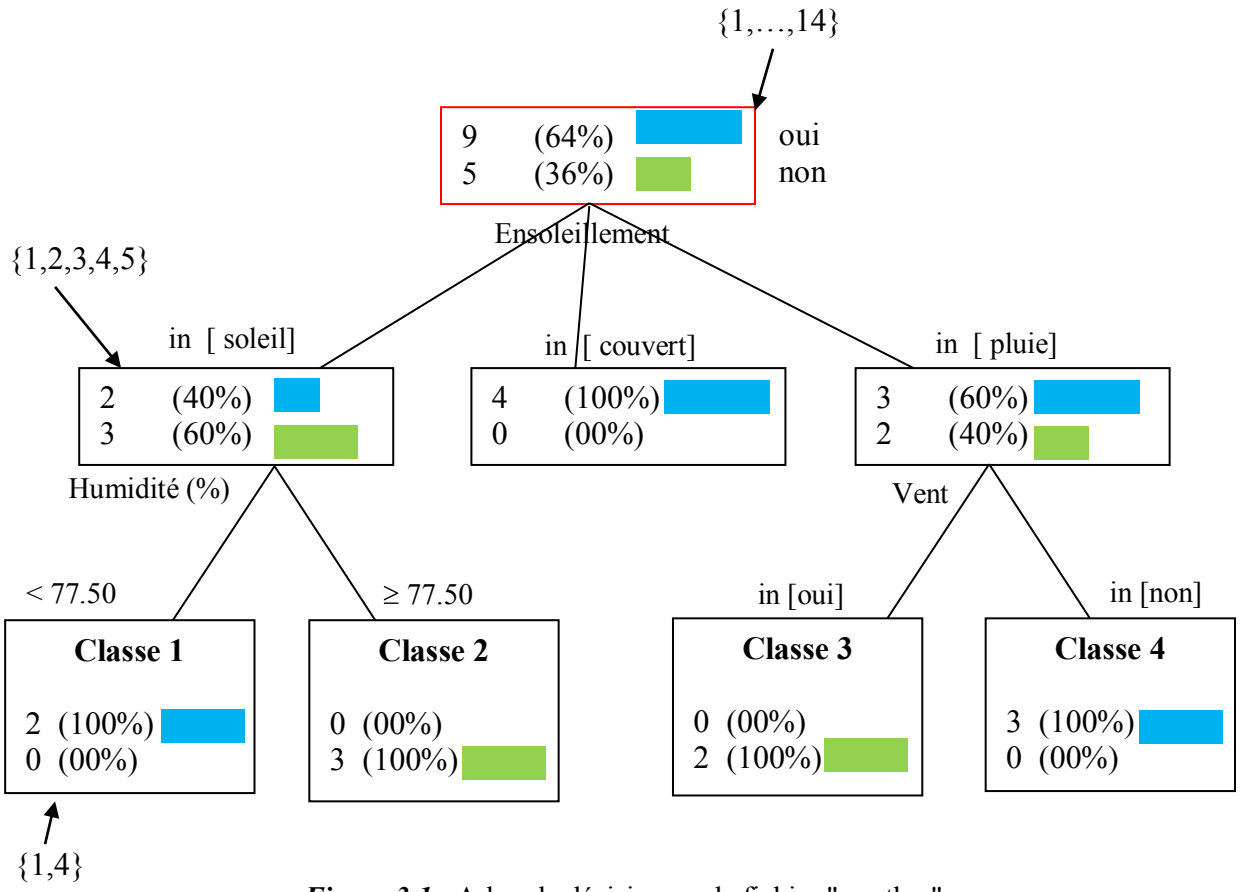


Figure 3.1 : Arbre de décision sur le fichier "weather".

- ✓ Le premier sommet est appelé la « racine » de l'arbre. Il est situé sur le premier niveau. Nous y observons la distribution de fréquence de la variable à prédire «Jouer». Nous constatons qu'il y a bien 14 observations, dont 9 « oui » (ils vont jouer) et 5 «non ».

- ✓ La variable « ensoleillement » est la première variable utilisée ; on parle de variable de segmentation. Comme elle est composée de 3 modalités {soleil, couvert, pluie}, elle produit donc 3 sommets enfants.

- ✓ La première arête (la première branche), à gauche, sur le deuxième niveau, est produite à partir de la modalité « soleil » du variable « ensoleillement ». Le sommet qui en résulte couvre 5 observations correspondant aux individus {1, 2, 3, 4, 5}, la distribution de fréquence nous indique qu'il y a 2 « jouer = oui » et 3 « jouer = non ».

- ✓ La seconde arête, au centre, correspond à la modalité « couvert » de la variable de segmentation « ensoleillement » ; le sommet correspondant couvre 4 observations, tous

ont décidé de jouer (dans le tableau ce sont les individus n°6 à 9). Ce sommet n'ayant plus de sommets enfants, ce qui est normal puisqu'il est « pur » du point de vue de la variable à prédire, il n'y a pas de contre-exemples. On dit qu'il s'agit d'une feuille de l'arbre.

- ✓ Reprenons le nœud le plus à gauche sur le deuxième niveau de l'arbre. Ce sommet, qui n'est pas pur, est segmenté à l'aide de la variable « humidité ». Comme le descripteur est continu, il a été nécessaire de définir un seuil dit de discrétisation qui permet de produire le meilleur partitionnement. Dans notre exemple, le seuil qui a été choisi est 77.5 %. Il a permis de produire deux feuilles complètement pures.

- ✓ Ce processus est réitéré sur chaque sommet de l'arbre jusqu'à l'obtention de feuilles pures. Il s'agit bien d'un arbre de partitionnement : un individu ne peut être situé dans deux feuilles différentes de l'arbre.

- ✓ Le modèle de prédiction peut être lu très facilement. On peut traduire un arbre en une base de règles sans altération de l'information. Le chemin menant d'un sommet vers la racine de l'arbre peut être traduit en une partie prémisse d'une règle de prédiction de type attribut-valeur « SI variable 1 = valeur 1 ET variable 2 = valeur 2 … ».

- ✓ Pour classer un nouvel individu, il suffit de l'injecter dans l'arbre, et de lui associer la conclusion attachée à la feuille dans laquelle il aboutit.

En effet, toutes les données ayant l'attribut Ensoleillement="Soleil" et l'attribut Humidité> 77.5 appartiennent à la **classe 1**. Toute nouvelle donnée peut être classée en testant ses valeurs d'attributs l'un après l'autre en commençant de la racine jusqu'à atteindre une feuille c'est-à-dire une décision. Pour construire un tel arbre, plusieurs algorithmes existent : ID3, CART, C4.5,.... On commence généralement par le choix d'un attribut puis le choix d'un nombre de critères pour son nœud. On crée pour chaque critère un nœud concernant les données vérifiant ce critère. L'algorithme continue d'une façon récursive jusqu'à obtenir des nœuds concernant les données de chaque même classe.

Algorithme CONSTRUIRE-ARBRE (*D* : ensemble de données)

1: Créer nœud *N*
2: Si tous les exemples de *D* sont de la même classe *C* alors
3: Retourner *N* comme une feuille étiquetée par *C* ;
4: Si la liste des attributs est vide alors
5: Retourner *N* Comme une feuille étiquetée de la classe de la majorité dans *D*;
6: Sélectionner l'attribut *A* du meilleur Gain dans *D*;
7: Etiqueter *N* par l'attribut sélectionné ;
8: Liste d'attributs Liste d'attributs - *A*;
9: Pour chaque valeur V_i de *A*
10: Soit D_i l'ensemble d'exemples de *D* ayant la valeur de $A = V_i$;
11: Attacher à *N* le sous arbre généré par l'ensemble D_i et la liste d'attributs
12: Fin Pour ;
13: Fin ;

Cette simplicité apparente ne doit pas cacher des problèmes réels qui se posent lors de la construction de l'arbre [RAK 05] :

1. La première question est le choix de la variable de segmentation sur un sommet. Par exemple, pourquoi avons-nous choisi la variable « ensoleillement » à la racine de l'arbre ? Nous remarquons que le choix d'une variable de segmentation est relatif au sommet et non au niveau que nous sommes en train de traiter : les sommets à gauche et à droite du second niveau ont été segmentés avec des variables différentes. Il nous faut donc un indicateur (une mesure) qui permet d'évaluer objectivement la qualité d'une segmentation et ainsi de sélectionner le meilleur parmi les descripteurs candidats à la segmentation sur un sommet.
2. Pour mettre en œuvre la variable « humidité » au second niveau de l'arbre, nous avons été obligés de fixer un seuil (77.5%) pour segmenter le groupe d'observations. Comment a été fixé ce seuil ? Une fois que le seuil a été défini, comment sont mis en concurrence les variables continues et catégorielles pour la segmentation d'un sommet?
3. L'objectif est de produire un partitionnement pur des observations de la base, ce qui est le cas de notre exemple. Que faire lorsque cela n'est pas possible ? De manière plus générale, est-ce qu'un partitionnement totalement pur est souhaitable sur le fichier

de données ; est-ce qu'il est possible d'utiliser des règles plus efficaces pour définir la taille adéquate de l'arbre de décision ?

4. Enfin, si la prise de décision sur une feuille semble naturelle lorsqu'elle est pure, quelle est la règle de décision optimale lorsque qu'une feuille contient des représentants des différentes modalités de la variable à prédire ?

Répondre à ces questions permet de définir une méthode d'induction des arbres de décision à partir de données. La très grande majorité des méthodes recensées à ce jour respectent ce schéma, il est alors facile de les positionner les unes par rapport aux autres. On comprend également que le champ des stratégies possibles étant restreint, il paraît illusoire de trouver une avancée miraculeuse sur un des 4 points ci-dessus qui permettrait de surclasser les techniques existantes. C'est pour cette raison que, si la communauté scientifique a été très prolixe dans les années 90 en explorant de manière quasi-exhaustive les variantes sur chacun de ces points, les comparaisons sur données réelles ont montré qu'elles produisaient des arbres avec des performances similaires. Des différences peuvent cependant apparaître dans des cas particuliers où telle ou telle caractéristique d'une variante que l'on a choisie s'avère mieux adaptée [RAK 05].

➢ Choix de la variable de segmentation

Il s'agit de choisir parmi les attributs des données, celui qui les sépare le mieux du point de vue de leurs classes déjà connues. Pour choisir le meilleur attribut, on calcule pour chacun une valeur appelée "Gain" qui dépend des différentes valeurs prises par cet attribut. Cette mesure est basée sur les recherches en théorie d'informations menées par C.Shannon [SHAN 48].

Par exemple, l'algorithme ID3 utilise le concept d'entropie introduite initialement par Shannon en 1948 [SHAN 48].

Soit un ensemble X d'exemples dont une proportion p_+ sont positifs et une proportion p_- sont négatifs. (Bien entendu, $p_+ + p_- = 1$) L'entropie de X est :

$$H(X) = -p_+ \log_2(p_+) - p_- \log_2(p_-) \qquad (3.1)$$

Bien sur

$$0 \leq H(X) \leq 1$$

Si $p_+ = 0$ ou $p_- = 0$, alors $H(X) = 0$. Ainsi, si tous exemples sont soit tous positifs, soit tous négatifs, l'entropie de la population est nulle. Si $p_+ = p_- = 0.5$, alors $H(X) = 1$. Ainsi, s'il y a autant de positifs que de négatifs, l'entropie est maximale.

$$Gain\,(X, a_j) = H(X) - \sum_{v \in valeurs\,(a_j)} \frac{\left|X_{a_j=v}\right|}{|X|} H\left(X_{a_j=v}\right) \qquad (3.2)$$

Où $X_{a_j=v}$ est l'ensemble des exemples dont l'attribut considéré a_j prend la valeur v, et la notation $|X|$ indique le cardinal de l'ensemble X.

Exemple :

Le Gain du champ "Vent" du *Tableau 3.1* est calculé comme suit :

$$Gain\,(X, vent) = H(X) - \frac{9}{14}H(X_{a=oui}) - \frac{5}{14}H(X_{a=non})$$

on a :

$$H(X) = -\frac{5}{14}\log_2\frac{5}{14} - \frac{9}{14}\log_2\frac{9}{14} = 0.940$$

$$H(X_{a=non}) = -\left(\frac{6}{8}\log_2\frac{6}{8} + \frac{2}{8}\log_2\frac{2}{8}\right) = 0.811$$

Et

$$H(X_{a=oui}) = -\left(\frac{3}{6}\log_2\frac{3}{6} + \frac{3}{6}\log_2\frac{3}{6}\right) = 1.0$$

D'où :

$$Gain\,(X, vent) = 0.940 - \frac{9}{14}1.0 - \frac{5}{14}0.811 = 0.009$$

Le principe de l'algorithme ID3 pour déterminer la variable de segmentation est de prendre la variable du gain d'information maximum.

4. Construction

Par la suite, le principe général de la construction d'un arbre de décision sera présenté.

Durant la construction d'un arbre de décision, le but est de séparer au mieux les classes de manière à obtenir des feuilles (ou de manière équivalente des éléments de la partition de Ω_X correspondant à l'arbre) le plus "pures" possible en terme de classe. Cette pureté, ou homogénéité, de classe se mesure à l'aide de fonctions d'impureté notées *i*.

Différentes fonctions d'impureté existent, parmi les plus connues on peut citer l'*entropie de Shannon* (utilisée dans *C*4.5), l'*indice de Gini* (utilisé dans *CART*). Voici leurs expressions pour une feuille t_h contenant potentiellement *K* classes :

$$entropie : i(t_h) = -\sum_{k=1}^{K} \alpha_h^k \log(\alpha_h^k) \qquad (3.3)$$

$$indice\ de\ Gini : i(t_h) = \sum_{k=1}^{K} \alpha_h^k (1 - \alpha_h^k) \qquad (3.4)$$

où α_h^k est la probabilité de la classe *k* au sein de t_h. Il est à noter que le choix du critère de pureté n'a que peu d'influence sur l'efficacité de l'arbre obtenu comme souligné dans [BRE 84].

La construction d'un arbre se fait donc de manière itérative ; des *critères d'arrêt*, appelés aussi *pré-élagage* doivent donc être définis préalablement. Ces critères varient suivant l'algorithme choisi. Ils définissent des situations pour lesquelles une feuille t_h ne doit plus être coupée.

Voici les principales situations pour lesquelles un arrêt peut être envisagé [SUT 14] :

- ✓ la taille de la feuille (ou de l'un de ses potentiels enfants) est trop petite: $|t_h| < taille_{min}$
- ✓ le gain d'impureté maximale est trop petit : $\Delta i_{max} < gain_{min}$
- ✓ le nombre de feuilles maximum est atteint : $H = H_{max}$
- ✓ la profondeur maximale : $profondeur(P_H) \geq profondeur_{max}$
- ✓ la feuille est suffisamment pure : $i(t_h) < i_{min}$

Les choix des critères d'arrêt est déterminant pour la complexité des arbres obtenus ainsi que pour les temps de calcul. Selon le cadre de travail, ils pourront donc être ajustés. Des critères d'arrêt restrictifs donnant des arbres plutôt petits faciliteront la compréhension et l'interprétabilité par les différents experts impliqués mais auront un pouvoir prédictif limité par rapport à des arbres obtenus avec des critères d'arrêt plus laxistes qui seront donc plus denses mais qui correspondront à un partitionnement plus fin de Ω_X [SUT 14].

➢ Choix de la bonne taille de l'arbre

Une fois l'arbre de décision construit, il peut contenir plusieurs anomalies qui peuvent être dues au bruit ou aux valeurs extrêmes, et qui peuvent conduire au problème de sur-apprentissage (overfitting). Ce problème est la déduction d'informations plus que supporte l'ensemble de données d'apprentissage. L'arbre peut être aussi d'une taille très importante qui peut épuiser les ressources de calcul et de stockage. Pour surmonter ce problème, on effectue des opérations d'élagage qui consistent à éliminer de l'arbre les branches les moins significatives (qui déduisent d'un nombre réduit d'enregistrements ou de ceux qui appartiennent à diverses classes). L'élagage peut être effectué avant ou après l'apprentissage, on parle souvent de pré et post-élagage :

– **Pré-élagage** : effectué lors de la construction de l'arbre, lorsqu'on calcule les caractéristiques statistiques d'une partie des données tel que le gain, on peut décider de l'importance ou non de sa subdivision, et ainsi on coupe complètement des branches qui peuvent être générée.

– **Post-élagage** : effectué après la construction de l'arbre en coupant des sous arbres entiers et en les remplaçant par des feuilles représentant la classe la plus fréquente dans l'ensemble des données de cet arbre. On commence de la racine et on descend, pour chaque nœud interne (non feuille), on mesure sa complexité avant et après sa coupure (son remplacement par une feuille), si la différence est peu importante, on coupe le sous arbre et on le remplace par une feuille.

5. Apprentissage d'un arbre de décision

L'*apprentissage* consiste à apprendre un modèle à partir de données ou d'observations. Nous considérons ici le problème de *prédiction* où le modèle consiste à prédire une sortie (la prédiction) à partir d'une entrée (les données).

Lorsque cette prédiction porte sur une variable $Y \in \Omega_Y = \{\omega_1, \dots, \omega_K\}$ dite *catégorique*, *nominale* ou *qualitative*, la prédiction rentre dans le cadre général de la *classification*. Si *Y* est une variable *numérique*, il s'agit alors du cadre de la *régression*.

La classification (de même que la régression) a pour but l'attribution d'une classe à un individu à partir de ses *J attributs* $X = (X^1, \dots, X^J) \in \Omega_X = \Omega_{X^1} \times \dots \times \Omega_{X^J}$ (ou variables descriptives). Un classifieur est donc une fonction $f: \Omega_X \longrightarrow \Omega_Y$

Un classifieur se construit à partir d'un échantillon d'apprentissage qui contient *N* réalisations du couple (*X*, *Y*) et que l'on notera :

$$E = \begin{pmatrix} x_1, y_1 \\ \vdots \\ x_N, y_N \end{pmatrix} = \begin{pmatrix} x_1^1, \dots, x_1^J, y_1 \\ \vdots \\ x_N^1, \dots, x_N^J, y_N \end{pmatrix} \qquad (3.5)$$

Parmi les classifieurs les plus reconnus en classification supervisée [SUT 14] on peut citer le classifieur naïf de Bayes, les arbres de décision, les k-plus-proches-voisins (k-ppv), les réseaux de neurones, les séparateurs à vaste marge (SVM).

En général les classifieurs sont évalués sur leur aptitude à bien *classer* des individus d'un autre échantillon appelé l'*échantillon de test*. Cependant leur stabilité, leur interprétabilité, les temps de calculs qu'ils nécessitent sont autant de critères pouvant rentrer en compte dans leur évaluation suivant le problème. Il n'y a donc pas de meilleur classifieur d'une façon générale, tout dépend de ce qu'on attend d'eux suivant les objectifs de notre problème.

Arbres de décision : formalisme

Un arbre de décision est formellement une structure d'arbre comprenant des nœuds, des branches et des feuilles (ou nœuds terminaux). Un arbre a *H* feuilles sera noté $P_H = \{t_1, \dots, t_H\}$ représentent les feuilles de P_H.

Relativement à P_H, on définit la variable *feuille* $Z_{P_H} \in \Omega_{Z_{P_H}} = \{1, \dots, H\}$.

Un arbre de décision se lit de haut en bas, à chaque nœud est attaché un attribut X^j et chaque branche issue de ce nœud correspond à une sous-partie de l'espace de définition Ω_{X^J} de X^J.

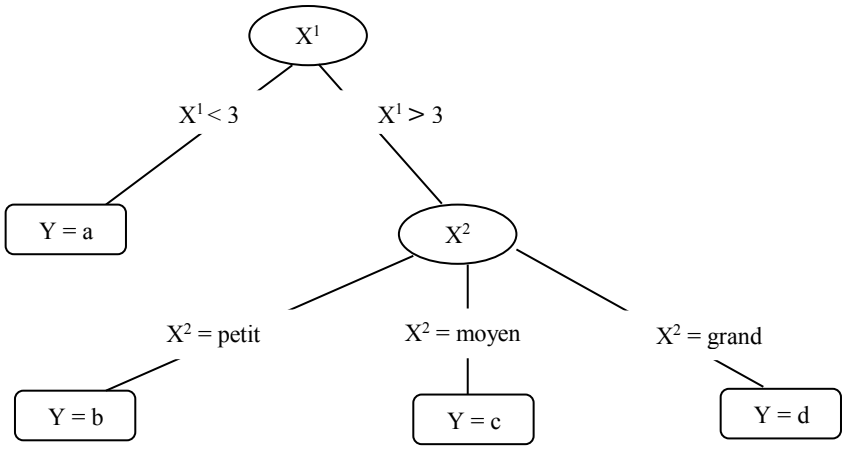


Figure 3.2 : Exemple d'arbre de décision.

On peut donc voir un arbre de décision comme une partition de l'espace des attributs Ω_X avec une classe attribuée à chaque élément de cette partition, ces éléments de la partition étant les équivalents des feuilles d'un arbre.

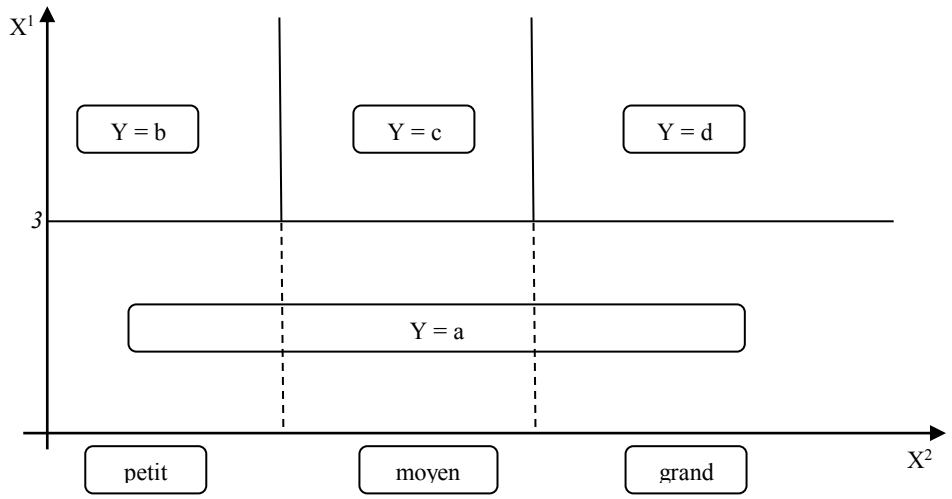


Figure 3.3 : Partition correspondante.

On notera A_h^j l'élément de partitions de Ω_{XJ} correspondant à la feuille t_h. Un arbre de décision P_H pourra donc s'écrire

$$P_H = \{t_1, \dots, t_H\} = \{A_1^1 \times \dots \times A_1^J, \dots, A_H^1 \times \dots \times A_H^J\}$$

où × est le produit cartésien.

La *figure 3.2* représente un exemple simple d'arbre de décision. Un nouvel individu dont les attributs sont $X^1 = 5$ et $X^2 =$ petit sera donc classifié $Y = b$. Par la suite on confondra les feuilles t_h avec les sous-espaces qu'elles représentent. La *figure 3.3* représente la partition de Ω_X correspondant à l'arbre de la figure 3.2.

6. Algorithmes de construction d'arbres de décision

Dans cette section, nous nous focalisons sur quelques algorithmes de construction d'arbres de décision : ID3 (Iterative Dichotomiser *3*), les arbres de classification et de régression (*CART* : Classification And Regression Trees) et les forêts aléatoires.

6.1. Algorithme ID3

ID3 (Iterative Dichotomiser 3): développé en 1986 par Ross Quinlan [QUI 86]. Il peut être appliqué seulement sur les caractéristiques nominales. Il est utilisé pour le classement.

ID3 construit l'arbre de décision récursivement. A chaque étape de la récursion, il calcule parmi les attributs restant pour la branche en cours, celui qui maximisera le gain d'information. C'est-à-dire l'attribut qui permettra le plus facilement de classer les exemples à ce niveau de cette branche de l'arbre. Le calcul ce fait à base de l'entropie de Shannon [SHAN 48]. L'algorithme suppose que tous les attributs sont catégoriels ; si des attributs sont numériques, ils doivent être discrétisés pour pouvoir l'appliquer. ID3 utilise l'*algorithme Construire-arbre* précédent (***section 3*** : *Un exemple introductif*).

Problèmes de l'algorithme ID3

- Solution globale non garantie (la solution trouvée est un optimum local ; une amélioration possible : backtracking pour tenter d'obtenir une meilleure solution).
- Sur-apprentissage (overfitting) : pour l'éviter il faut préférer les arbres de taille réduite (problème d'élagage de l'arbre).
- Pas adapté pour des données numériques continues (ces données doivent être quantifiées avant d'être utilisée avec cet algorithme, mais comment faire cette quantification de façon optimale ?).

Pour essayer de pallier à ces insuffisances deux extensions ont été proposées, l'algorithme C4.5 [QUI 14] et l'algorithme C5.0.

6.2. Algorithme CART

CART introduite par Breiman et al. (1984) [BRE 84] est basée sur un partitionnement récursif de l'espace des données. C'est une amélioration de l'algorithme ID3, il prend en compte les attributs numérique ainsi que les valeurs manquantes. C'est une méthode non-paramétrique efficace, simple à implémenter et utilisable à la fois en régression et en

classification. Le principe général de *CART* est de construire une règle de prédiction au moyen d'un partitionnement récursif et binaire de l'espace des données. La partition ainsi obtenue peut être représentée sous la forme d'un arbre binaire facilement interprétable. La *Figure 3.4* illustre la correspondance entre une partition dyadique et un arbre binaire.

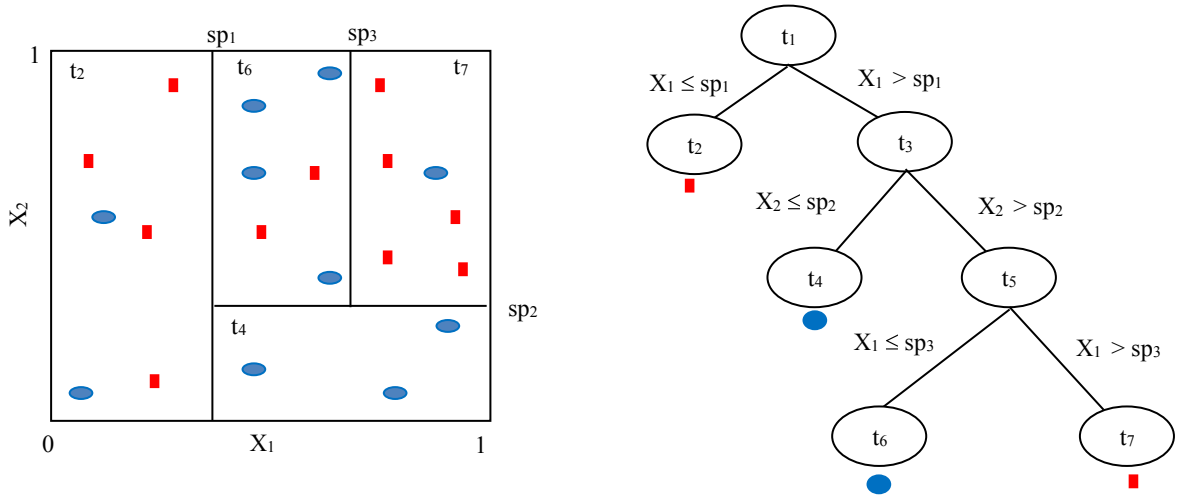


Figure 3.4 : Un exemple d'arbre CART en classification binaire. A chaque feuille est associée la classe la mieux représentée.

Construction d'un arbre CART

L'algorithme CART construit un arbre de décision d'une manière analogue à l'algorithme ID3. Contrairement à ce dernier, l'arbre de décision généré par CART est binaire et le critère de segmentation est l'indice de Gini.

Pour construire un arbre CART à partir des données de l'échantillon D_n d'apprentissage, l'algorithme procède en deux étapes [POT 18] :

• Étape 1 : Élaboration d'un arbre maximal.

Cette étape consiste en un partitionnement récursif et dyadique de l'espace des données X. Au départ, l'espace X tout entier est associé à la racine de l'arbre, que l'on note t_1. L'algorithme commence par diviser la racine t_1 en deux sous-espaces disjoints t_{1L} et t_{1R} (appelés nœuds fils) comme suit :

$$t_{1L} = \{X_i, i \leq n : X_{ij} \leq sp\} \quad and \quad t_{1R} = \{X_i, i \leq n : X_{ij} > sp\},$$

où $j = 1, \ldots, p$ et $sp \in \mathcal{R}$. Une division δ est donc définie par un couple $\delta = (j, sp)$ où j désigne l'indice de la variable de coupure et sp désigne une valeur seuil pour cette variable. Le choix

de ce couple repose sur la définition d'une fonction Q d'impureté. La méthode sélectionne la coupure $\delta_{t_1}^*$ qui maximise la décroissance d'impureté définie par :

$$\Delta Q(t_1, \delta) = n_{t_1} Q(t_1) - n_{t_{1R}} Q(t_{1R}) - n_{t_{1L}} Q(t_{1L}) \qquad (3.6)$$

où t_{1L} et t_{1R} désignent les deux nœuds fils de t_1 définis par la coupure δ et n_{t1} (respectivement $n_{t_{1L}}$ et $n_{t_{1R}}$) désigne le nombre d'observations dans la racine t_1 (respectivement dans les nœuds fils t_{1L} et t_{1R}). En régression, la fonction d'impureté $Q(t)$ correspond le plus souvent à la variance du nœud t :

$$Q(t) = \frac{1}{n_t} \sum_{i:X_i \in t} (Y_i - \bar{Y}_t)^2 \qquad (3.7)$$

où $\bar{Y}_t$ est la moyenne des Y_i des observations contenues dans le nœud t. En classification, l'*indice de Gini* est généralement utilisé pour définir l'impureté d'un nœud t :

$$Q(t) = \sum_{k=1}^{K} \pi_k(t) \left(1 - \pi_k(t)\right) \qquad (3.8)$$

où $\pi_k(t) = \frac{1}{n_t} \sum_{i:X_i \in t} 1_{Y_i = k}$ est la proportion d'observations de la classe k dans le nœud t. Dans les deux cas, l'objectif est de partager les observations de l'échantillon D_n en deux groupes disjoints les plus homogènes possible au sens de la variable réponse Y.

Une fois la racine de l'arbre découpée, la procédure est répétée sur chaque nœud fils, puis de manière récursive sur tous les autres nœuds jusqu'à ce que chaque nœud soit homogène, c'est-à-dire que toutes les observations contenues dans le nœud partagent la même valeur pour Y. Les nœuds terminaux, qui ne sont pas découpés sont appelés feuilles. A la fin du découpage, les feuilles forment une partition fine de l'espace des données X, qui peut être représentée sous la forme d'un arbre maximal, noté T_{max}. Une prédiction $\hat{y}_t$ est associée à chaque feuille t de l'arbre T_{max} (la moyenne empirique de la réponse Y dans le nœud t en régression ou en classification, la classe de Y la mieux représentée dans le nœud t). De l'arbre T_{max}, on déduit alors la règle de prédiction notée $\hat{f}_{T_{max}}$ et définie, pour toute observation $x \in X$, par :

$$\hat{f}_{T_{max}}(x) = \sum_{t \in \tilde{T}_{max}} \hat{y}_t 1_t(x) \qquad (3.9)$$

où $\tilde{T}_{max}$ désigne l'ensemble des nœuds terminaux de T_{max} et $1_t(x)$ désigne la fonction indicatrice égale à *1 si* $x \in t$ *et 0* sinon (voir *figure 3.4*).

Étape 2 : Élagage et sélection de l'arbre final.

L'arbre maximal T_{max} souvent trop complexe n'est généralement pas optimal au sens d'un critère de performance choisi (par exemple en classification, l'erreur de classification). Un nombre excessif de coupures conduit à un arbre qui a tendance à sur-ajuster. Pour éviter cela, T_{max} est élagué suivant la méthode *minimal cost-complexity pruning* introduite par (Breiman et al., 1984) [BRE 84].

Ce procédé consiste à extraire une suite de sous-arbres de T_{max} par minimisation du critère pénalisé défini pour tout sous-arbre T de T_{max}, noté $T \preceq T_{max}$, et pour tout $\alpha \in \mathbb{R}^+$ par :

$$R_\alpha(T) = R(T, D_n) + \alpha\left|\tilde{T}\right|, \qquad (3.10)$$

où $\left|\tilde{T}\right|$ désigne le nombre de feuilles de l'arbre T et $R(T, D_n)$ correspond à l'erreur empirique du modèle T estimée à partir des données de l'échantillon D_n. En régression, $R(T, D_n)$ désigne le critère des moindres carrés

$$R(T, D_n) = \frac{1}{n} \sum_{i:(X_i,Y_i)\in D_n} (Y_i - \hat{f}_T(X_i))^2 \qquad (3.11)$$

et en classification, $R(T, D_n)$ désigne l'erreur de classification

$$R(T, D_n) = \frac{1}{n} \sum_{i:(X_i,Y_i)\in D_n} 1_{Y_i \neq \hat{f}_T(X_i)} \qquad (3.12)$$

Dans l'équation (3.10), α est un paramètre à régler/à choisir. Il permet de contrôler la complexité de l'arbre. Plus α est grand, plus les arbres ayant beaucoup de feuilles sont pénalisés.

La méthode d'élagage consiste à trouver pour toute valeur $\alpha \in R$, le plus petit sous-arbre de T_{max} optimal au sens du critère pénalisé (3.10). Une recherche exhaustive de chaque arbre optimal se révèle souvent trop coûteuse. Aussi, Breiman et al. (1984) [BRE 84] propose une stratégie efficace, qui repose sur le résultat suivant.

Théorème : (Breiman et al., 1984) [BRE 84]. Pour tout arbre maximal T_{max}, il existe une suite finie et strictement croissante de paramètres

$$0 = \alpha_1 < \cdots < \alpha_K$$

associée à une suite de sous-arbres emboités $T_{max} \geq T_1 > \cdots > T_K = \{t_1\}$ tous élagués de T_{max} et vérifiant pour tout $1 \leq k < K$,

$$pour\ tout\ \alpha \in [\alpha_k; \alpha_{k+1}[,\ T_k = \underset{T \leq T_{max}}{\operatorname{argmin}} R_\alpha(T)\ ,$$

et

$$pour\ tout\ \alpha \geq\ \alpha_k,\ T_k = \underset{T \leq T_{max}}{\operatorname{argmin}} R_\alpha(T)\ .$$

Ainsi, l'extraction de la suite d'arbres optimaux repose sur un nombre fini de valeurs pour α et chaque arbre de la suite est obtenu par élagage du précédent. En d'autres termes, pour chaque $k = 1, \ldots, K$, T_k est le plus petit sous arbre de T_{k-1} minimisant R_{α_k} (en posant ici $T_0 = T_{max}$). De plus, la suite $\{T_k\}_{1 \leq k \leq K}$ contient toute l'information puisque pour tout $\alpha \geq 0$, le plus petit sous-arbre optimal au sens de R_α est contenu dans la suite.

L'arbre final est le meilleur sous-arbre de la suite $\{T_k\}_{1 \leq k \leq K}$ au sens d'un critère donné et évalué sur un échantillon témoin ou par validation croisée. Des garanties théoriques justifiant la stratégie d'élagage et la sélection de l'arbre final ont été obtenues en régression [GEY 05] et en classification [GEY 12].

6.3. Forêts aléatoires

Les méthodes de partitionnement et particulièrement la méthode *CART* connaissent un succès important. Elles sont maintenant couramment utilisées dans de nombreux domaines notamment dans le domaine médical [SAT 11] ou en écologie [PES 11]. Cependant, ces méthodes s'avèrent être très instables. En effet, une simple perturbation de quelques observations dans l'échantillon d'apprentissage peut modifier complètement l'arbre ainsi construit. Et Si le coté interprétable des arbres de décision n'est pas important et que l'efficacité prédictive est le seul objectif, de nombreuses méthodes faisant intervenir des arbres de décision ont été développées dans le but d'améliorer cette efficacité prédictive. La plus connue est sans doute les *forêts aléatoires* proposées par Breiman en 2001 [BRE 01].

Les forêts aléatoires permettent de résoudre cette faiblesse des arbres de décision et en améliorent les performances prédictives.

Cette méthodologie consiste en l'apprentissage d'un certain nombre d'arbres de décision sur des sous-échantillons de l'échantillon d'apprentissage (obtenus par *bootstrap*). La prédiction se fait ensuite en combinant les prédictions de chaque arbre (par *vote* par exemple). Les conditions associées aux nœuds peuvent concerner plusieurs attributs (tirés aléatoirement) combinés éventuellement de façon non linéaire.

L'efficacité prédictive des *forêts aléatoires* et leurs propriétés mathématiques [BIA 08] en font des classifieurs très puissants et donc très populaires, notamment en *apprentissage automatique*. La méthodologie des forêts aléatoires tire avantage d'incertitudes aléatoires injectées à différents stades de l'apprentissage (ré-échantillonage aléatoire, ensembles aléatoires d'attributs potentiellement associés aux nœuds).

Algorithme des forêts aléatoires

Les forêts aléatoires sont basées sur le *bagging* [BRE 96], approche qui consiste à agréger une collection d'estimateurs construits à partir d'échantillons *bootstrap* [FRE 96] : Le Boosting est une méthode séquentielle, chaque échantillon étant tiré en fonction des performances de la règle de base appliquée sur l'échantillon précédent. Une forêt aléatoire est une agrégation d'arbres aléatoires. Le principe de construction d'une forêt est tout d'abord de générer indépendamment un grand nombre (noté *ntree*) d'échantillons *bootstrap* $D_n^1, \dots, D_n^{ntree}$ en tirant aléatoirement, pour chacun d'eux, a_n observations (avec ou sans remise) dans l'échantillon D_n d'apprentissage. Ensuite, *ntree* arbres de décision $T^l, \dots, T^{ntree}$ sont construits à partir des échantillons *bootstrap* $D_n^1, \dots, D_n^{ntree}$ et en utilisant une variante de *CART*. En effet chaque arbre est ici construit de la façon suivante. Pour découper un nœud, l'algorithme choisi aléatoirement et sans remise un nombre *mtry* de variables explicatives, puis il détermine la meilleure coupure uniquement suivant les *mtry* variables sélectionnées. De plus, l'arbre construit est pleinement développé et n'est pas élagué. La forêt aléatoire, que l'on note $\{T^b\}_1^{ntree}$, est enfin obtenue en agrégeant les *ntree* arbres ainsi construits. Elle définit une règle de prédiction qui correspond à la moyenne empirique des prédictions en régression et au vote majoritaire en classification. La construction des forêts aléatoires de Breiman [BRE 96] est décrite par l'Algorithme *forêts aléatoires*.

Algorithme Forêts aléatoires.

Input : Échantillon d'apprentissage D_n, $ntree \in \mathrm{N}$, $a_n \in \{1, \dots, n\}$, $mtry \in \{1, \dots, d\}$, $nodesize \in \{1, \dots, n\}$

For b = 1 **to** *ntree* **do**

1. Construction de l'échantillon *bootstrap* D_n^b : tirer uniformément et avec remise a_n observations dans D_n^b.
2. Construction de l'arbre T^b à partir de D_n^b : répéter de manière récursive le procédé suivant sur chaque nœud, jusqu'à ce que chaque nœud soit homogène ou contienne moins de *nodesize* observations :

   (a) Tirer un sous-ensemble $Mmtry \subset \{X_1, \dots, X_d\}$ de cardinal *mtry* uniformément et sans remise.

   (b) Choisir la meilleure coupure au sens du critère de coupure de CART (voir équation 3.3) et en se basant uniquement sur le sous-ensemble *Mmtry* de variables.

   (c) Diviser le nœud en deux nœuds fils selon la coupure précédemment choisie.
3. Définition de la règle de prédiction $\hat{f}_b$ à partir de l'arbre maximal T^b.

Output: la collection d'arbres $T^1, \dots, T^{ntree}$ et la collection associée de règles de prédiction $\hat{f}_1, \dots, \hat{f}_{ntree}$.

Prédiction de la forêt aléatoire en $x \in X$:

En régression : $\hat{f}_{rf}(x) = \frac{1}{ntree} \sum_{b=1}^{ntree} \hat{f}_b(x)$,

En classification : $\hat{f}_{rf}(x) = \underset{k=1,\dots,K}{\operatorname{argmax}} \left(\sum_{b=1}^{ntree} 1_{\hat{f}_b(x)=k} \right)$

L'algorithme des forêts aléatoires comporte plusieurs paramètres [POT 18] :

- ✓ Le nombre d'arbres *ntree* de la forêt. Sa valeur par défaut est 500. Notons que ce paramètre n'est pas vraiment un paramètre à calibrer dans le sens où une plus grande valeur de ce paramètre mènera toujours à des prédictions plus stables qu'une plus petite valeur de ce paramètre.
- ✓ Le nombre *mtry* de variables choisies pour le découpage de chaque nœud. Sa valeur par défaut est *mtry = d/3* en régression et $mtry = \sqrt{d}$ en classification. C'est sans doute le paramètre le plus important à calibrer puisqu'il peut grandement influencer les performances de la forêt.
- ✓ Le nombre minimum d'observations *nodesize* en dessous duquel un nœud n'est plus découpé. La valeur par défaut de ce paramètre est *nodesize = 1* en classification et *nodesize = 5* en régression. En général, ce paramètre est laissé à sa valeur par défaut.
- ✓ Le nombre d'observations a_n dans chaque échantillon *bootstrap*. Par défaut, chaque échantillon *bootstrap* contient $a_n = n$ observations tirées avec remise dans l'échantillon initial D_n.

Plusieurs auteurs se sont intéressés au choix et à l'influence de ces paramètres [BIA 16, BRE 01, GEN 10]. En général, les valeurs par défaut des paramètres donnent de bons résultats.

Les forêts aléatoires connaissent aujourd'hui un large succès. La méthode a permis de résoudre efficacement un grand nombre de problèmes dans des domaines variés comme par exemple en écologie [PRA 06], en bioinformatique [DIA 06], ou encore en analyse d'image [SHO 11]. Outre ses très bonnes performances et sa large applicabilité, la méthode ne dépend que d'un petit nombre de paramètres ce qui la rend aussi facilement utilisable.

7. Les Avantages et les inconvénients

L'induction par arbres de décision est une technique arrivée à maturité ; ses caractéristiques, ses points forts et ses points faibles sont maintenant bien connus ; il est possible de la situer précisément sur l'échiquier des très nombreuses méthodes d'apprentissage [HAST 09, RAK 05].

Les arbres présentent des performances comparables aux autres méthodes supervisées; les nombreuses comparaisons empiriques l'ont suffisamment montré [LIM 00]. La méthode

est non paramétrique ; elle ne postule aucune hypothèse a priori sur la distribution des données ; elle est résistante aux données atypiques ; le modèle de prédiction est non linéaire. Lorsque la base d'apprentissage est de taille importante, elle présente des propriétés similaires aux algorithmes des plus proches voisins [BRE 84].

Il faut néanmoins tempérer ce constat. Le premier reproche qu'on peut lui adresser est son incapacité, avec les algorithmes classiques (C4.5, CART, CHAID,…), à détecter les combinaisons de variables ; ceci est dû au principe de construction pas à pas de l'arbre, entraînant une certaine « myopie ». Le second reproche est dans la nécessité de disposer d'un échantillon d'apprentissage de grande taille. L'arbre certes peut reproduire approximativement toutes formes de frontières, mais au prix d'une fragmentation rapide des données, avec le danger de produire des feuilles avec très peu d'individus. Corollaire à cela, les arbres sont en général instables ; les bornes de discrétisation notamment dans les parties basses de l'arbre sont entachées d'une forte variabilité. Ainsi, certains chercheurs préconisent de procéder à la discrétisation préalable des variables avant la construction de l'arbre [DOU 95].

L'induction par arbre de décision est capable de traiter de manière indifférenciée les données continues et discrètes. Elle dispose de plus d'un mécanisme naturel de sélection de variables. Elle doit être privilégiée lorsque l'on travaille dans des domaines où le nombre de descripteurs est élevé, dont certains, en grand nombre, sont non-pertinents. Nous devons également relativiser cette affirmation. En effet, non sans surprise, des travaux dans le domaine de la sélection de variables ont montré que la réduction préalable des descripteurs dans des domaines fortement bruités améliorait considérablement les performances des arbres de décision [YU 03]. Il y a principalement deux causes à cela : à force de multiplier les tests, l'algorithme multiplie également le risque d'introduire des variables non-significatives dans l'arbre. Ce risque est d'autant plus élevé que les méthodes comme C4.5, très utilisées dans la communauté de l'apprentissage automatique, adoptent la construction « hurdling » (introduire une variable même si elle induit un gain nul) en misant, parfois à tort, sur le post-élagage pour éliminer les branches non-pertinentes de l'arbre.

Enfin, dernier point de différenciation, qui assure en grande partie la popularité des arbres auprès des praticiens : leur capacité à produire une connaissance simple et directement utilisable, à la portée des non-initiés. Un arbre de décision peut être lu et interprété directement ; il est possible de le traduire en base de règles sans perte d'information. A la fin des années 80, on considérait que cette méthode assurait le renouveau des systèmes experts en

éliminant le goulot d'étranglement que constitue le recueil des règles [KON 93]. Cette qualité est renforcée par la possibilité qu'a l'expert d'intervenir directement dans le processus de création du modèle de prédiction.

L'appropriation de l'outil par les experts du domaine assure dans le même temps une meilleure interprétation et compréhensibilité des résultats.

8. Conclusion

Les arbres de décision répondent simplement à un problème de discrimination, c'est une des rares méthodes que l'on peut présenter assez rapidement à un public non spécialiste du traitement des données sans se perdre dans des formulations mathématiques délicates à appréhender.

Dans ce chapitre nous avons présenté un bref historique d'arbres de décision. Nous avons arboré un exemple pour mieux saisir les définitions des arbres de décision. Nous avons détaillé l'idée de la construction et d'apprentissage d'un arbre de décision. Nous avons exposé l'algorithme ID3, la méthode CART et les forêts aléatoires. Enfin, nous avons cité quelques avantages et quelques inconvénients des arbres de décision.

Dans le chapitre suivant, nous présenterons en détail nos contributions.

Chapitre 4

Contributions

1. Introduction

Nous avons apporté deux contributions basées sur la recherche d'images. Notre première contribution consiste en une combinaison de l'indexation et recherche textuelle d'images avec l'indexation et recherche visuelle d'images. L'objectif de cette combinaison est d'améliorer le résultat de recherche. Pour notre deuxième contribution, il s'agit de la proposition d'une nouvelle plateforme dans le Cloud. Le but principal de cette plateforme est bien de réduire le maximum le temps de réponse pour une requête donnée.

Dans les chapitres précédents, nous avons détaillé les principales étapes de conception d'un système de recherche par contenu, les définitions de bases du Cloud computing, ainsi que les concepts fondamentaux des arbres de décision.

Il nous reste donc à présenter l'architecture générale de nos systèmes en détaillant le déroulement des différentes étapes d'indexation et de recherche d'images pour chaque contribution.

2. Contribution 1 : Indexation et recherche visuo-textuelle des bases de données images

2.1. Architecture générale

Comme tout système de recherche d'images par contenu, notre système se compose de deux phases : phase hors ligne de structuration des données et une phase en ligne d'interrogation de la base. La *figure 4.1* présente les différentes étapes par lesquelles nous sommes passés pour la réalisation de notre système.

2.2. Contexte des expérimentations

Nous avons développé notre système de recherche '*INDEX*' en C++, sous le système d'exploitation ***Windows 7***.

Nous avons utilisé SGBD relationnel MySQL pour la création des différentes bases de données. C'est un SGBD de données gratuit qui offre de nombreuses fonctionnalités pour la gestion de bases de données relationnelles.

➢ Bases d'images

Le type de base d'images utilisées

Dans une base de données, il existe une multitude de types d'images (IRM, microscopie médicale, dessin,…). Pour limiter notre étude, nous ne considérons dans cette contribution que les bases d'images généralistes qui contiennent des images de nature variée. Nous nous intéresserons principalement aux images fixes.

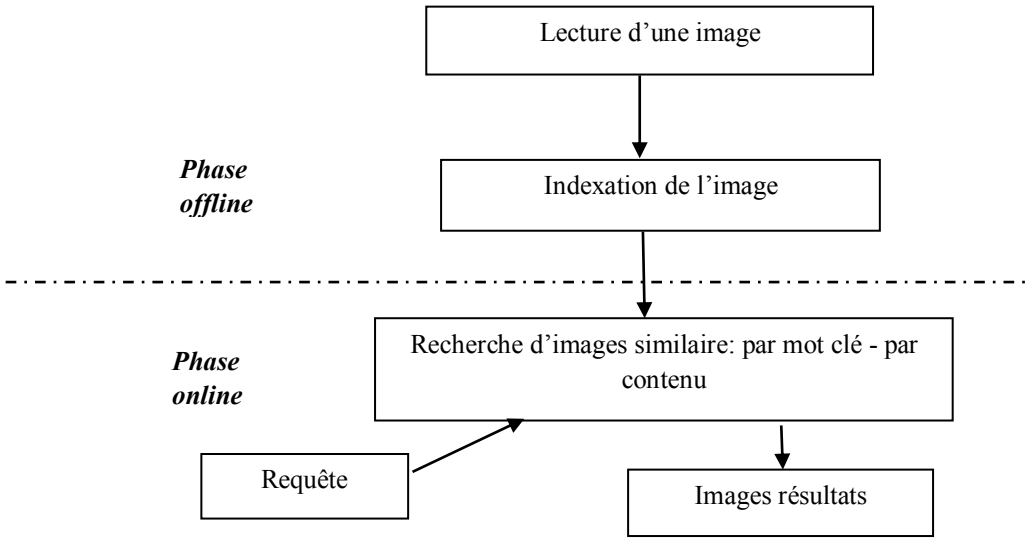


Figure 4.1 : Architecture générale de « INDEX ».

2.3. Phase hors ligne

La phase hors ligne de notre système englobe les phases suivantes:

2.3.1. Indexation textuelle manuelle

Nous avons donné pour chaque image de notre base, un ou plusieurs mots clés qui interprètent l'image.

Nous avons utilisé six (O6) catégories qui sont :

- Animal
- Plante
- Maison

- Transport
- Nature
- Insecte

Cette phase d'indexation est lente, puisque c'est une phase manuelle. Elle demande beaucoup de concentration, et une bonne connaissance de la langue.

2.3.2. Indexation par le contenu

- **Histogrammes de couleur**

La couleur est l'une des composantes les plus utilisées en indexation d'image. Elle est indépendante de la taille de l'image et de son orientation. Dans notre système, nous utilisons l'espace couleur : RVB. Nous avons choisi cet espace parce qu'il est couramment utilisé. C'est un système des trois couleurs fondamentales. Il associe à chaque couleur trois composantes (ou canaux), qui correspondent aux intensités respectives de trois couleurs primaires de la synthèse additive.

Les histogrammes sont résistants à un certain nombre de transformations sur l'image. Ils sont invariants aux rotations, aux translations, et aux changements d'échelle. Ils sont cependant sensibles aux changements d'illumination et aux conditions d'éclairage.

2.4. Phase en ligne

Comme nous l'avons expliqué dans le premier chapitre, le but des systèmes de recherche d'image est de permettre à un utilisateur de trouver, dans des bases d'images, toutes celles qui sont semblables à une image qui l'intéresse.

La recherche par similarité n'est pas de mesurer sur les images directement mais plutôt sur la base des vecteurs caractéristiques (signatures d'images).

Étant donnée une requête, un système de recherche d'images doit être capable, sur la base des index dont il dispose, de retrouver les images les plus similaires à une image requête en terme d'une distance donnée.

Nous avons proposé trois choix de recherche :

2.4.1. Recherche par mot clé

La recherche dans cette phase est facile, puisque tout simplement c'est une recherche par mot clé. La *figure 4.2* représente le résultat de la recherche obtenue avec le mot clé voiture.

Figure 4.2 : Un exemple d'une recherche effectuée par notre système (résultat de la recherche par mot clé = voiture) (a, b, c, d : les différentes fenêtres d'affichages)

2.4.2. Recherche par image exemple

Dans cette étape, nous avons utilisé le principe d'un CBIR.

2.4.2.1. Mesure de similarité

Nous avons utilisé la technique de corrélation d'histogrammes. Avec cette méthode, on calcule une certaine mesure de similarité pour mesurer si deux histogrammes sont « proches » l'un de l'autre.

Soit h_1 et h_2 deux histogrammes de même taille N, la formule utilisée pour la corrélation d'histogramme est la suivante :

$$d(h_1, h_2) = \frac{\sum_{i=1}^{N} \bar{h}_1(i)\bar{h}_2(i)}{\sqrt{\sum_{i=1}^{N} \bar{h}_1(i)^2 \sum_{i=1}^{N} \bar{h}_2(i)^2}} \qquad (4.1)$$

$$\text{où} \qquad \bar{h}(i) = h(i) - \frac{1}{N}\sum_{i=1}^{N} h(i) \qquad (4.2)$$

2.4.2.2. Algorithme utilisé

L'algorithme utilisé pour la recherche des images similaires est le suivant :

Etape 1 : Lecture de l'image requête.

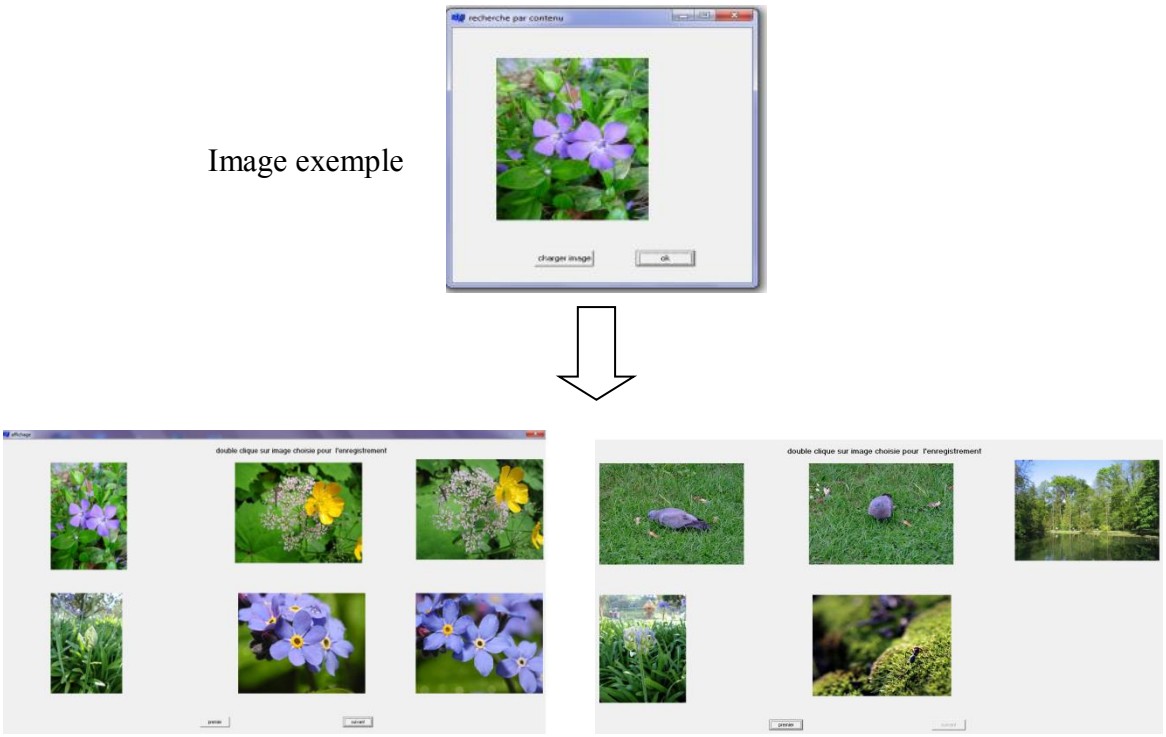
Etape 2 : Calcul de l'histogramme de trois couleurs R,V,B de l'image requête.

Etape 3 : Calcul de la distance par une fonction de similarité avec l'image requête.

Etape 4 : Tri des distances pour chaque image calculée.

Etape 5 : Affichage des N premiers résultats.

Image exemple



(a) (b)

Figure 4.3 : Exemple1 d'une recherche effectuée par notre système (résultat de la recherche par contenu) (a, b: les différentes fenêtres d'affichages)

(a) (b)

Figure 4.4 : Exemple 2 d'une recherche effectuée par notre système (résultat de la recherche par contenu) (a, b: les différentes fenêtres d'affichages)

Les deux *figures 4.3 et 4.4* présentent les résultats d'une recherche effectuée par notre système (résultat de la recherche par contenu). Les résultats correspondent aux images les plus « proches » de l'image proposée en exemple. Les caractéristiques extraites des images sont relatives à la couleur.

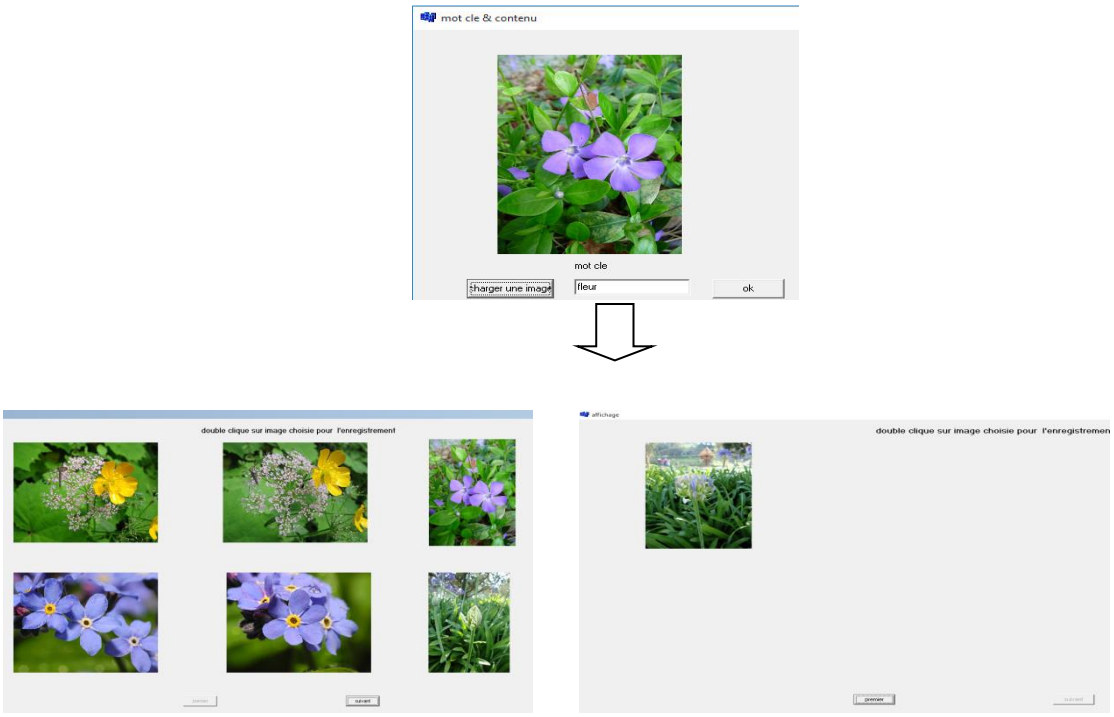
On remarque que les résultats obtenus de la recherche correspondent partiellement à des images visuellement similaires à la requête. Ce phénomène est connu sous le nom de **fossé/vide sémantique (*semantic gap*)**. Pour résoudre ce problème, nous avons combiné entre l'image et le texte.

2.4.3. Recherche par mot clé et par contenu (visuo-textuelle)

L'indexation par mot clé et par contenu est une combinaison des deux méthodes de recherche citées précédemment. Elle consiste à procéder à une recherche par mot clé et une recherche par contenu de façon séparée, puis à calculer l'intersection des deux ensembles

d'images obtenus par les deux méthodes. On obtient ainsi un nouvel ensemble d'images. Les *figures 4.5 et 4.6* représentent le résultat de la recherche visuo-textuelle.

En combinant l'image et le texte, nous avons comblé le vide sémantique.



(a) (b)

Figure 4.5 : Exemple 1 d'une recherche effectuée par notre logiciel (résultat de la recherche visuo-textuelle) (a, b: les différentes fenêtres d'affichages)

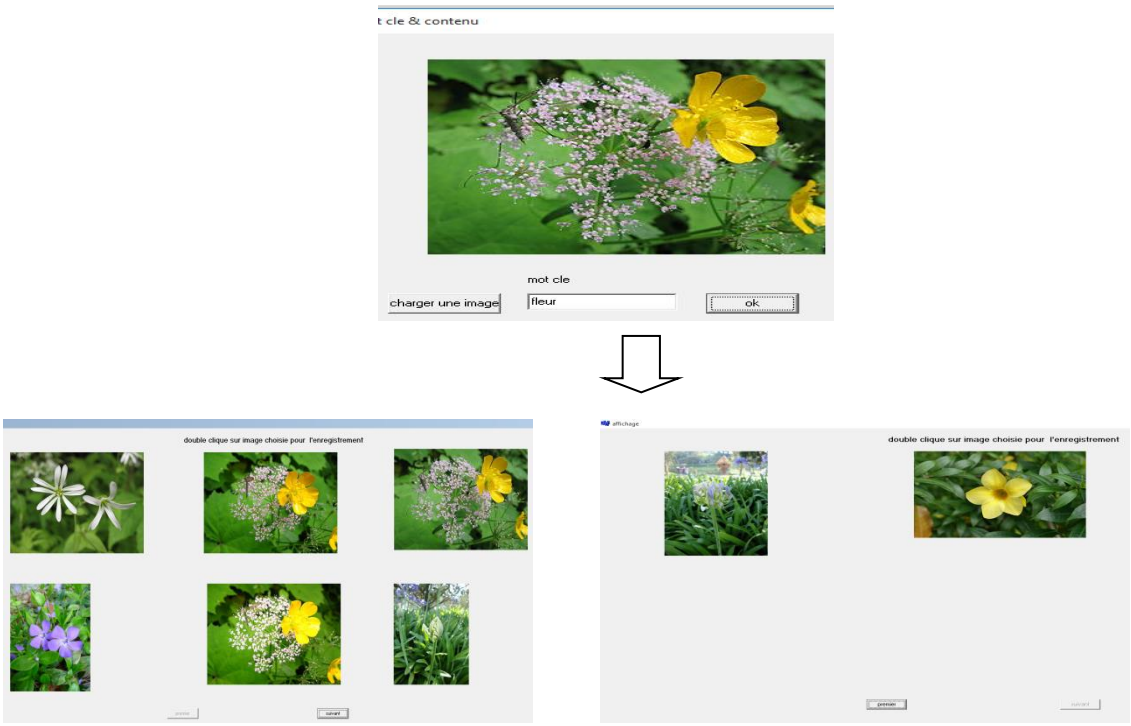


Figure 4.6 : Exemple 2 d'une recherche effectuée par notre logiciel (résultat de la recherche visuo-textuelle) (a, b: les différentes fenêtres d'affichages)

3. Contribution 2 : Modèle de plateforme en tant que service (PaaS) efficace sur le Cloud public pour le système CBIR

La recherche d'images par contenu (CBIR) est un domaine en pleine croissance avec l'avancement de nouvelles capacités technologiques. Les chercheurs s'intéressent aux méthodes de recherche par contenu qui permettraient de trouver des objets dans des bases de données en utilisant uniquement le contenu numérique d'images. Il s'agit d'un domaine de recherche très actif, notamment dans le domaine médical [QUE 08].

Actuellement, la gestion d'une base de données d'images médicales est une tâche très difficile. Un système de recherche et de récupération automatique est nécessaire pour gérer les bases de données d'images médicales [KUM 16]. La gestion et l'indexation de ces grandes bases de données d'images deviennent de plus en plus complexes [DMI 09].

Les systèmes CBIR ont changé la façon dont les pathologistes diagnostiquent un patient, facilitant le diagnostic assisté par ordinateur en récupérant des images similaires à

une image de requête. Les images récupérées sont classées selon leur pertinence [RAM 13].

Le volume croissant d'images médicales conduit à la nécessité de concevoir des systèmes précis pour un stockage et une indexation efficaces de ces images. L'amélioration des techniques d'analyse et de récupération des images médicales est l'un des plus grands défis. La production annuelle d'images des grands centres de radiologie est volumineuse. La mise en œuvre de techniques informatiques pour une gestion d'indexation efficace, un traitement automatisé et des recherches pertinentes nécessite un développement efficace, rapide et précis d'outils d'aide à la décision. Le déploiement de tels outils doit être effectué avec soin en raison des caractéristiques particulières du domaine médical [TRO 09].

Le but principal de cette contribution est de minimiser le temps de réponse pour une requête donnée et de combler le vide/fossé sémantique.

Nous nous sommes concentrés sur l'hybridation texte/image, c'est-à-dire que la description de l'image est extraite de son contenu en même temps que d'autres sources d'informations externes. L'indexation textuelle n'est pas directement concurrente mais plutôt complémentaire de l'indexation basée-contenu. En effet, l'image seule ne permet pas de répondre à des requêtes abstraites comme par exemple la recherche d'images traduisant le concept de « incertitude » alors que cela semble possible par le texte. La coopération de l'information image et de l'information textuelle n'est cependant pas évidente et immédiate à mettre en œuvre.

3.1. Le modèle PaaS-CBIR proposé

Cette section présente le fonctionnement du PaaS-CBIR.

Le PaaS-CBIR est proposé comme un outil qui peut aider les spécialistes à établir un diagnostic précis et en temps opportun. Le PaaS-CBIR est une nouvelle plateforme en tant que service (PaaS) qui fournit des logiciels en tant que service avec une latence minimale lors de la phase en ligne. Pendant la phase hors ligne, le PaaS-CBIR cherche à organiser la base de données à l'aide d'un arbre de décision afin d'assurer le parallélisme et la collaboration.

L'Organisation mondiale de la santé (OMS) estime que le fardeau mondial du cancer a atteint 18,1 millions de nouveaux cas et 9,6 millions de décès en 2018 [25]. Un homme sur

cinq et une femme sur six dans le monde développeront un cancer au cours de leur vie, et un homme sur huit et une femme sur 11 mourront de cette maladie. À l'échelle mondiale, le nombre total de personnes vivant avec un cancer dans les cinq ans suivant le diagnostic, connu sous le nom de prévalence sur cinq ans, est estimé à 43,8 millions.

L'OMS [25] a également cité le cancer du poumon (Lung) comme le cancer le plus souvent diagnostiqué chez les hommes (14,5% du total pour les hommes et 8,4% pour les femmes) et la principale cause de décès par cancer chez les hommes (22,0%, soit environ un décès sur cinq). Chez l'homme, le cancer du poumon est suivi du cancer de la prostate (incidence 13,5%) et du cancer colorectal (incidence 10,9%) et, pour la mortalité, du cancer du foie (Liver) (10,2%) et du cancer de l'estomac (Stomach) (9,5%). Le cancer du sein (Breast) est le cancer le plus souvent diagnostiqué chez les femmes (24,2%, soit environ un sur quatre des nouveaux cas de cancer diagnostiqués chez les femmes dans le monde). Le cancer du sein est le cancer le plus souvent diagnostiqué dans 154 des 185 pays couverts par GLOBOCAN 2018. Le cancer du sein est également la principale cause de décès par cancer chez les femmes (15,0%), suivi du cancer du poumon (13,8%) et du cancer colorectal (9,5%), qui sont également le troisième et deuxième types de cancer les plus courants, respectivement. Le cancer du col utérin (Cervical) se classe au quatrième rang pour l'incidence (6,6%) et la mortalité (7,5%).

Sur la base des statistiques expliquées ci-dessus, nous avons construit notre plateforme en tant que service pour organiser efficacement notre base de données.

3.2. Architecture PaaS-CBIR

Sur la base des statistiques de l'organisation mondiale de la santé discutées ci-dessus [25], nous avons réparti les serveurs Cloud en fonction des types de cancers (voir *figure 4.7*).

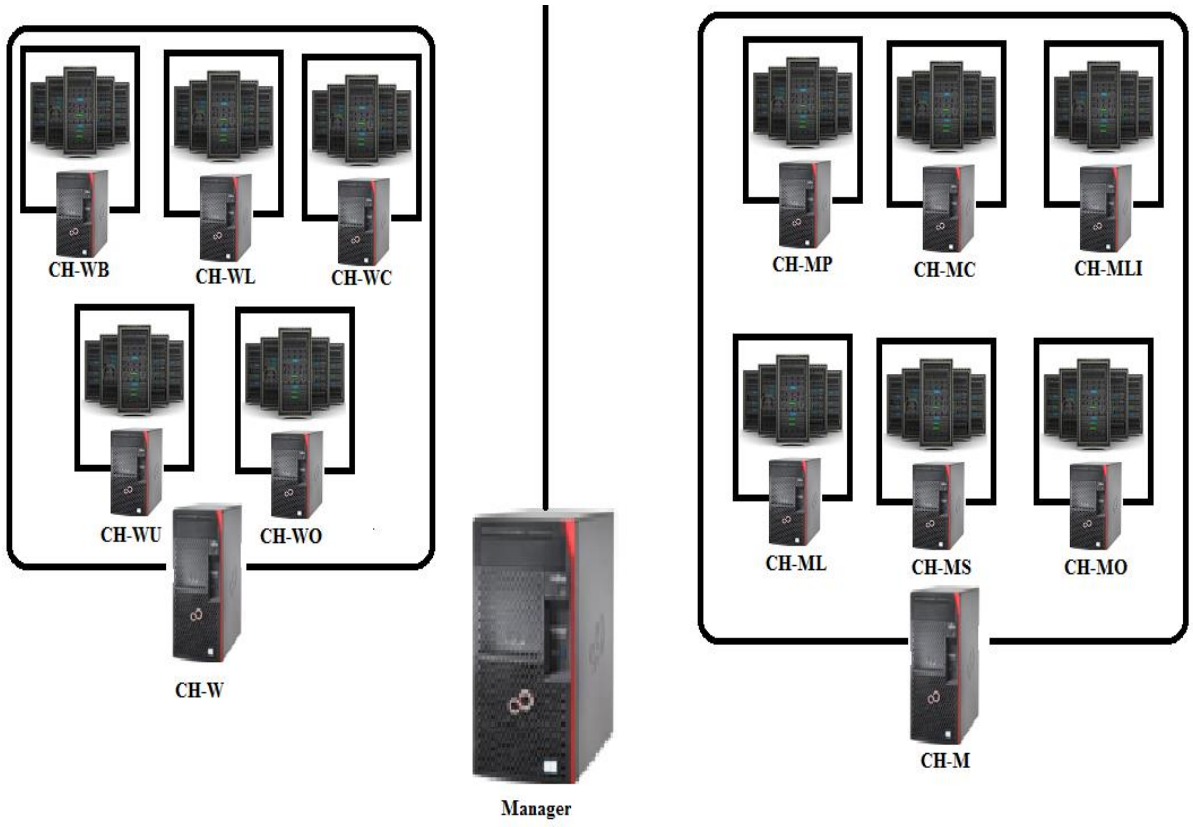


Figure 4.7 : Clusterisation des serveurs.

Les étiquettes de l'architecture PaaS-CBIR sont décrites dans le *tableau 4.1*.

Symbol	Description
CH-W	Cluster Head Women
CH-M	Cluster Head Men
CH-WB	Cluster Head Women Breast
CH-WL	Cluster Head Women Lung
CH-WC	Cluster Head Women Colorectal
CH-WU	Cluster Head Women Cervical
CH-WO	Cluster Head Women Other
CH-MP	Cluster Head Men Prostate
CH-ML	Cluster Head Men Lung
CH-MC	Cluster Head Men Colorectal
CH-MLI	Cluster Head Men Liver
CH-MS	Cluster Head Men Stomach
CH-MO	Cluster Head Men Other

Tableau 4.1 : Notation utilisée dans PaaS-CBIR.

- **Hypothèse**

• Pendant la phase hors ligne, tous les fichiers sont stockés au format DICOM.

• Tous les liens entre les serveurs sont robustes.

3.3. Description du sous-cluster

Chaque sous-cluster représente un système CBIR. Chaque sous-cluster contient la base de données d'images et la base de données d'index. Le sous-cluster ne contient que des images de l'organe spécifique. Chaque sous-cluster contient trois agents, un agent-couleur, un agent-texture et un agent-forme. Chaque sous-cluster est indépendant des autres sous-clusters.

3.4. Le rôle de chaque serveur

3.4.1. Manager

La phase hors ligne

Une fois que toutes les images sont téléchargées vers le Manager, ce dernier classe les images à l'aide de l'arbre de décision (voir *chapitre 3*). En fonction du sexe du patient, le Manager oriente l'image requête soit vers le CH-M soit vers le CH-W.

La phase en ligne

Le Manager reçoit une requête contenant l'image, le sexe et le type d'organe. Sur la base des informations reçues (sexe et organe), le Manager oriente la demande vers le cluster-head approprié (CH-W / CH-M).

Lorsque le Manager reçoit les images résultantes de la requête de CH-W ou CH-M, le Manager oriente ces images vers l'utilisateur approprié.

3.4.2. CH-W

La phase hors ligne

Une fois que le CH-W reçoit toutes les images du Manager, le CH-W les classe à l'aide d'un arbre de décision (voir *chapitre 3*). Selon le type d'organe, le CH-W oriente l'image vers le sous-cluster approprié (CH-WB, CH-WL, CH-WC, CH-WU ou CH-WO).

La phase en ligne

Lorsque le CH-W reçoit une requête du Manager, basée sur le type d'organe (Beast, Lung, Colorectal, Cervical, Other), le CH-W transmet la requête au sous-cluster qui stocke les images de l'organe.

Une fois que le CH-W reçoit les images résultantes de l'un des (CH-WB, CH-WL, CH-WC, CH-WU, CH-WO), le CH-W délègue ces images au Manager.

3.4.3. CH-M

Le processus de délégation d'images requêtes du CH-M dans les phases en ligne et hors ligne est similaire au processus du CH-W. La seule différence est que le CH-M oriente les images vers différents types de sous-cluster, y compris le CH-MP, le CH-ML, le CH-MC, le CH-MLI, le CH-MS et CH-MO.

3.4.4. CH-WX

Pour chaque CH-WX (où X est : Beast, Lung, Colorectal, Cervical, Other)

La phase hors ligne

Lorsque le CH-WX reçoit toutes les images du CH-W, le CH-WX extrait les caractéristiques d'images telles que la couleur, la texture et la forme selon les détails expliqués dans le *chapitre 1 paragraphe 7.2*. L'extraction introduit trois requêtes: requête-forme, requête-couleur et requête-texture. Ces requêtes sont envoyées respectivement à l'agent-forme, l'agent-couleur et l'agent-texture. Tous ces agents travaillent en parallèle pour extraire les caractéristiques de l'image. Chaque agent envoie les résultats au CH-WX. Le CH-WX regroupe ensuite les différentes caractéristiques en un seul vecteur de caractéristiques qui est stocké dans la base de données d'index. Ce traitement est appliqué à toutes les images de la base de données.

La phase en ligne

Lorsque CH-WX reçoit une demande de CH-W, il extrait les caractéristiques (forme, couleur et texture) de l'image requête comme expliqué dans le *chapitre 1 paragraphe 7.2*.

L'extraction soulève trois requêtes: requête-forme, requête-couleur et requête-texture. Ces requêtes sont envoyées respectivement à l'agent-forme, l'agent-couleur et l'agent-texture. Tous ces agents effectuent, en parallèle, la recherche de la demande correspondante (forme, couleur et texture). Chaque agent envoie les résultats au CH-WX. Le CH-WX agrège ensuite les différents résultats pour avoir un vecteur descripteur. Le CH-WX compare le vecteur de l'image requête avec chaque vecteur de la base de données d'index en utilisant la distance euclidienne comme expliqué dans le *chapitre 1 paragraphe 7.3.1*. Cette comparaison produit une liste de vecteurs les plus similaires au vecteur de l'image requête. Le CH-WX recherche les images appropriées dans la base de données et les envoie au CH-W. Lorsque le CH-W reçoit les images sélectionnées du CH-WX, le CH-W transmet ensuite les images au Manager.

3.4.5. CH-MX

Le CH-MX (où X est : Lung, Prostate, Colorectal, Liver, Stomach, Other) a le même processus que le CH-WX.

3.5. Diagramme de séquence

La *figure 4.8* présente le diagramme de séquence d'une requête d'utilisateur (par exemple, CH-WX).

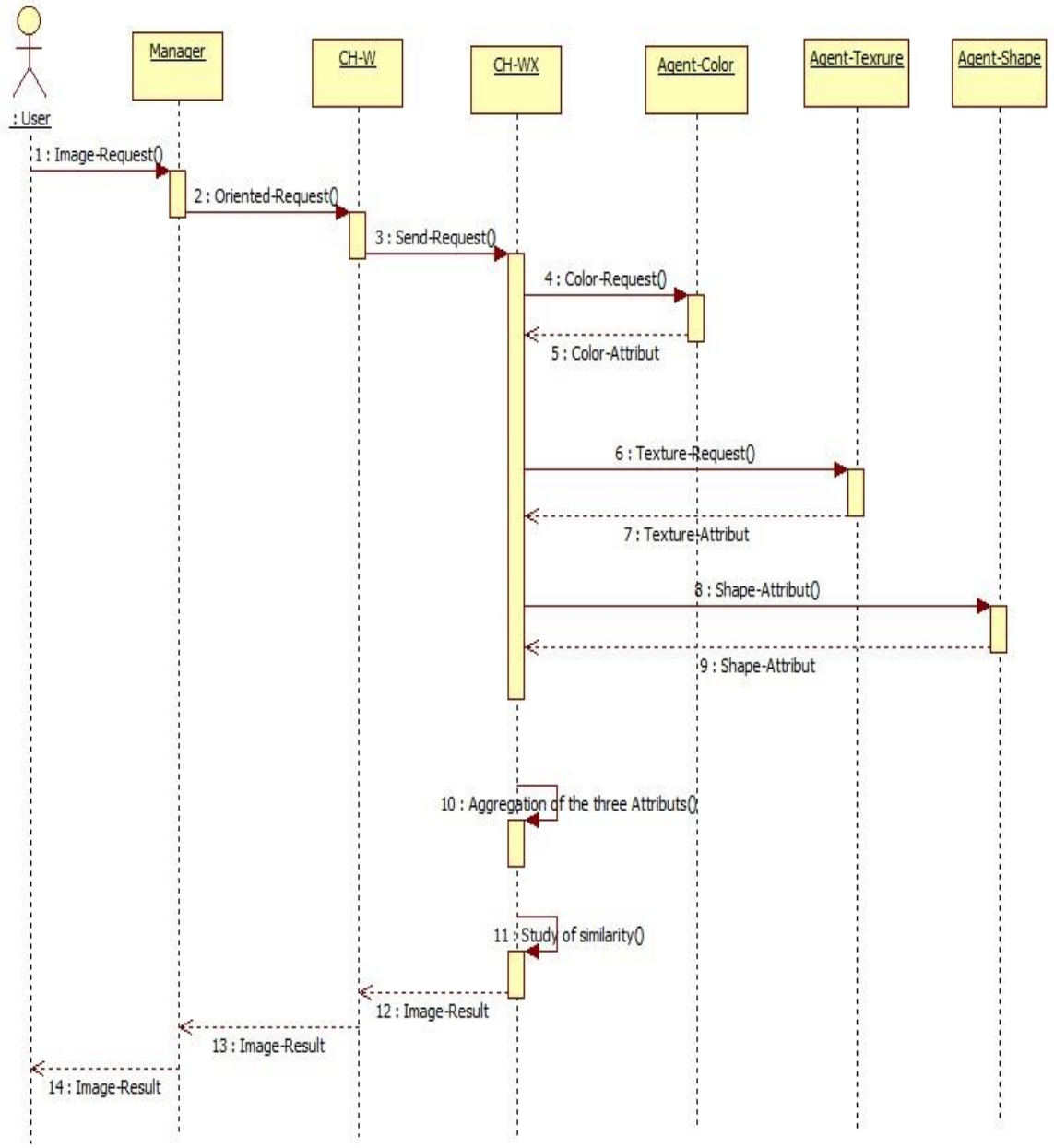


Figure 4.8 : diagramme de séquence d'une requête de l'utilisateur.

3.6. Description des algorithmes

3.6.1. La phase hors ligne

Algorithm 1 Manager

N: nombre total des images.
Img: image.
begin
for i:= 1 to N do
    If (sex-Img_i = female) then
        Oriented (Img_i) to CH-W
    Else / * (sex = male) * /
        Oriented (Img_i) to CH-M
    End if
End for
end.

Algorithm 2 CH-W

N: nombre d'image-sex = female.
Img: image.
X: breast, lung, colorectal, cervical and other.
Begin
For i:=1 to N do
    If (organ-Img_i=organ-typeX) then
        Oriented (Img_i) to CH-WX
    End if
End for
End.

Algorithm 3 CH-M

N: nombre d' image-sex = male.
Img: image.
X: prostate, lung, colorectal, liver, stomach and other.
Begin
For i:=1 to N do
    If (organ-Img_i=organ-typeX) then
        Oriented (Img_i) to CH-MX
    End if
End for
End.

Algorithm 4 CH-WX

X: breast, lung, colorectal, cervical and other.
N: nombre d'organ-image = organ-typeX.
Img: image
begin
For i:=1 to N do
/*Extract characteristics (color, shape and texture)*/
Send (Img_i) to agent-color.
Send (Img_i) to agent-texture.
Send (Img_i) to agent-shape.
If agent-color receive (Img_i) then
Extract attribute-color
End if
If agent-texture receive (Img_i) then
Extract attribute-texture
End if
If agent-shape receive (Img_i) then
Extract attribute-shape
End if
/*processing in parallel for the three agents*/
If I receive (((attribute-color) from the agent-color) and ((attribute-shape) from the agent-shape) and ((attribute-texture) from the agent-texture)) then
Feature-vector-Img_i:= Aggregation of the three features
End if
Store (Feature-vector-Img_i in Index database)

End for
End.

3.6.2. La phase en ligne

Algorithm 5 Manager

begin
If I receive (client-request: image-example, sex, organ) from client then
If (sex = female) then
Orient (client-request) to CH-W
Else / * (sex = male) * /
Orient (client-request) to CH-M
End if
End if
If I receive (((image-result) from CH-W) or ((image-result) from CH-M))) then
Transmit (image-result) to client
End if
end

Algorithm 6 CH-W

X: breast, lung, colorectal, cervical and other.
begin
If I receive (client-request) from the Manager then
    If (organ = organ-typeX) then
        Orient (client-request) to CH-WX
    End if
End if
If I receive (((image-result) from the CH-WX) then
    Transmit (image-result) to Manager
End if
End.

Algorithm 7 CH-M

X: prostate, lung, colorectal, liver, stomach and other.
begin
If I receive (client-request) from the Manager then
    If (organ = organ-typeX) then
        Orient (client-request) to CH-MX
    End if
End if
If I receive (((image-result) from the CH-MX) then
    Transmit (image-result) to Manager
End if
End.

Algorithm 8 CH-WX

```
begin
If I receive (client-request) from CH-W then
        /*Extract characteristics (color, shape and texture)*/
        Send (image-request) to agent-color
        Send (image-request) to agent-texture
        Send (image-request) to agent-shape
        If agent-color receive (image-request) then
                Extract attribute-color
        End if
        If agent-texture receive (image-request) then
                Extract attribute-texture
        End if
        If agent-shape receive (image-request) then
                Extract attribute-shape
        End if
/*processing in parallel for the three agents*/
        If I receive (((attribute-color) from the agent-color) and ((attribute-shape) from the
      agent-shape) and ((attribute-texture) from the agent-texture)) then
                Feature-vector-request:=Aggregation of the three characteristics
        End if
            List-Index:=  study of similarity between Feature-vector-request and Feature-
            vector-images in the index database
        Image-Result:= List-Similar Images
        Send (Images-Result) to CH-W
End if
End.
```

3.7. Méthodologie proposée

La couleur, la texture et la forme sont les principales caractéristiques d'une image. Ceux-ci sont extraits et placés dans un vecteur de descripteur/caractéristique. Lorsque l'utilisateur soumet une image-requête (phase en ligne), le même principe d'extraction de caractéristiques que celui utilisé dans la phase hors ligne est appliqué à cette demande. La distance euclidienne, comme expliqué dans le *chapitre 1 paragraphe 7.3.1*, est utilisée pour comparer la similitude entre le vecteur de caractéristiques de l'image-requête et les vecteurs de caractéristiques des images de la base de données.

Notre proposition passe par deux phases: la phase hors ligne et la phase en ligne.

3.7.1. Phase hors ligne

3.7.1.1. Collecte des images dans la base de données

- Toutes les images sont stockées au format DICOM, voir figure *4.10-a*.
- Chaque image est traitée au niveau du Manager pour être orientée vers le serveur approprié (CH-M / CH-W), voir la figure *4.10-b*.

3.7.1.2. Structure d'un arbre de décision

Un arbre de décision consiste en un ensemble de règles qui divise une population de cas (une base de données de patients, dans notre cas) en groupes (cluster) homogènes. Chaque règle associe une conjonction de tests sur les attributs d'un cas à un groupe (par exemple: "si sexe = male, alors le cas appartient au groupe" Men").

Ces règles sont organisées sous la forme d'un arbre dont la structure est la suivante (voir *figure 4.9*).

-Chaque nœud non terminal correspond à un test sur un descripteur (par exemple: "Sex =?").

-Chaque arc correspond à une réponse à un test (par exemple: "male")

- Chaque feuille correspond à un groupe de cas qui ont fourni la même réponse à tous les tests d'une règle (exemple: "hommes atteints de cancer du poumon").

La *figure 4.9* est un exemple d'une structure d'arbre de décision. Cet arbre de décision est composé de quatre règles qui divisent la population en quatre groupes: «si sexe = female et organe = Breast alors CH-WB», «si sexe = female et organe = Cervical alors CH-WU», «si sexe = male et organe = Lung alors CH-ML », « si sexe = male et organe = Prostate alors CH-MP».

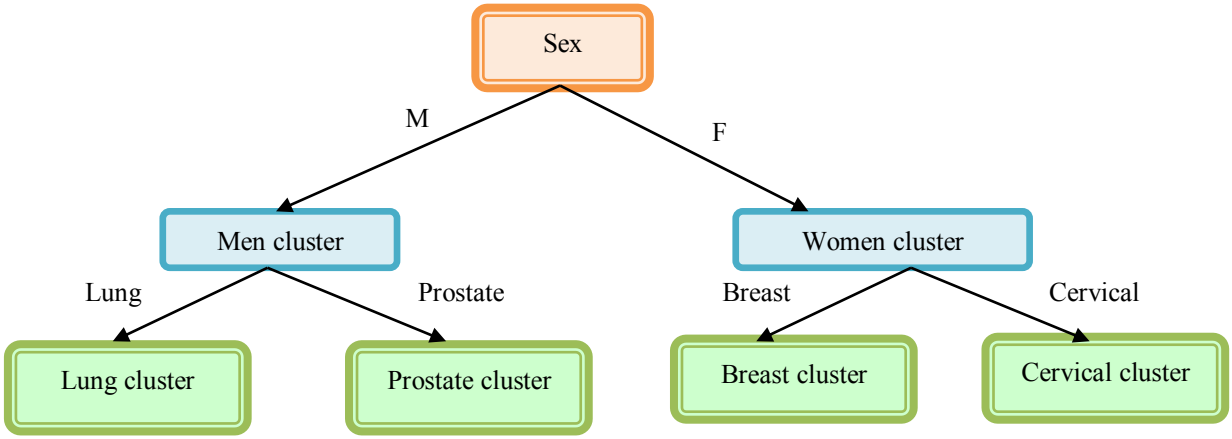


Figure 4.9 : Exemple d'une structure d'arbre de décision.

3.7.1.3. Construire un arbre de décision

Pour construire automatiquement un arbre de décision, nous devons rechercher les attributs les plus discriminants parmi tous les attributs disponibles (images et attributs contextuels, dans notre cas) et créer des groupes de cas homogènes à partir de tests sur ces attributs. Le mécanisme de construction, illustré à la *figure 4.10*, est basé sur un apprentissage supervisé. Pour commencer la construction, nous devons avoir plusieurs exemples classifiés. À l'initialisation de l'apprentissage, l'arbre est simplement une feuille, rassemblant toute la population comme le montre la *figure 4.10-a*. Récursivement, nous divisons ensuite chaque feuille *F* de l'arbre en construction. Nous recherchons le descripteur *d* le plus discriminant au sein de la population *P* regroupée en *F*. *P* est ensuite réparti entre les nouveaux nœuds enfants, un pour chaque réponse possible au test sur *d* voir *figure 4.10 (b, c, d)*.

La valeur des arbres de décision pour notre cas est que nous obtenons une segmentation de l'espace de données manipulé (voir *figure 4.11*).

Un exemple d'apprentissage d'un arbre de décision est illustré à la *figure 4.10 :* 1) l'attribut le plus discriminant sur l'ensemble de la population est le sexe, nous divisons donc la population totale entre celle des hommes et celle des femmes. 2) L'attribut le plus discriminant sur la population féminine est le type d'organe. Ainsi, nous divisons cette population selon le type d'organe. 3) L'attribut le plus discriminant sur la population des hommes est également le type d'organe; cette population est donc également divisée selon le type d'organe.

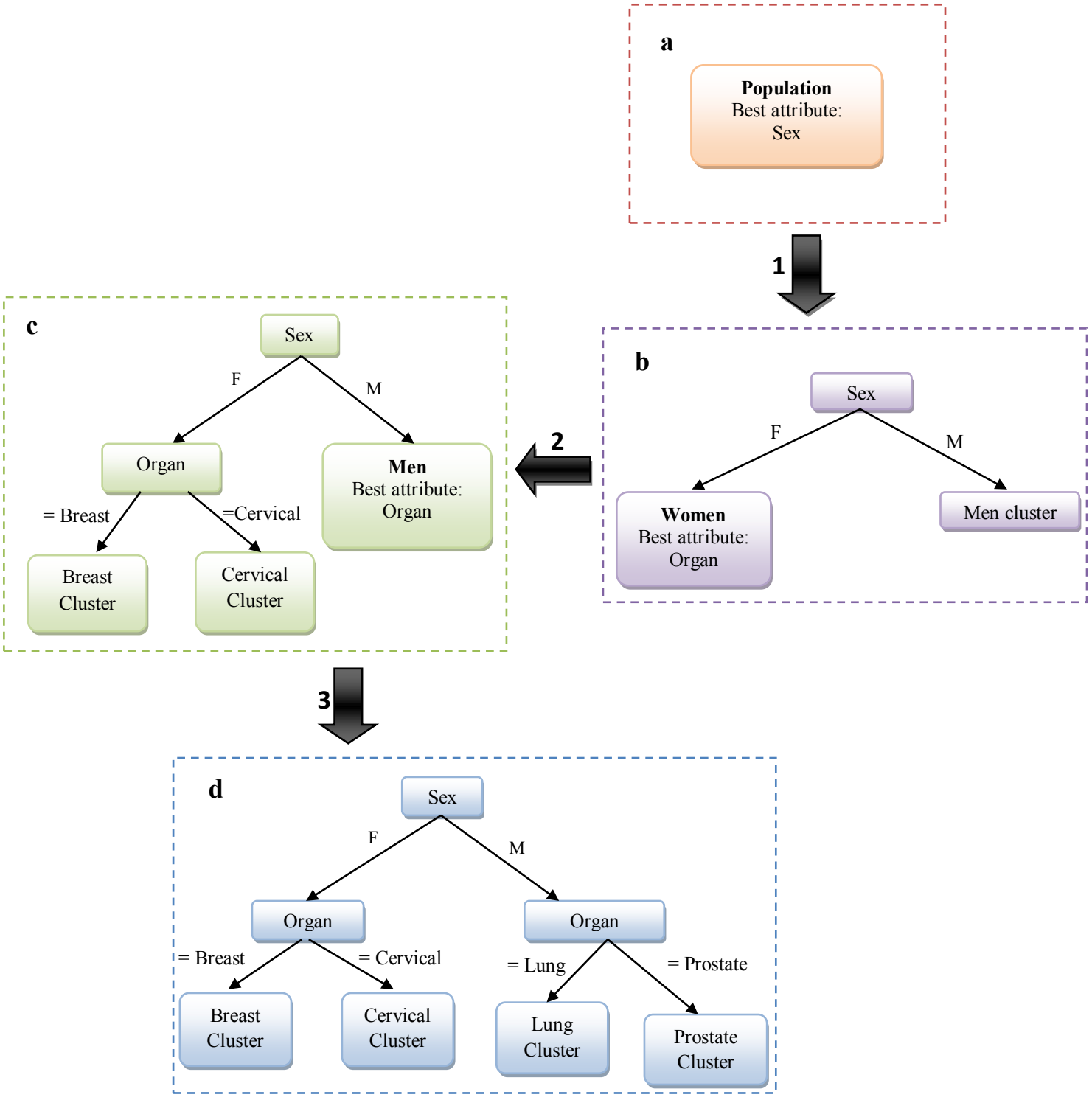


Figure 4.10 : Apprentissage d'un arbre de décision.

Pour classer les données, nous appliquons les Arbre de décision. La *figure 4.11* montre le principe de la classification des données.

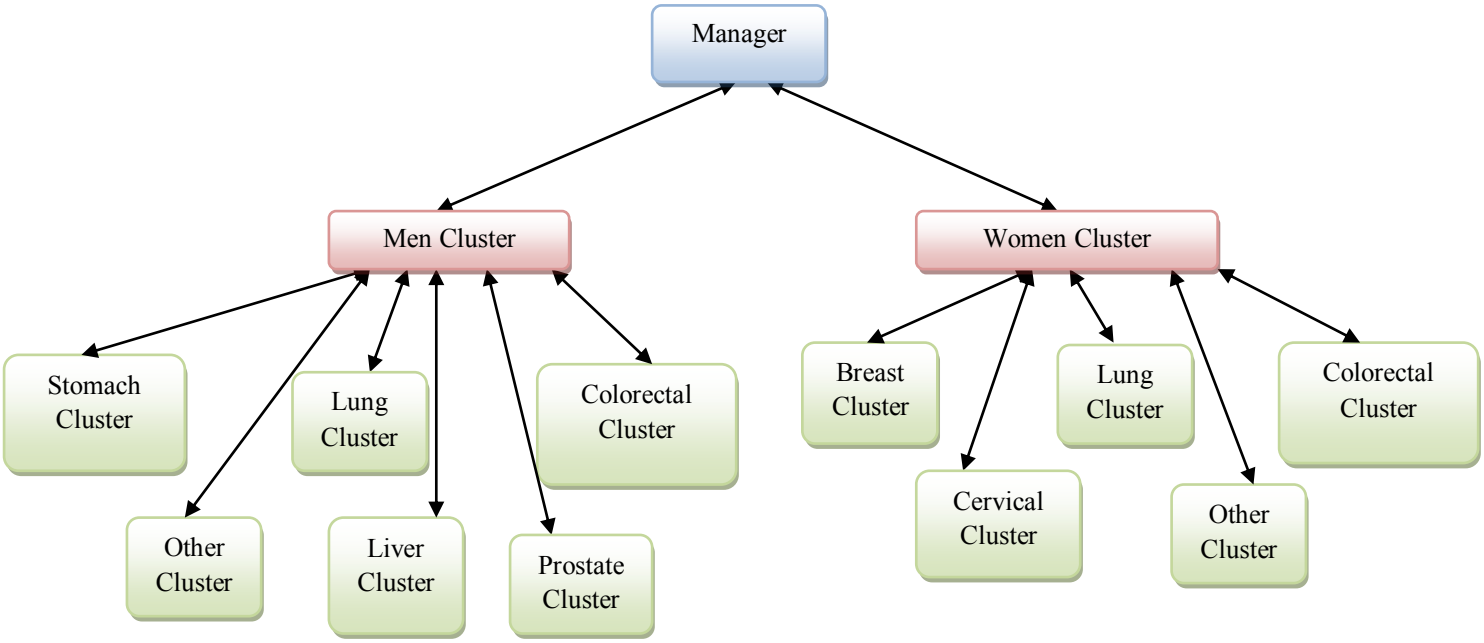


Figure 4.11 : classification des données.

Dans notre proposition, chaque cluster représente un serveur (ou même un ensemble de serveurs). Nous avons proposés que chaque cluster ait trois agents pour effectuer le traitement; un agent responsable de la couleur, un agent responsable de la texture et un agent responsable de la forme. Ces trois agents sont utilisés dans l'extraction des caractéristiques. Notre architecture organise 11 sous-clusters, et chaque sous-cluster déploie trois agents qui permettent une extraction plus rapide des caractéristiques (descripteurs) des images par rapport à l'architecture classique.

3.7.1.4. Extraction des caractéristiques (descripteurs)

L'objectif de cette étape est de construire un vecteur caractéristique de chaque image. Le vecteur caractéristique d'image contient des champs numériques.

Le choix des attributs dépend fortement des images de la base de données. Ainsi, les attributs qui donnent d'excellents résultats sur un ensemble d'images de base de données peuvent donner de mauvais résultats sur un autre ensemble. Aucun attribut universel ne fonctionne correctement sur une base de données d'images. Les attributs sont sensibles au contexte. Le principe général de notre proposition est indépendant du choix des attributs.

Nous proposerons un ensemble d'attributs pour valider et illustrer notre proposition, mais la proposition sera aussi utile en utilisant des familles d'attributs différents de ceux que nous utilisons.

- ***Extraction des caractéristiques de couleur***

La couleur est extraite à l'aide de deux attributs, tels que les histogrammes de couleur et les moments de couleur. Les moments de couleur ont été divisés en 3 moments d'ordre inférieur qui sont la moyenne de c μ_c, l'écart type σ_c et l'asymétrie θ_c, comme expliqué dans le *chapitre 1 paragraphe 7.2.1*.

- ***Extraction des caractéristiques de texture***

La texture est extraite en utilisant la matrice de cooccurrence de Haralick. Seules les quatre caractéristiques les plus appropriées sont largement utilisées (énergie, entropie, contraste et moment de différence inverse (IDM)), comme expliqué dans le *chapitre 1 paragraphe 7.2.2*.

- ***Extraction des caractéristiques de forme***

La caractéristique de forme est extraite en utilisant les sept moments invariants en rotation de Hu comme expliqué dans le *chapitre 1 paragraphe 7.2.3*.

Les caractéristiques extraites (couleur, texture et forme) sont stockées dans le vecteur de caractéristique/descripteur.

3.7.2. Phase en ligne

La phase hors ligne de notre proposition contient les étapes suivantes:

1. L'utilisateur soumet une requête qui contient l'image, le sexe et le type d'organe. Pour orienter la requête vers le cluster approprié, nous suivons le même principe du construction d'arbre de décision expliqué dans la phase hors ligne. L'extraction des caractéristiques (couleur, texture et forme) de l'image requête suit le même processus que l'extraction des caractéristiques d'image de la base de données dans la phase hors ligne.
2. La distance euclidienne est utilisée pour comparer la similitude entre le vecteur des caractéristiques de l'image requête et les vecteurs des caractéristiques de la base de données d'index.

3. Les images sont ensuite classées en fonction de la similitude.

3.8. Tolérance aux pannes dans PaaS-CBIR

Pour assurer la disponibilité des images dans PaaS-CBIR, nous proposons un mécanisme basé sur le Time Triggered Communication Protocol (TTC/P) pour assurer les services de surveillance, de sauvegarde et de restauration dans PaaS-CBIR.

En TTC/P, les serveurs sont autorisés à diffuser des données selon la technique TDMA (Time Division Multiple Access). TTP divise un tour TDMA en tranches de temps (TS : Time Slot). Il réserve un TS à chaque serveur du système pendant lequel il diffuse ses données [HAM 18, KOP 91, MAI 02, MAZ 08]. Le TTC/P garantit la tolérance aux pannes, la sureté de fonctionnement et la communication rapide des réseaux critiques.

Hypothèses

- Les horloges du serveur sont toujours synchronisées.
- Au plus, un serveur (représentant un organe) échouera à la fois.
- Seules des défaillances transitoires peuvent se produire.
- Il n'y a pas de panne de liens.
- Une défaillance transitoire ne peut se produire qu'au niveau du serveur récepteur.

3.8.1. Détection de la panne

Une panne peut survenir à tout moment et en tout lieu. Si un serveur tombe en panne, cela pourrait affecter le temps requis pour récupérer les images stockées sur ce serveur. Pour cette raison, une réplication d'image est nécessaire. Nous avons proposé une solution qui duplique le contenu de chaque serveur (i) dans serveur (i + 1). Ainsi, chaque serveur (i + 1) contient les images de son serveur voisin (i).

Chaque CH-X (où X est Men ou Women), organise l'adresse IP de tous les serveurs dans un ordre alternatif (round-robin). Une fois que le CH-X crée la liste d'ordonnancement, il la diffuse à tous les serveurs de son cluster. Sur la base de cette liste, chaque serveur peut communiquer avec son voisin dans la liste. Un serveur détecte l'état de santé de son voisin en lui envoyant un message 'heart-beat'.

Dans notre contribution, un tour TDMA est divisé en N intervalles de temps, où N est le nombre de serveurs d'organes. Chaque serveur a une période fixe (intervalle de temps $i \bmod N$) de communication unicast pendant laquelle il peut vérifier l'état de santé de son voisin. Si

l'expéditeur (i) reçoit un accusé de réception de son voisin, alors il en déduit que son voisin fonctionne correctement; sinon, il est hors service. Si le serveur (i + 1) est hors service, le serveur (i) informe immédiatement le CH-X de la défaillance du serveur (i + 1), comme montré sur *figure 4.12*.

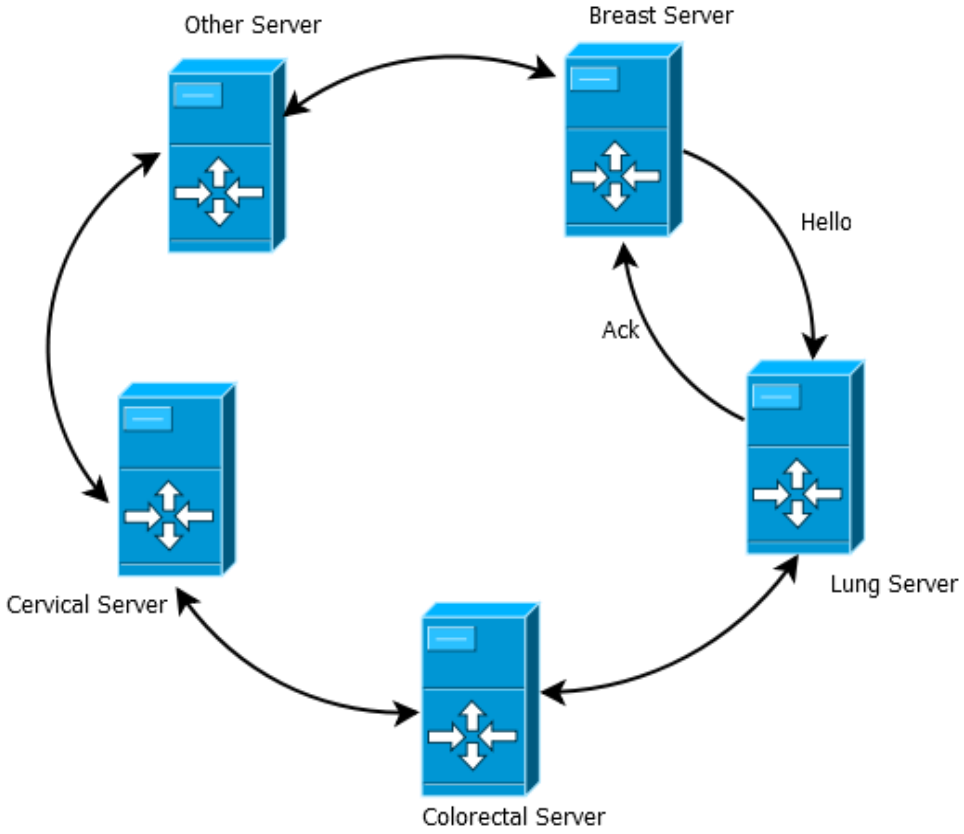


Figure 4.12 : Surveillance du système basée sur l'ordre préétabli des serveurs pour le système CBIR.

3.8.2. Réplication de données

A la réception d'une nouvelle image (par exemple, image du sein), par le serveur approprié *i* (par exemple, Breast Server), les nouvelles données reçues sont dupliquées sur son voisin *i+1*(par exemple, Lung Server). Cela se produit pendant la phase hors ligne. Les images originales d'un serveur *i+1* sont stockées séparément des images répliquées du voisin serveur *i*.

3.8.3. Récupération de données

Lorsque CH-X reçoit une requête *i* stockée dans le serveur *i*, il envoie un message hello au serveur *i*, pour vérifier qu'il fonctionne correctement. Si serveur *i* répond au CH-X

avec un accusé de réception, ce dernier lui assigne la requête; sinon, il transmet la requête au serveur *i+1*, comme indiqué sur la *figure 4.13*.

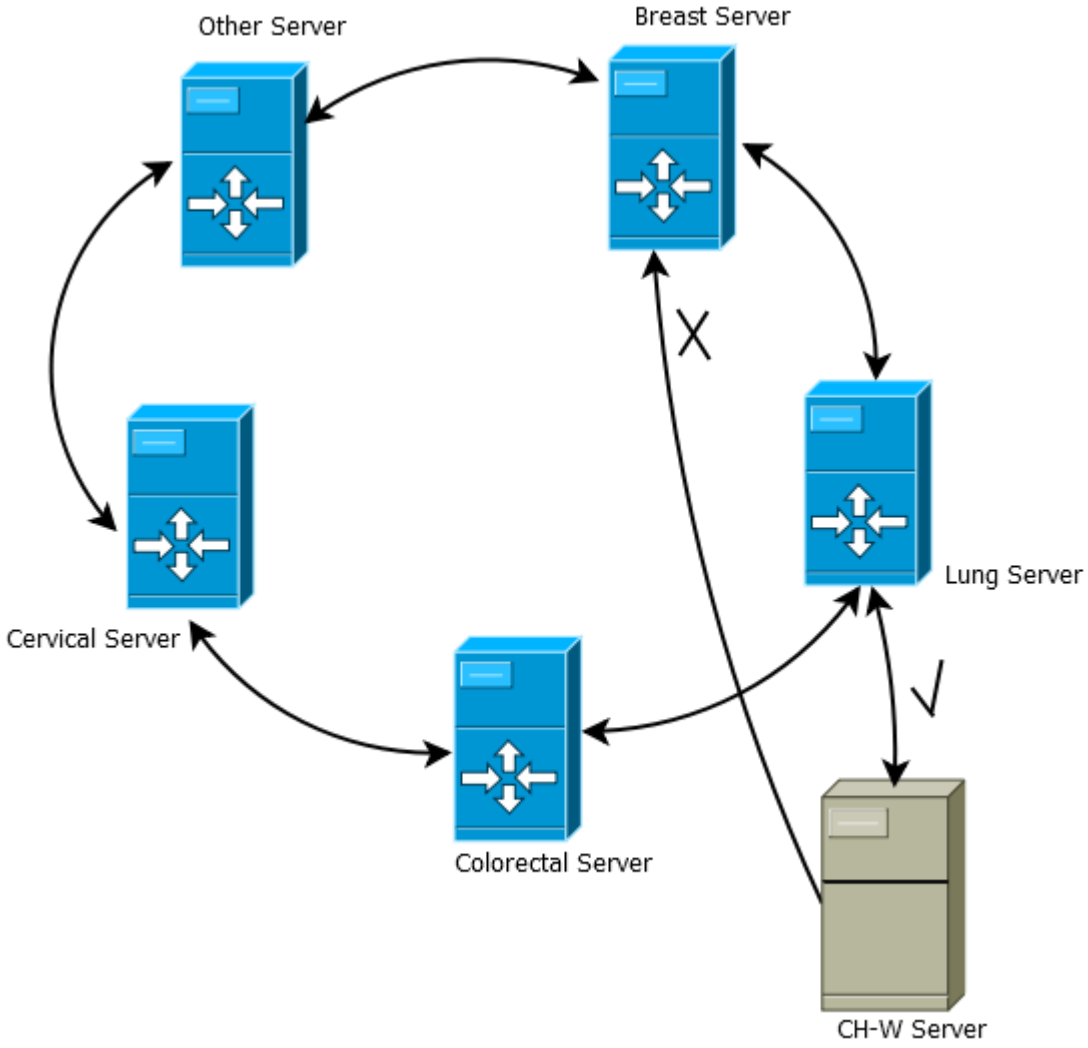


Figure 4.13 : Récupération de la version d'image répliquée.

3.8.4. Complexité de la récupération des données

Complexité = $N \times M$

M est le nombre d'images sur chaque serveur.

N est le nombre de serveurs.

Dans PaaS-CBIR, $N = 1$ et $M = 100$.

Dans l'architecture classique $N = 11$ et $M = 100$

La complexité dans PaaS-CBIR est de 100 et dans l'architecture classique, 1100 unités de temps. Comme le montre la *figure 4.14*, la complexité est proportionnelle au nombre d'images dans chaque serveur.

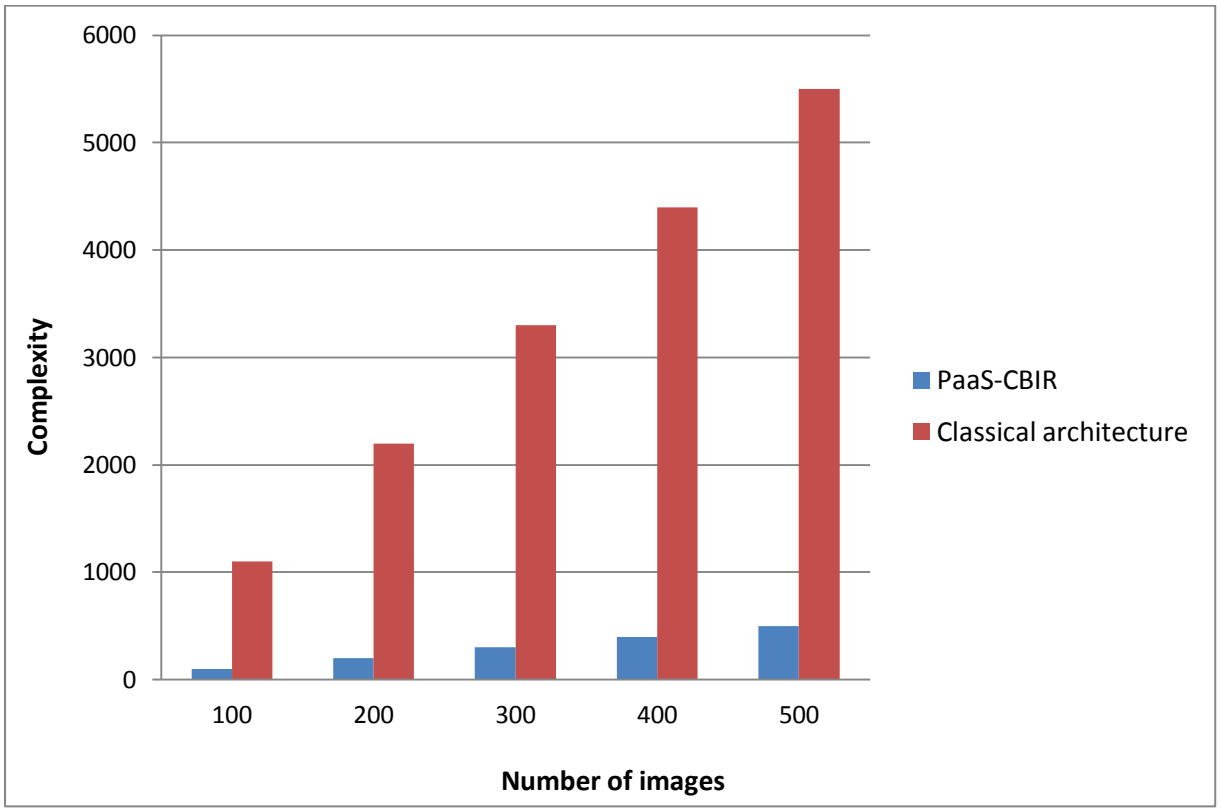


Figure 4.14 **:** Complexité des deux architectures en fonction du nombre d'images.

3.9. Évaluation formelle du PaaS-CBIR

Nous présentons une évaluation formelle du système PaaS-CBIR.

Théorème 1

Le PaaS-CBIR assure le traitement parallèle de onze types de demandes différents en même temps.

Preuve

PaaS-CBIR est basé sur l'arborescence indexée qui sépare la base de données selon deux sous-clusters (Men ou Women). Le sous-cluster qui stocke les images des males est composé de six serveurs. Chaque serveur traite chaque type d'organe indépendamment. Le sous-cluster qui contient la base de données des femelles contient cinq serveurs. Ces serveurs fonctionnent en parallèle, ce qui permet d'exécuter onze requêtes différentes en même temps.

Théorème 2

PaaS-CBIR surpasse l'architecture classique de 89% en termes de temps de réponse.

Preuve

N: nombre total d'images

a: temps de comparaison entre l'image requête et une image stockée dans la base de données.

b1: Dans l'architecture classique, le temps d'extraction des caractéristiques de l'image requête = extraction_time -forme + extraction_time -couleur + extraction_time - texture.

b2: Dans PaaS-CBIR, le temps d'extraction des caractéristiques de l'image requête = Max (extraction_time-forme, extraction_time-couleur, extraction_time-texture).

Nous prenons la valeur maximale, car il y a trois agents qui sont exécutés en parallèle. Par conséquent, *b1*> *b2*

L'architecture classique:

Le temps de réponse_Classique = $\sum communication_{time} + b_1 + (N * a\ time)$

$Communication_{time}$= nombre de serveurs * α

α : temps nécessaire pour transmettre des informations d'un serveur à un autre.

Time: est une unité de temps.

Le temps de réponse_Classique = $11 * \alpha + b_1 + (N * a\ time)$

PaaS-CBIR:

La recherche se fera au niveau de: $\frac{N}{2} = X$(selon le sexe)

Au pire des cas:

La recherche se fera au niveau de: $\frac{X}{5} = X1$ (selon l'organe)

Le temps de réponse_PaaS-CBIR = $\sum communication_{time} + b_2 + X1 * a\ time$

$$X1 = \frac{X}{5} = \frac{\frac{N}{2}}{5} = \frac{N}{10}$$

Le temps de réponse_PaaS-CBIR =$\sum communication_{time} + b_2 + (\frac{1}{10} N * a\ time)$

$Communication_{time}$= nombre de serveurs * α

α: temps nécessaire pour transmettre des informations d'un serveur à un autre.

Time: est une unité de temps.

Le temps de réponse_PaaS-CBIR $= 2 * \alpha + b_2 + \left(\frac{1}{10} N * a\ time\right)$

Théorème 3

PaaS-CBIR fournit des résultats précis aux spécialistes utilisant la base de données. Il évite complètement le problème de fossé/vide sémantique.

Preuve

La recherche est orientée vers un sous-ensemble de la base de données qui ne contient que des images pertinentes présélectionnées par sexe, puis par organe, ce qui limite le type d'image (par exemple, sein). Parce que nous recherchons un ensemble d'images pertinentes, seules les images liées à la demande d'image seront renvoyées, ainsi le problème de fossé sémantique est minime.

Théorème 4

PaaS-CBIR diminue la quantité de surcharge, définie comme le mouvement des tâches pendant la communication inter-processus et inter-processeur, provoqué par une délégation déséquilibrée des demandes.

Preuve

Dans l'architecture classique, supposons que nous ayons *K* serveurs et *N* requêtes. Chaque demande doit être vérifiée de manière séquentielle jusqu'à ce qu'elle soit trouvée. Dans le pire des cas, lorsque les serveurs (*K-1*) ne contiennent pas la demande requise, il y a une forte probabilité que la demande soit satisfaite sur le $K^{ème}$ serveur. Ainsi, les serveurs (*K-1*) sont inutilement occupés, ce qui augmente les frais généraux de ces serveurs. PaaS-CBIR délègue la demande directement au serveur approprié, ce qui diminue le taux de surcharge.

Théorème 5

PaaS-CBIR manipule le Big Data avec une grande efficacité.

Preuve

PaaS-CBIR offre une capacité de traitement de stockage illimitée, ce qui permet de gérer un nombre innombrable d'images. Cet utilitaire satisfait la caractéristique Volume. Il permet également d'incorporer des données provenant de sources et de formats distincts associés à différentes images. Cet avantage contient la caractéristique Variété. PaaS-CBIR fournit une réponse rapide aux demandes en raison de l'architecture distribuée et le parallélisme conçu dans la proposition satisfait la troisième caractéristique du Big Data, qui est la vitesse.

Théorème 6

PaaS-CIBR garantit la disponibilité des images avec une latence minimale en cas de panne du serveur.

Preuve

Si CH-X reçoit une demande du gestionnaire et que CH-X détecte que le serveur(i) qui stocke les images requises est hors service, alors CH-X oriente directement la demande d'image vers le voisin du serveur (serveur (i + 1)). Ainsi, il récupère les versions répliquées des images similaires sans récupérer tous les serveurs du Cloud.

4. Conclusion

Dans ce chapitre, nous avons détaillé nos contributions.

Pour notre première contribution, nous avons mis en place un système d'indexation et de recherche visuo-textuelle d'images. En utilisant dans un premier lieu une recherche par mot clé. En second lieu, nous avons exploité le principe d'un CBIR pour faire une recherche par image exemple. Finalement, nous avons combiné entre les deux premières méthodes de recherche pour aboutir à une recherche visuo-textuelle d'images.

Pour notre deuxième contribution, nous nous sommes intéressés aux bases de données spécialisées. L'approche que nous avons explorée est basée sur les CBIR. Cette contribution est une proposition d'un modèle efficace de plate-forme en tant que service (PaaS) dans le Cloud public pour le système CBIR. L'objectif principal de cette contribution est de

minimiser le temps de réponse pour une image-requête et de diminuer le vide/fossé sémantique. La plate-forme proposée est rapide car la plupart du temps de calcul est consacré à la phase hors ligne. Le PaaS-CBIR peut être mis en œuvre de manière pratique pour soutenir le diagnostic médical.

Conclusion générale

Les progrès technologiques constants dans le domaine de l'archivage des données numériques nous permettent aujourd'hui d'avoir accès à une quantité d'informations inégalée dans l'histoire. Tous les domaines de l'activité humaine sont concernés, et les problèmes ne sont pas simplement les volumes d'informations archivées, mais aussi l'utilisation de ces données et la recherche d'informations pertinentes pour une utilisation donnée. Une problématique nouvelle est apparue: gérer les quantités énormes et croissantes de données (plus particulièrement les bases images).

Cette thèse s'est articulée autour de l'indexation et la recherche dans les grandes bases d'images (***chapitre 1***), il s'agissait de décrire les images par un ensemble de descripteurs et de les stocker autour d'une structure d'indexation performante en utilisant une distance ou une mesure de similarité.

Le défi principal motivant ce travail réside dans le fait de minimiser le temps de réponse pour une requête donnée et de combler le vide/fossé sémantique.

Notre travail est constitué de deux phases principales:

La **première phase** est la phase hors ligne (étape d'indexation), nous calculons les signatures d'images et nous les stockons dans une base de données. Lors de la phase d'indexation, le calcul de signature consiste en l'extraction des caractéristiques visuelles des images. Cependant, plusieurs méthodes de mesures de similarité ont été proposées dans la littérature dont certaines d'entre elles sont présentées dans le premier chapitre.

La **seconde phase** est la phase en ligne (étape de recherche). Nous utilisons le principe de la recherche-par-similarité selon lequel les images sont ordonnées à l'aide d'une mesure de similarité à la requête. Cette dernière prend la forme d'une image exemple. Le système calcule la signature selon le même mode que lors de la première phase d'indexation.

Cette thèse s'est focalisée sur l'hybridation texte/image, c'est-à-dire que la description de l'image est extraite de son contenu en même temps que d'autres sources d'informations externes. L'indexation textuelle n'est pas directement concurrente mais plutôt complémentaire de l'indexation basée-contenu. En effet, l'image seule ne permet pas de

répondre à des requêtes abstraites. La coopération de l'information image et de l'information textuelle n'est cependant pas évidente et immédiate à mettre en œuvre.

Dans un premier temps, nous avons implémenté un système d'indexation et de recherche visuo-textuelle pour les bases de données généralistes. Puis dans la seconde et la principale contribution, nous nous sommes intéressés aux bases de données spécialisées dans le domaine médical, où de plus en plus d'informations relatives aux patients, aux pathologies et aux connaissances médicales sont enregistrées, archivées dans des bases de données utiles pour la formation et le diagnostic. Ces bases de données contenant des fichiers DICOM qui contiennent à la fois des images numériques et des informations sémantiques, ce qui nécessite la manipulation de Big data (***chapitre 2***). L'approche que nous avons explorée est basée sur le Content Based Image Retrieval (CBIR).

Plus précisément, la contribution de ce travail est la proposition d'un modèle de plate-forme en tant que service (PaaS) efficace sur un Cloud public (***chapitre 2***) pour le système CBIR. La plateforme proposée est rapide car l'essentiel du temps de calcul est passé pendant la phase hors ligne.

Le PaaS-CBIR a été proposé pour faciliter le diagnostic médical. PaaS-CBIR manipule les informations sémantiques, ainsi que des images numériques. En utilisant les arbres de décision (***chapitre 3***), l'architecture proposée permet d'améliorer le temps de calcul d'un système d'aide au diagnostic médical. Le PaaS-CBIR peut être implémenté d'une manière pratique pour soutenir le diagnostic médical. PaaS-CBIR gère le Big Data avec une grande efficacité. PaaS-CBIR crée des liens intelligents entre les serveurs qui garantissent la disponibilité des données avec un temps de latence minimum.

Enfin, l'ensemble des travaux présentés dans cette thèse ont fait l'objet d'une communication dans une conférence internationale ([HAD 14]) et d'une publication dans une revue scientifique ([HAD 20]).

Perspectives

Cette thèse constitue une première tentative de proposition d'une plate-forme en tant que service (PaaS) dans le Cloud pour le système CBIR. Pour notre travail futur, nous prévoyons d'enrichir notre proposition en utilisant d'autres méthodes (transformée en ondelettes, SIFT, SURF,…) pour extraire des caractéristiques de l'image. La seconde

amélioration que nous envisageons interviendra dans la phase de sauvegarde, cette contribution permettra le stockage dynamique dans la plateforme PaaS-CBIR. Pour la suite, nous proposerons une hiérarchie pour l'utilisation des signatures/index et une deuxième hiérarchie pour les attributs qu'elles contiennent (couleur, texture et forme).

Bibliographie

[ABI 19] ABIRAMI, N. et GAVASKAR, S. CONTENT BASED IMAGE RETRIEVAL TECHNIQUES FOR RETRIEVAL OF MEDICAL IMAGES FROM LARGE MEDICAL DATASETS–A SURVEY. *International Journal of Advanced Research in Computer Science*, 2019, vol. 10, no 1, p. 50.

[AIG 96] AIGRAIN, Philippe, ZHANG, HongJiang, et PETKOVIC, Dragutin. Content-based representation and retrieval of visual media: A state-of-the-art review. *Multimedia tools and applications*, 1996, vol. 3, no 3, p. 179-202.

[ALM 16] ALMARABEH, Tamara, MAJDALAWI, Yousef Kh, et MOHAMMAD, Hiba. Cloud computing of e-government. 2016.

[ALZ 16] ALZAHRANI, Hibatullah. A brief survey of cloud computing. *Global Journal of Computer Science and Technology*, 2016.

[ANS 16] ANSHARI, Muhammad, ALAS, Yabit, et GUAN, Lim Sei. Developing online learning resources: Big data, social networks, and cloud computing to support pervasive knowledge. *Education and Information Technologies*, 2016, vol. 21, no 6, p. 1663-1677.

[BAT 18] BATRA, Mridula et AGRAWAL, Rashmi. Comparative analysis of decision tree algorithms. In : *Nature inspired computing*. Springer, Singapore, 2018. p. 31-36.

[BAY 06] BAY, Herbert, TUYTELAARS, Tinne, et VAN GOOL, Luc. Surf: Speeded up robust features. In : *European conference on computer vision*. Springer, Berlin, Heidelberg, 2006. p. 404-417.

[BEL 02] BELONGIE, Serge, MALIK, Jitendra, et PUZICHA, Jan. Shape matching and object recognition using shape contexts. *IEEE transactions on pattern analysis and machine intelligence*, 2002, vol. 24, no 4, p. 509-522.

[BELS 56] BELSON, William A. A technique for studying the effects of a television broadcast. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 1956, vol. 5, no 3, p. 195-202.

[BEN 17] BENKRAMA Soumia. *Indexation et Recherche d'Images par Contenu dans les Grands Corpus d'Images. 2017.* Thèse de doctorat. Université des Sciences et Technologies d'Oran Mohamed Boudiaf.

[BIA 08] BIAU, GÃŠrard, DEVROYE, Luc, et LUGOSI, GÃĄbor. Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research*, 2008, vol. 9, no Sep, p. 2015-2033.

[BIA 16] BIAU, Gérard et SCORNET, Erwan. A random forest guided tour. *Test*,

2016, vol. 25, no 2, p. 197-227.

[BOL 95] BOLON, Philippe, CHASSERY, Jean-Marc, COCQUEREZ, Jean-Pierre, *et al. Analyse d'images: filtrage et segmentation*. 1995.

[BRE 84] BREIMAN, Leo, FRIEDMAN, Jerome, STONE, Charles J., *et al. Classification and regression trees*. CRC press, 1984.

[BRE 96] BREIMAN, Leo. Bagging predictors. *Machine learning*, 1996, vol. 24, no 2, p. 123-140.

[BRE 01] BREIMAN, Leo. Random forests. *Machine learning*, 2001, vol. 45, no 1, p. 5-32.

[CHA 04] CHABRIAIS, J. et GIBAUD, B. DICOM, le standard pour l'imagerie médicale. *EMC-Radiologie*, 2004, vol. 1, no 6, p. 577-603.

[CHAN 15] CHANG, Wo L., GRADY, Nancy, *et al. NIST Big Data Interoperability Framework: Volume 1, Big Data Definitions*. 2015.

[COD 02] CODD, Edgar F. A relational model of data for large shared data banks. In : *Software pioneers*. Springer, Berlin, Heidelberg, 2002. p. 263-294.

[COI 05] COIFMAN, Ronald R., LAFON, Stephane, LEE, Ann B., *et al.* Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the national academy of sciences*, 2005, vol. 102, no 21, p. 7426-7431.

[DAU 85] DAUGMAN, John G. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *JOSA A*, 1985, vol. 2, no 7, p. 1160-1169.

[DEN 01] DENG, Yining et MANJUNATH, B. S. Unsupervised segmentation of color-texture regions in images and video. *IEEE transactions on pattern analysis and machine intelligence*, 2001, vol. 23, no 8, p. 800-810.

[DIA 06] DÍAZ-URIARTE, Ramón et DE ANDRES, Sara Alvarez. Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 2006, vol. 7, no 1, p. 3.

[DIE 71] DIERK, S. F. The SMART retrieval system: Experiments in automatic document processing—Gerard Salton, Ed.(Englewood Cliffs, NJ: Prentice-Hall, 1971, 556 pp., $15.00). *IEEE Transactions on Professional Communication*, 1972, no 1, p. 17-17.

[DMI 09] DIMITROVSKI, Ivica, GUGULJANOV, Pero, et LOSKOVSKA, Suzana. Implementation of web-based medical image retrieval system in oracle. In : *2009 2nd International Conference on Adaptive Science & Technology (ICAST)*. IEEE, 2009. p. 192-197.

[DOU 95] DOUGHERTY, James, KOHAVI, Ron, et SAHAMI, Mehran. Supervised and unsupervised discretization of continuous features. In : *Machine learning*

proceedings 1995. Morgan Kaufmann, 1995. p. 194-202.

[DUB 10] DUBEY, Rajshree S., CHOUBEY, Rajnish, et BHATTACHARJEE, Joy. Multi feature content based image retrieval. *International Journal on Computer Science and Engineering*, 2010, vol. 2, no 6, p. 2145-2149.

[FER 03] FERRELL, Regina K., GLEASON, Shaun S., et TOBIN JR, Kenneth W. Application of fractal encoding techniques for image segmentation. In : *Sixth International Conference on Quality Control by Artificial Vision*. International Society for Optics and Photonics, 2003. p. 69-77.

[FLI 95] FLICKNER, Myron, SAWHNEY, Harpreet, NIBLACK, Wayne, *et al.* Query by image and video content: The QBIC system. *computer*, 1995, vol. 28, no 9, p. 23-32.

[FOU 02] FOURNIER, Jérôme. *Indexation d'images par le contenu et recherche interactive dans les bases généralistes*. 2002. Thèse de doctorat. Cergy-Pontoise.

[FRE 96] FREUND, Y., Schapire, R. E., et al. Experiments with a new boosting algorithm. In *13th International Conference on Machine Learning*, 1996, volume 96, pages 148–156.

[GAB 46] GABOR, Dennis. Theory of communication. Part 1: The analysis of information. *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, 1946, vol. 93, no 26, p. 429-441.

[GAR 83] GARDARIN, Georges, GARDARIN, Georges, GARDARIN, Georges, *et al.* *Bases de données: les systèmes et leurs langages*. Eyrolles, 1983.

[GAZ 13] EL-GAZZAR, Rania et WAHID, Fathul. An Analytical Framework to Understand the Adoption of Cloud Computing: An Institutional Theory Perspective. In : *ICCSM2013-Proceedings of the International Conference on Cloud Security Management: ICCSM 2013*. Academic Conferences Limited, 2013. p. 91.

[GEN 10] GENUER, Robin. *Forêts aléatoires: aspects théoriques, sélection de variables et applications*. 2010. Thèse de doctorat.

[GEY 05] GEY, Servane et NEDELEC, Elodie. Model selection for CART regression trees. *IEEE Transactions on Information Theory*, 2005, vol. 51, no 2, p. 658-670.

[GEY 12] GEY, Servane. Risk bounds for cart classifiers under a margin condition. *Pattern Recognition*, 2012, vol. 45, no 9, p. 3523-3534.

[GRE 01] GRECU, Horia et LAMBERT, Patrick. Indexation par descripteurs flous: Application à la recherche d'images. In : *18° Colloque sur le traitement du signal et des images, FRA, 2001*. GRETSI, Groupe d'Etudes du Traitement du Signal et des Images, 2001.

[GUP 17] GUPTA, A. S. B. et THAKUR, S. S. Cloud computing: its characteristics, security issues and challenges. Review of Computere Engineering Studies 4 (2): 76-81. 2017.

[HAD 14] HADI, Akram, HADI, Fairouz et BOUKERRAM, Abdallah. Indexation et recherche visuo-textuelle des bases de données images. *International Conference on Artificial Intelligence and Information Technology (ICAIIT'2014), Ouargla, Algeria*.

[HAD 20] HADI, Fairouz, ALIOUAT, Zibouda, et HAMMOUDI, Sarra. Efficient Platform as a Service (PaaS) Model on Public Cloud for CBIR System. *Ingénierie des Systèmes d'Information*, 2020, vol. 25, no 2, p. 215-225.

[HAF 95] HAFNER, James, SAWHNEY, Harpreet S.. , EQUITZ, William, *et al.* Efficient color histogram indexing for quadratic form distance functions. *IEEE transactions on pattern analysis and machine intelligence*, 1995, vol. 17, no 7, p. 729-736.

[HAM 18] HAMMOUDI, Sarra; ALIOUAT, Zibouda; HAROUS, Saad. A new Infrastructure as a Service for IoT-Cloud. In: *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 2018. p. 786-792.

[HAR 73] HARALICK, Robert M., SHANMUGAM, Karthikeyan, et DINSTEIN, Its' Hak. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, 1973, no 6, p. 610-621.

[HART 72] HART, Peter E. et DUDA, R. O. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 1972, vol. 15, no 1, p. 11-15.

[HAST 09] HASTIE, Trevor, TIBSHIRANI, Robert, et FRIEDMAN, Jerome. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[HAS 15] HASHEM, Ibrahim Abaker Targio, YAQOOB, Ibrar, ANUAR, Nor Badrul, *et al.* The rise of "big data" on cloud computing: Review and open research issues. *Information systems*, 2015, vol. 47, p. 98-115.

[HOU 10] HOUARI, Kamel et MOHAMED-KHIREDDINE, Kholladi. *Recherche d'images par le contenu*. 2010. Thèse de doctorat. Thèse de doctorat, Université Mentouri Constantine.

[HOUG 18] HOUGH, Paul VC. *Method and means for recognizing complex patterns*. U.S. Patent No 3,069,654, 18 déc. 1962.

[HU 62] Hu, M.K. 1962. Visual pattern recognition by moment invariants. IRE Transactions on Information Theory, 8(2):179-187. https://doi.org/10.1109/TIT.1962.1057692

[HUA 97] HUANG, Jing, KUMAR, S. Ravi, MITRA, Mandar, *et al.* Image indexing using color correlograms. In : *Proceedings of IEEE computer society conference on Computer Vision and Pattern Recognition*. IEEE, 1997. p. 762-768.

[HUN 62] HUNT, Earl B. Concept learning: An information processing problem. 1962.

[HUT 11] HUTH, Alexa et CEBULA, James. The basics of cloud computing. *United States Computer*, 2011.

[JOB 14] JOBAY, Rahmeh et SLEIT, Azzam. Quantum inspired shape representation for content based image retrieval. *Journal of Signal and Information Processing*, 2014, vol. 5, no 02, p. 54.

[KAS 80] KASS, Gordon V. An exploratory technique for investigating large quantities of categorical data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 1980, vol. 29, no 2, p. 119-127.

[KE 04] KE, Yan et SUKTHANKAR, Rahul. PCA-SIFT: A more distinctive representation for local image descriptors. In : *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* IEEE, 2004. p. II-II.

[KON 93] KONONENKO, Igor. Inductive and Bayesian learning in medical diagnosis. *Applied Artificial Intelligence an International Journal*, 1993, vol. 7, no 4, p. 317-337.

[KOP 91] KOPETZ, Hermann. Event-triggered versus time-triggered real-time systems. In : *Operating Systems of the 90s and Beyond*. Springer, Berlin, Heidelberg, 1991. p. 86-101.

[KUL 51] KULLBACK, Solomon et LEIBLER, Richard A. On information and sufficiency. *The annals of mathematical statistics*, 1951, vol. 22, no 1, p. 79-86.

[KUM 16] KUMAR, Manoj et SINGH, Manglem. CBMIR: Content Based Medical Image Retrieval System Using Texture and Intensity for Eye Images. *International Journal of Scientific & Engineering Research, Volume 7, Issue 9, September-2016*

[KUS 16] KUSRINI, Kusrini, ISKANDAR, M. Dedi, et WIBOWO, Ferry Wahyu. Multi Features Content-Based Image Retrieval Using Clustering And Decision Tree Algorithm. *Telkomnika*, 2016, vol. 14, no 4, p. 1480.

[LAF 01] LAFFERTY, John, MCCALLUM, Andrew, et PEREIRA, Fernando CN. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[LAFO 06] LAFON, Stephane, KELLER, Yosi, et COIFMAN, Ronald R. Data fusion and multicue data matching by diffusion maps. *IEEE Transactions on pattern*

analysis and machine intelligence, 2006, vol. 28, no 11, p. 1784-1797.

[LAN 08] LANDRE, Jérôme. *Analyse multirésolution pour la recherche et l'indexation d'images par le contenu dans les bases de données images-Application à la base d'images paléontologique Trans' Tyfipal*. 2005. Thèse de doctorat.

[LIM 00] LIM, Tjen-Sien, LOH, Wei-Yin, et SHIH, Yu-Shan. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine learning*, 2000, vol. 40, no 3, p. 203-228.

[LOH 97] LOH, Wei-Yin et SHIH, Yu-Shan. Split selection methods for classification trees. *Statistica sinica*, 1997, p. 815-840.

[LOW 04] LOWE, David G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 2004, vol. 60, no 2, p. 91-110.

[MA 99] MA, Wei-Ying et MANJUNATH, Bangalore S. Netra: A toolbox for navigating large image databases. *Multimedia systems*, 1999, vol. 7, no 3, p. 184-198.

[MAH 19] MAHMOUDI, Sidi Ahmed, BELARBI, Mohammed Amin, DADI, El Wardani, *et al.* Cloud-Based Image Retrieval Using GPU Platforms. *Computers*, 2019, vol. 8, no 2, p. 48.

[MAHA 30] MAHALANOBIS, P. C. On Tests and Meassures of Groups Divergence I. Journal of the Asiatic Society of. Benagal. 1930.

[MAI 02] MAIER, Reinhard. Event-triggered communication on top of time-triggered architecture. In : *Proceedings. The 21st Digital Avionics Systems Conference*. IEEE, 2002. p. 13C5-13C5.

[MAIT 85] MAITRE, Henri. 2-Un panorama de la transformation de Hough. *Traitement du signal*, 1985.

[MAL 89] MALLAT, Stephane G. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 1989, vol. 11, no 7, p. 674-693.

[MAN 96] MANJUNATH, Bangalore S. et MA, Wei-Ying. Texture features for browsing and retrieval of image data. *IEEE Transactions on pattern analysis and machine intelligence*, 1996, vol. 18, no 8, p. 837-842.

[MAW 17] MAWLOUD mosbah. *Mesures de distance dans le contexte de la recherche d'images par le contenu (cbir). 2017.* Thèse de doctorat. Université 20 Août 1955-Skikda.

[MAZ 08] MAZO, Manuel et TABUADA, Paulo. On event-triggered and self-triggered control over sensor/actuator networks. In : *2008 47th IEEE Conference on Decision and Control*. IEEE, 2008. p. 435-440.

[MEE 13] MEENA, Mamta, MALLYA, Shrutika, et TALATI, Shrusti. Effectiveness of Saas Cloud Model For Retrieving Images From CBIR System. *International Conference on Innovative and Advanced Technologies in Engineering*, 2013.

[MEE 16] MEENA, Mamta, BHARADI, Vinayak A., *et al.* Hybrid Wavelet Based CBIR System using Software as a Service (SaaS) Model on public Cloud. *Procedia Computer Science*, 2016, vol. 79, p. 278-286.

[MEH 96] MEHTA, Manish, AGRAWAL, Rakesh, et RISSANEN, Jorma. SLIQ: A fast scalable classifier for data mining. In : *International conference on extending database technology*. Springer, Berlin, Heidelberg, 1996. p. 18-32.

[MEL 11] MELL, Peter, GRANCE, Tim, *et al.* The NIST definition of cloud computing. 2011.

[MIB 11] MIBULUMUKINI, MAKIESE. De la perception des images à l'algorithme Log-Gabor PCA. In : *Workshop sur les Technologies de l'Information et de la Communication*. 2011.

[MIK 05] MIKOLAJCZYK, Krystian et SCHMID, Cordelia. A performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 2005, vol. 27, no 10, p. 1615-1630.

[MOR 63] MORGAN, James N. et SONQUIST, John A. Problems in the analysis of survey data, and a proposal. *Journal of the American statistical association*, 1963, vol. 58, no 302, p. 415-434.

[MOR 73] MORGAN, James N. et MESSENGER, Robert C. THAID, a sequential analysis program for the analysis of nominal scale dependent variables. 1973.

[MUL 04] MÜLLER, Henning, MICHOUX, Nicolas, BANDON, David, *et al.* A review of content-based image retrieval systems in medical applications—clinical benefits and future directions. *International journal of medical informatics*, 2004, vol. 73, no 1, p. 1-23.

[NAN 01] NANCI, Dominique, ESPINASSE, Bernard, COHEN, Bernard, *et al. Ingénierie des systèmes d'information: Merise: deuxième génération*. Paris : Vuibert, 2001.

[NAS 98] NASTAR, Chahab, MITSCHKE, Matthias, MEILHAC, Christophe, *et al.* Surfimage: a flexible content-based image retrieval system. In : *Proceedings of the sixth ACM international conference on Multimedia*. 1998. p. 339-344.

[NIC 05] NICOLAS, Stéphane, KESSENTINI, Yousri, PAQUET, Thierry, *et al.* Handwritten document segmentation using hidden Markov random fields. In : *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*. IEEE, 2005. p. 212-216.

[ODR 13] O'DRISCOLL, Aisling, DAUGELAITE, Jurate, et SLEATOR, Roy D. 'Big data', Hadoop and cloud computing in genomics. *Journal of biomedical informatics*, 2013, vol. 46, no 5, p. 774-781.

[ORT 97] ORTEGA, Michael, RUI, Yong, CHAKRABARTI, Kaushik, *et al.* Supporting similarity queries in MARS. In : *Proceedings of the fifth ACM international conference on Multimedia*. 1997. p. 403-413.

[PAI 18] PAIZ-REYES, Evelyn, NUNES-DE-LIMA, Nadile, et YILDIRIM-YAYILGAN, Sule. GIF image retrieval in cloud computing environment. In : *International Conference Image Analysis and Recognition*. Springer, Cham, 2018. p. 261-268.

[PES 11] PESCH, Roland, SCHMIDT, Gunther, SCHROEDER, Winfried, *et al.* Application of CART in ecological landscape mapping: Two case studies. *Ecological Indicators*, 2011, vol. 11, no 1, p. 115-122.

[POT 18] POTERIE, Audrey. *Arbres de décision et forêts aléatoires pour variables groupées*. 2018. Thèse de doctorat. INSA de Rennes.

[PRA 06] PRASAD, Anantha M., IVERSON, Louis R., et LIAW, Andy. Newer classification and regression tree techniques: bagging and random forests for ecological prediction. *Ecosystems*, 2006, vol. 9, no 2, p. 181-199.

[QUE 08] QUELLEC, Gwénolé. *Indexation et fusion multimodale pour la recherche d'information par le contenu. Application aux bases de données d'images médicales*. 2008. Thèse de doctorat. Télécom Bretagne.

[QUI 86] QUINLAN, J.. Ross . Induction of decision trees. *Machine learning*, 1986, vol. 1, no 1, p. 81-106.

[QUI 14] QUINLAN, J. Ross. *C4. 5: programs for machine learning*. Elsevier, 2014.

[RAK 05] RAKOTOMALALA, Ricco. Arbres de décision. *Revue Modulad*, 2005, vol. 33, p. 163-187.

[RAM 13] RAMAMURTHY, B. et CHANDRAN, K. R. CBMIR: Content Based Medical Image Retrieval Using Shape and Texture Content. *Advances in Modeling B*, 2013, p. 84-95.

[REG 14] REGNIERS, Olivier. *Méthodes d'analyse de texture pour la cartographie d'occupations du sol par télédetection très haute résolution: application à la fôret, la vigne et les parcs ostréicoles*. 2014. Thèse de doctorat.

[ROS 69] ROSENFELD, Azriel. Picture processing by computer. *ACM Computing Surveys (CSUR)*, 1969, vol. 1, no 3, p. 147-176.

[RUB 13] RUBNER, Yossi et TOMASI, Carlo. *Perceptual metrics for image database navigation*. Springer Science & Business Media, 2013.

[RUI 96] RUI, Yong, SHE, Alfred C., et HUANG, Thomas S. Modified Fourier descriptors for shape representation-a practical approach. In : *Proc of First International Workshop on Image Databases and Multi Media Search*. Citeseer, 1996. p. 22-23.

[SAT 11] SATHYADEVI, G. Application of CART algorithm in hepatitis disease diagnosis. In : *2011 International Conference on Recent Trends in Information Technology (ICRTIT)*. IEEE, 2011. p. 1283-1287.

[SCL 99] SCLAROFF, Stan, LA CASCIA, Marco, SETHI, Saratendu, *et al.* Unifying textual and visual cues for content-based image retrieval on the world wide web. *Computer Vision and Image Understanding*, 1999, vol. 75, no 1-2, p. 86-98.

[SHA 18] AL-SHAWAKFA, Emad et ALSGHAIER, Hiba. An empirical study of cloud computing and big data analytics. *International Journal of Innovative Computing and Applications*, 2018, vol. 9, no 3, p. 180-188.

[SHAN 48] SHANNON, Claude E. A mathematical theory of communication. *The Bell system technical journal*, 1948, vol. 27, no 3, p. 379-423.

[SHO 11] SHOTTON, Jamie, FITZGIBBON, Andrew, COOK, Mat, *et al.* Real-time human pose recognition in parts from single depth images. In : *CVPR 2011*. Ieee, 2011. p. 1297-1304.

[SME 00] SMEULDERS, Arnold WM, WORRING, Marcel, SANTINI, Simone, *et al.* Content-based image retrieval at the end of the early years. *IEEE Transactions on pattern analysis and machine intelligence*, 2000, vol. 22, no 12, p. 1349-1380.

[SMI 97] SMITH, John R. et CHANG, Shih-Fu. Querying by color regions using the VisualSEEk content-based visual query system. *Intelligent multimedia information retrieval*, 1997, vol. 7, no 3, p. 23-41.

[SRI 18] SRIVASTAVA, Priyanshu et KHAN, Rizwan. A review paper on cloud computing. *International Journals of Advanced Research in Computer Science and Software Engineering*, 2018, vol. 8, no 6, p. 17-20.

[STR 95] STRICKER, Markus Andreas et ORENGO, Markus. Similarity of color images. In : *Storage and retrieval for image and video databases III*. International Society for Optics and Photonics, 1995. p. 381-392.

[SUD 18] SUDHEER, Devulapalli et KRISHNAN, Rajakumar. An Efficient Image Retrieval System Using Edge, LBP and Wavelet based Texture Analysis. Jour of Adv Research in Dynamical & Control Systems, Vol. 10, 10-Special Issue, 2018

[SUT 14] SUTTON-CHARANI, Nicolas. *Apprentissage à partir de données et de connaissances incertaines: application à la prédiction de la qualité du caoutchouc*. 2014. Thèse de doctorat.

[SWA 91] SWAIN, Michael J. et BALLARD, Dana H. Color indexing. *International journal of computer vision*, 1991, vol. 7, no 1, p. 11-32.

[TOL 06] TOLLARI, Sabrina. *Indexation et recherche d'images par fusion d'informations textuelles et visuelles*. 2006. Thèse de doctorat. Toulon.

[TRO 09] TROJACANEC, Katarina, DIMITROVSKI, Ivica, et LOSKOVSKA, Suzana. Content based image retrieval in medical applications: an improvement of the two-level architecture. In : *IEEE EUROCON 2009*. IEEE, 2009. p. 118-121.

[TUC 90] TUCERYAN, Mihran et JAIN, Anil K. Texture segmentation using Voronoi polygons. *IEEE transactions on pattern analysis and machine intelligence*, 1990, vol. 12, no 2, p. 211-216.

[TUC 93] TUCERYAN, Mihran et JAIN, Anil K. Texture analysis. In : *Handbook of pattern recognition and computer vision*. 1993. p. 235-276.

[VER 00] VERTAN, Constantin et BOUJEMAA, Nozha. Using fuzzy histograms and distances for color image retrieval. In : *Challenge of Image Retrieval*. 2000.

[WES 00] WESTERVELD, Thijs. Image Retrieval: Content versus Context. In : *RIAO*. 2000. p. 276-284.

[XU 11] XU, Wen-hua, QIN, Zheng, et CHANG, Yang. Clustering feature decision trees for semi-supervised classification from high-speed data streams. *Journal of Zhejiang University SCIENCE C*, 2011, vol. 12, no 8, p. 615.

[YU 03] YU, Lei et LIU, Huan. Efficiently handling feature redundancy in high-dimensional data. In : *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2003. p. 685-690.

[YUA 18] YUAN, Lina, CHEN, Huajun, et GONG, Jing. Classifications Based Decision Tree and Random Forests for Fanjing Mountains' Tea. In : *IOP Conference Series: Materials Science and Engineering*. 2018. p. 08.

[ZHA 15] ZHANG, Yin, QIU, Meikang, TSAI, Chun-Wei, *et al.* Health-CPS: Healthcare cyber-physical system assisted by cloud and big data. *IEEE Systems Journal*, 2015, vol. 11, no 1, p. 88-95.

Webographies

[1] Extensible Markup Language. http://www.w3.org/XML

[2] Text REtrieval Conference. https://trec.nist.gov/

[3] dicom standard, page d'accueil. https://www.dicomstandard.org

[4] Moteur de recherche web Google. http://www.google.fr.

[5] Moteur de recherche web Lycos. http://www.lycos.fr

[6] Motion Picture Expert Group (MPEG), page d'accueil. https://mpeg.chiariglione.org/standards/mpeg-7

[7] Moteur de recherche web Amazon. https://www.amazon.fr/

[8] Moteur de recherche web IBM. https://www.ibm.com/dz-fr

[9] http://www.granddictionnaire.com/ficheOqlf.aspx?Id_Fiche=26501384 , [Consulté le 04 juin 2020].

[10] https://missarte.wordpress.com/les-fournisseurs/ [Consulté 03-06-2020]

[11] Microsoft Azure Web Services. https://azure.microsoft.com/fr-fr/services/cloud-services/

[12] Amazon Web Services. https://www.scaleway.com/fr/instances-virtuelles/general-purpose/?gclid=Cj0KCQjwwuD7BRDBARIsAK_5YhXE7qdbaiZ3La_JwpckakGA4zm-NPGcX0P-RJjTBqzwRNxP3tniGVoaAiBuEALw_wcB

[13] la plate-forme d'hébergement d'applications. https://www.lws.fr/landing3.php

[14] Microsoft Office 365. https://www.office.com/

[15] Microsoft Skype. https://www.skype.com/fr/

[16] Google Apps. https://play.google.com/store/apps/details?id=com.google.android.googlequicksearchbox&hl=fr

[17] Sales force. https://www.salesforce.com/fr/

[18] Clusters Amazon EC2. https://docs.aws.amazon.com/AmazonECS/latest/developerguide/clusters.html

[19] Google Docs – SaaS. https://docs.google.com/document/u/0/

[20] OneDrive. https://www.microsoft.com/fr-fr/microsoft-365/onedrive/online-cloud-storage

[21] Google Drive. https://www.google.com/intl/fr_tg/drive/

[22] iCloud. https://www.icloud.com/

[23] Dropbox. https://www.dropbox.com/fr/

[24] Hadoop . https://hadoop.apache.org/

[25] World Health Organization, https://www.who.int. [Consulté le 10 Janvier 2019].